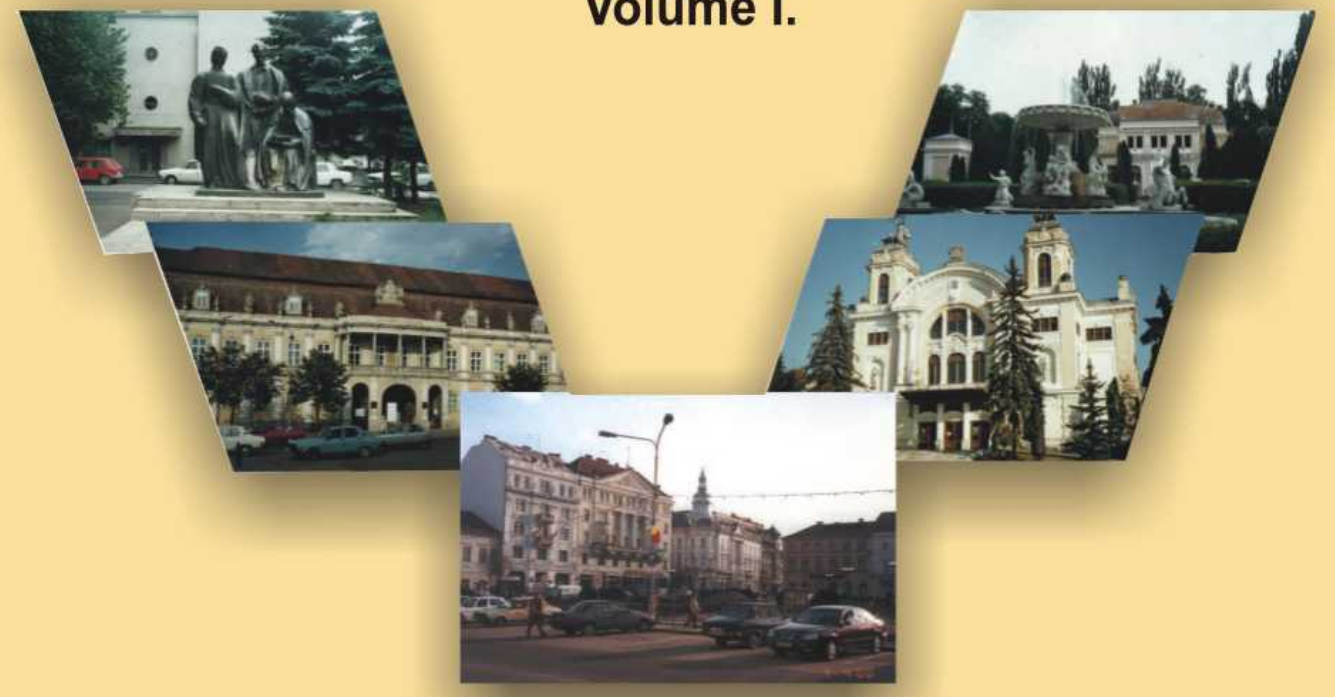
STUDIA UNIVERSITATIS BABEŞ-BOLYAI
INFORMATICA SERIES

KEPT 2007

KNOWLEDGE ENGINEERING

PRINCIPLES AND TECHNIQUES

Volume I.



Special Issue

Babeş-Bolyai University

KEPT 2007

KNOWLEDGE ENGINEERING

PRINCIPLES

AND

TECHNOLOGIES

Cluj-Napoca, June 6–8, 2007

Sponsors of the conference

Siemens Program and System Engineering SRL

Integrated Systems Development Corporation

Transart S.R.L.

CONTENTS

Invited Lectures

Natural Language Processing

Artificial Intelligence

Software Engineering

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 9–10

SEMANTIC SIMILARITY KNOWLEDGE AND ITS APPLICATIONS

DIANA INKPEN

Abstract

Semantic relatedness refers to the degree to which two concepts or words are related. Humans are able to easily judge if a pair of words are related in some way. For example, most would agree that "apple" and "orange" are more related than are "apple" and "toothbrush". Semantic similarity is a subset of semantic relatedness.

In this talk I will present several methods for computing the similarity of two words, following two directions: dictionary-based methods that use WordNet, Roget's thesaurus, or other resources; and corpus-based methods that use frequencies of co-occurrence in corpora (cosine method, latent semantic indexing, mutual information, etc). I will present several applications of word similarity knowledge: detecting words that do not fit into their context (real-word error correction), detecting speech recognition errors, solving TOEFL-style synonym questions, and synonym choice in context (for writing aid tools).

I will also present a method for computing the similarity of two texts, based on the similarities of their words. Applications of text similarity knowledge include: designing exercises for second language-learning, acquisition of domain-specific corpora, information retrieval, and text categorization.

At the end, I will present cross-language extensions of the methods for similarity of words and texts.

Biography

Dr. Diana Inkpen is an Assistant Professor of Computer Science at the School of Information Technology and Engineering, University of Ottawa since July 2003. She obtained her doctorate from the University of Toronto, Department of Computer Science. She has a Masters in Computer Science and Engineering from the Technical University of Cluj-Napoca, Romania. Her research projects and

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.

©2007 Babeş-Bolyai University, Cluj-Napoca

publications are in the areas of Computational Linguistics and Artificial Intelligence, more specifically: Information Retrieval, Information Extraction, Natural Language Understanding, Natural Language Generation, Speech Processing, and Intelligent Agents for the Semantic Web.

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 11–11

SOME REPRESENTATION STRUCTURES FOR COMPUTATIONAL LINGUISTICS

ALAIN LECOMTE

Abstract

In this talk, I will present useful tools for producing semantic representations derived by an analysis of a sentence, and I will suggest how to take the discourse context in consideration for this goal. All these tools are taken from recent developments in logics, mainly in Resource Sensitive Logics (particularly Linear Logic). After viewing classical tools like the lambda calculus and its use in a Montagovian perspective, we will recall some aspects of the now well known "Curry-Howard" isomorphism. These classical views are now surpassed by new structures: proof-nets, which replace lambda terms and whose main advantages reside in their geometrical properties, and continuations, which make it possible to take contexts as arguments. I will particularly develop the point on the calculus of continuations (and its lambda-mu calculus version) with regards to the question of interpretation in context (anaphors, deictics, ...).

Biography

Prof. Alain Lecomte is presently professor of Theoretical Linguistics at the University Paris 8 (France), and member of the Laboratory of "Formal Structures of Language" (UMR 7023, CNRS). He was previously professor of logics and epistemology at the University Grenoble 2 and member of the Institute of Applied Mathematics of Grenoble. He is also an associate member of the team "SIGNES", an INRIA project in Bordeaux. He obtained his doctorate of Applied Mathematics from the University of Grenoble, and his habilitation in Computational Linguistics from the University of Clermont-Ferrand. His research projects are in the areas of Theoretical Linguistics, Computational Linguistics and Logics. He is presently coordinating a national project on "Ludics and Formal Pragmatics".

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.

©2007 Babeş-Bolyai University, Cluj-Napoca

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 12–13

USING WIKIPEDIA FOR AUTOMATIC WORD SENSE DISAMBIGUATION

RADA MIHALCEA

Abstract

Ambiguity is inherent to human language. In particular, word sense ambiguity is prevalent in all natural languages, with a large number of the words in any given language carrying more than one meaning.

In this talk, I will present a new approach for building sense tagged corpora using Wikipedia as a source of sense annotations. Starting with the hyperlinks available in Wikipedia, I will show how one can generate sense annotated corpora that can be used for building accurate and robust sense classifiers. Through word sense disambiguation experiments performed on the Wikipedia-based sense tagged corpus generated for a subset of the Senseval ambiguous words, I will show that the Wikipedia annotations are reliable, and the quality of a sense tagging classifier built on this data set exceeds by a large margin the accuracy of an informed baseline that selects the most frequent word sense by default.

Biography

Rada Mihalcea is an Assistant Professor of Computer Science at the University of North Texas. Her research interests are in lexical semantics, graph-based algorithms for natural language processing, minimally supervised natural language learning, and multilingual natural language processing. She is currently involved in a number of research projects, including knowledge-based word sense disambiguation, (non-traditional) methods for building annotated corpora with volunteer contributions over the Web, graph-based algorithms for text processing, opinion and sentiment analysis, and computational humour. She has published a large number of articles in books, journals, and proceedings, in these and related areas. She is the president of the ACL Special Group on the Lexicon (SIGLEX), and a board member for the ACL Special Group on Natural Language Learning (SIGNLL). She serves on the editorial boards of the journal of Computational Linguistics, the journal of Language Resources and Evaluations, the Journal of Natural Language

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.

©2007 Babeş-Bolyai University, Cluj-Napoca

Engineering, the Journal of Research on Language and Computation, and the recently established journal of Interesting Negative Results in Natural Language Processing and Machine Learning.

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 14–14

THE ROLE OF LINGUISTIC INFORMATION FOR SHALLOW LANGUAGE PROCESSING

CONSTANTIN ORĂŞAN

Abstract

Many methods in computational linguistics rely on shallow processing to achieve their goals. The advantage of these methods in comparison to deep processing methods is that they do not require the building of elaborate representations of the text to be processed or to perform reasoning on this data, and as a result they can be more easily implemented. This talk will show how a shallow method for automatic summarisation can improve its performance by adding different types of linguistic information.

Biography

Dr. Constantin Orasan is a Senior Lecturer in Computational Linguistics at School of Humanities, Languages and Social Sciences, University of Wolverhampton, UK since October 2005. He has obtained his doctorate in automatic summarisation from the same university. His research interests are in automatic summarisation, question answering, anaphora and coreference resolution, corpus building and annotation.

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.

©2007 Babeş-Bolyai University, Cluj-Napoca

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 17–24

THE ROLE OF LINGUISTIC INFORMATION FOR SHALLOW LANGUAGE PROCESSING

CONSTANTIN ORĂŞAN[(1)]

Abstract. Many methods in computational linguistics rely on shallow processing to achieve their goals. The advantage of these methods in comparison to deep processing is that they do not require the building of elaborate representations of the text to be processed or to perform reasoning on this data, and as a result they can be more easily implemented. This paper shows how shallow methods for automatic summarisation can improve their performance by adding different types of linguistic information.

1. Introduction

In language processing it is generally accepted that two approaches can be employed. On the one hand there are *deep linguistic approaches* which build an elaborate representation of the problem they resolve in order to "understand" the texts they process and make inferences. On the other hand there are *shallow linguistic approaches* where different types of information are extracted from the text and then combined in order to solve the problem tackled, but no attempt is made to understand the text they process. Deep linguistic approaches have been widely used to implement different grammatical formalism, but they were also used with various degrees of success in real-world applications such as information extraction and automatic summarisation. The drawback of these methods is that they lack robustness and coverage due to the fact that quite often they rely on hand coded resources. In contrast, shallow approaches are robust at the expense of performance, which usually is lower than that of deep processing. Due to the fact that they require less effort to implement they have been widely used for all kind of purposes. In this paper, we show that it is possible to overcome some of the limitations of shallow processing methods and improve their performance by combining different linguistic information. In order to prove this, automatic summarisation is taken as a case study. The paper is structured as follows: Section 2 briefly presents existing methods in automatic summarisation classifying them in

2000 *Mathematics Subject Classification.* 91F20, 68T35.
Key words and phrases. automatic summarisation, evaluation, shallow processing.

©2007 Babeş-Bolyai University, Cluj-Napoca

shallow and deep methods. Section 3 presents and evaluates different automatic summarisation methods showing how the results improve with the addition of linguistic information. Section 4 discusses the results and concludes the paper.

2. Deep vs. shallow processing in automatic summarisation

The field of *automatic summarisation* develops automatic methods which try to replace the human summarisers by producing summaries using automatic means. Unfortunately, with the current technology it is difficult to produce automatic summaries which are of similar quality with summaries produced by professional summarisers. Instead, it can produce *indicative summaries* which can indicate the content of a document and can help readers decide whether the content is relevant to their interests.

In general, in automatic summarisation two main approaches are employed: *automatic extraction methods* and *automatic abstraction methods* [10]. The former produce *extracts* which are sets of units (i.e. sentences, paragraphs or clauses) extracted with no or little modifications from the source text(s), and normally employ shallow linguistic processing. The later produce *abstracts* which present the most important information in the text to be summarised, but contain units not present in the source. Quite often, in order to produce abstracts deep linguistic processing is required. The remaining of this section presents a brief overview of the methods employed in automatic summarisation with emphasis on whether they rely on deep or shallow processing.

The way most of the shallow automatic summarisation methods work is to determine a score for each sentence and on the basis of this score, extract the sentences with the highest scores until the desired length of summary is reached. The first such summarisation method relied on the distribution of words to determine sentences which contain important information for a text Luhn [14]. Even though the method was proposed almost 45 years ago, its promising results encouraged other researchers to apply similar approaches, in many cases in combination with other methods [7, 13, 25, 24, 9]

Edmundson [7] noticed that the presence of certain words can indicate that a sentence is important or that it can be discarded during the summarisation process. Given the beneficial influence of this method on the quality of extracts it was extended to phrases [20] , and now is widely used in combination with other methods [13, 24, 11]. In a similar manner *named entities* were used as an indicator of a sentence's importance [13, 22].

Shallow processing was also used to determine the discourse structure of texts and produce summaries. Cue words and phrases were used in [18, 16, 5] to derive the rhetorical structure of a text [15] and employ it in the summarisation process. Links between entities in a text were also used to produce summaries. Boguraev and Kennedy [4] and Azzam et. al. [1] employed anaphoric and coreferential links, whilst Barzilay and Elhadad [2] focused on lexical repetition.

A general characteristic of the methods presented above is that they do not try to "understand" the text. In contrast, methods which rely on deep linguistic processing try to imitate the way humans produce summaries by understanding a text first and then generate an abstract on the basis of information understood. Because, in order to function, these systems require large quantities of information about the domain, these methods are also called *knowledge rich methods*. The downside of this approach is the fact that the systems are domain dependent, which means they cannot easily be ported to different domains or to be used in domain independent contexts.

The best known system based on deep linguistic processing to produce summaries is FRUMP [6]. The approach taken by this system relies on *sketchy scripts* to encode information about the events it can "understand", the participants in these events, and the way in which the participants interact with each other and with the environment. The participants were identified using surface clues, and their actions traced using a simple inference engine. Rumelhart [21] developed a system to understand and summarise simple stories, using a grammar which generated semantic interpretations of the story on the basis of hand-coded rules. The SUSY system [8] is of particular interest because it tries to implement the theory proposed by Kintsch [12] to understand and summarise a text, and therefore attempts to replicate the way humans summarise texts. This system relies on linguistic knowledge to understand the meaning and structure of a text, and on world knowledge to reason and infer new information.

3. Linguistic information for shallow automatic summarisation

The previous section has presented several methods used in automatic summarisation. In this section some of these methods are implemented and evaluated, showing that in many cases addition of linguistic information has a beneficial effect on the informativeness of summaries. This section starts with a description of the corpus used in the experiments and the evaluation method employed. Sections 3.2 - 3.6 present the summarisation methods investigated here, followed by their evaluation in Section 3.7.

3.1. **Corpus and evaluation method.** For the experiments described in this paper, a corpus of journal articles published in the Journal of Artificial Intelligence Research (JAIR) was used. This corpus contains 65 articles with over 600,000 words. In order to assemble this corpus, the electronic versions of these articles were downloaded and converted to plain text. For the purpose of automatic summarisation the corpus was automatically annotated with sentence boundaries, token boundaries and part-of-speech information using the FDG tagger [23]. To evaluate the performance of automatic summarisation methods the author produced abstract was identified and extracted from the article.

The evaluation measure used in this paper calculates the similarity between an automatic extract and the author produced abstract using the *cosine measure*, a very popular measure for determining the similarity between two vectors. The formula to calculate this is:

$$\cos(\vec{S_a}, \vec{S_h}) = \frac{\sum_{i=1}^{n} S_a(i) S_h(i)}{\sqrt{\sum_{i=1}^{n} S_a(i)} \sqrt{\sum_{i=1}^{n} S_h(i)}} \quad (1)$$

where S_a and S_h are the vectors built from the automatic and human summaries respectively, n is the number of distinct words in $S_a \cup S_h$, and $S_a(i)$ and $S_h(i)$ are the frequencies of word i in S_a and S_h respectively. In order to make the similarity value more accurate, a stoplist is applied before building the vectors.

3.2. **Upper limit and baseline.** Using the evaluation method presented in the previous section, it is possible to identify in every text a set of sentences which has the maximum similarity score with the source's human abstract. This maximum figure represents the upper limit any extraction method could reach, and indicates that the only way to further increase similarity is to produce an abstract. In this paper, we employ the method proposed in [19] to identify the upper limit of extraction methods using both the greedy algorithm and the genetic algorithm proposed in that paper, depending on which compression rate is used. The results of the upper limit are presented in Table 1.

A baseline is usually a very simple method which does not really employ much knowledge to produce a summary, and which is normally used for comparison. In this research, it was decided to consider as baseline a method which extracts the first and last sentences from paragraphs, starting with the first paragraph in the text, until the desired length is achieved. The justification for this baseline can be found in research by Baxendale [3] who noticed that the first and last sentences from paragraphs are more important than others in scientific documents. The results of this baseline are presented in Section 3.7.

3.3. **Term-based summarisation.** Term-based summarisation assumes that the importance of a sentence can be determined on the basis of the words it contains. The most common way of achieving this is to weight all the words in a text, and calculate the score of a sentence by adding together the weights of the words within it. In this way, a summary can be produced by extracting the sentences with the highest scores until the desired length is reached. In this section two token weighting methods are used in the summarisation process: term frequency and TF*IDF. Each of them is presented next.

3.3.1. *Term frequency.* It was noticed that when a person writes a text, he or she normally repeats concepts as they progress through the text, and those concepts which are repeated most are the ones which are linked to the main topics of the text [14]. Using this observation, it is possible to assign to each token a score equal

to its frequency in order to indicate the topicality of the concept represented by it.

$$Score(t) = TF_t = the\,frequency\,of\,token\,\mathbf{t}\,in\,the\,text \tag{2}$$

The main drawback of term frequency is that it wrongly assigns high scores to frequent tokens such as prepositions and articles. For this reason, a stoplist is used to filter out these words.

3.3.2. *TF*IDF.* The elimination of stopwords does not ensure that only important tokens receive a high score. In order to address this problem, *document frequency* can be used. The assumption of this measure is that the importance of a token is inversely proportional to the number of documents in which it appears. The inverse document frequency (IDF) on its own is a relatively weak indicator of the token's importance, and for this reason very often it is used in conjunction with the term frequency. The formula used to calculate TF*IDF is:

$$TF * IDF(t) = TF_t * log\,\frac{N}{n_t} \tag{3}$$

where N is the number of documents in the collection, n_t is the number of documents in the collection which contain the token t and TF_t is the term frequency as calculated in the previous section.

3.4. **Anaphora resolution for automatic summarisation.** The term-based summariser presented in the previous section relies on word frequencies to calculate the score of a sentence, but because some of these words are referred to by pronouns the frequency of the concepts they represent are not correctly calculated. In this section, a robust anaphora resolver is used to assign semantic information to pronouns and in this way obtain more reliable frequency counts of concepts, which in turn improves the results of the term-based summariser. After experimenting with several anaphora resolvers, MARS [17], a robust anaphora resolution method which relies on a set of boosting and impeding indicators to select the antecedent of a pronoun, was selected. In order to evaluate the performance of MARS a third of the corpus was annotated with anaphoric links. The results of the evaluation indicate a success rate of around 51%.

3.5. **Indicating phrases.** First introduced by Paice [20], indicating phrases are groups of words which can be used to determine the importance of a sentence that contains them. For scientific domain typical examples of indicating phrases are phrases such as *in this paper we present, we conclude that.* In order to acquire a list of indicating phrases relevant to the scientific domain, a corpus of scientific abstracts was used to extract a list of 4-grams which was then manually edited. Sentence were scored according to how many indicating phrases they contained.

Compression rate	2%	3%	5%	6%	10%
Baseline	0.260	0.327	0.419	0.440	0.479
Term-based summariser using TF	0.415	0.443	0.461	0.468	0.484
Term-based summariser using TF*IDF	0.396	0.427	0.467	0.472	0.496
Summariser which uses indicating phrases	0.428	0.452	0.492	0.500	0.527
Term-based summariser using TF and MARS	0.455	0.480	0.494	0.500	0.513
Term-based summariser using TF*IDF and MARS	0.428	0.463	0.499	0.503	0.520
Combination	0.480	0.496	0.532	0.535	0.541
Upper limit of extraction methods	0.725	0.743	0.753	0.748	0.788

TABLE 1. The informativeness of summaries produced using different methods

3.6. **Combination of modules.** The term-based summariser and the summariser based on indicating phrases are rarely used independently. For this reason, a summariser which combines information from the two modules was implemented. In this summariser the scores assigned by the term-based summarisation module and the module based on indicating phrases are normalised to a value between 0 and 1, and are combined using a linear function. After experiments, it was decided to give each module a weight of 1.

3.7. **Evaluation results.** In order to evaluate the influence of different linguistic information on the informativeness of summaries produced 2%, 3%, 5%, 6% and 10% summaries have been produced for each text. Table 1 presents the results of the evaluation.

As expected, all the methods investigated in this paper perform better than the baseline. Term-based summarisers produce significantly better results than the baseline for all compression rates but 10% summaries where the differences are not statistically significant. Comparison between the results of the two term-based summarisers reveal that the performance of the two methods depend on the compression rate. For high compression rates (i.e. 2% and 3%) it is necessary to weight words using TF, whereas for lower compression rates TF*IDF should be used. The same phenomenon happens when the term-based summariser is enhanced with information from the anaphora resolver. The summariser based on indicating phrases performs significantly worse than the enhanced term-based summariser at high compression rates, but at lower compression rates (i.e. 5%, 6% and 10%) the differences are not statistically significant, or it performs better than the enhanced summariser. The combination of modules leads to much better results than running each module individually.

4. Discussion and conclusions

In this paper we investigated the influence of shallow linguistic information on the quality of automatic summarisation. To achieve this several automatic

summarisation methods were developed and evaluated. First a method to determine the upper limit of extraction methods was run to show that by using shallow linguistic methods, which produce extracts and not abstracts, there is a limit on how similar the automatic summaries can be in comparison to human abstracts. All the methods evaluated in the paper performed well below this upper limit.

The paper has also shown that by combining shallow linguistic information it is possible to improve the informativeness of automatic summaries. A baseline which relies on non-linguistic positional information was outperformed by all the methods which benefit from linguistic information. Simple lexical information about the frequency of tokens was enough to produce better results than the baseline. When more accurate words frequency are calculated using semantic information provided by an anaphora resolver the results improve significantly. Very similar summaries from the point of view of informativeness are produced using indicating phrases which can be considered both lexical information and pseudo-discourse markers. A summariser which combines all this information produces better summaries indicating that combination of shallow linguistic information can lead to better results.

References

[1] Saliha Azzam, Kevin Humphrey, and Robert Gaizauskas. Using coreference chains for text summarisation. In Amit Bagga, Breck Baldwin, and Sara Shelton, editors, *Coreference and Its Applications*, pages 77 – 84, University of Maryland, College Park, Maryland, USA, June 1999.

[2] Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*, pages 111 – 121. The MIT Press, 1999.

[3] Phyllis B. Baxendale. Man-made index for technical literature - an experiment. *I.B.M. Journal of Research and Development*, 2(4):354 – 361, 1958.

[4] Branimir Boguraev and Christopher Kennedy. Salience-based content characterisation of text documents. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*, pages 99 – 110. The MIT Press, 1999.

[5] Simon H. Corston-Oliver. Beyond string matching and cue phrases: Improving the efficiency and coverage in discourse analysis. In *AAAI Spring Symposium on Intelligent Text Summarisation*, pages 9 – 15, Stanford, California, USA, March 23-25 1998.

[6] G. DeJong. An overview of the FRUMP system. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for natural language processing*, pages 149 – 176. Hillsdale, NJ: Lawrence Erlbaum, 1982.

[7] H. P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264 – 285, April 1969.

[8] Danilo Fum, Giovanni Guida, and Carlo Tasso. Evaluating importance: a step towards text summarisation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 840 – 844, Los Altos CA, August 1985.

[9] Le An Ha and Constantin Orăsan. Concept-centred summarisation: producing glossary entries for terms using summarisation methods. In *Proceedings of RANLP2005*, pages 219 – 225, Borovets, Bulgaria, September 21 – 23 2005.

[10] Eduard Hovy. Text summarisation. In Ruslan Mitkov, editor, *The Oxford Handbook of computational linguistics*, pages 583 – 598. Oxford University Press, 2003.

[11] Eduard Hovy and Chin-Yew Lin. Automated text summarization in SUMMARIST. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*, pages 81 – 94. The MIT Press, 1999.

[12] Walter Kintsch. *The representation of meaning in memory*. The Experimental psychology series. Lawrence Erlbaum Associates Publishers, 1974.

[13] Julian Kupiec, Jan Pederson, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th ACM/SIGIR Annual Conference on Research and Development in Information Retrieval*, pages 68 – 73, Seattle, July 09 – 13 1995.

[14] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159 – 165, 1958.

[15] Willliam C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Description and construction of text structures. In *NATO Advanced Research Workshop on Natural Language Generation*, pages 85 – 95. 1986.

[16] Daniel Marcu. From discourse structures to text summaries. In Inderjeet Mani and Mark Maybury, editors, *Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization*, pages 82 – 88, Madrid, Spain, 1997. ACL.

[17] Ruslan Mitkov, Richard Evans, and Constantin Orăsan. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *Proceedings of CICLing-2002*, pages 168 – 186, Mexico City, Mexico, February 2002.

[18] Kenji Ono, Kakuo Sumita, and Seiji Miike. Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 344 – 348, Kyoto, Japan, 1994.

[19] Constantin Orăsan. Automatic annotation of corpora for text summarisation: A comparative stuty. In *Proceedings of the 6th International Conference CICLing2005*, pages 670 – 681, Mexico City, Mexico, February 2005. Springer-Verlag.

[20] Chris D. Paice. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In R. N. Oddy, C. J. Rijsbergen, and P. W. Williams, editors, *Information Retrieval Research*, pages 172 – 191. London: Butterworths, 1981.

[21] E. Rumelhart. Notes on a schema for stories. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 211 – 236. Academic Press Inc, 1975.

[22] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1818 – 1824, Las Palmas de Gran Canaria, Spain, May 2002.

[23] Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing*, pages 64 – 71, Washington D.C., USA, March 31 - April 3 1997.

[24] Simone Teufel and Marc Moens. Sentence extraction as a classification task. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scallable Text Summarization*, pages 58 – 59, Madrid, Spain, July 11 1997.

[25] Klaus Zechner. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *COLING - 96, The International Conference on Computational Linguistics*, pages 986 – 989, Copenhagen, Denmark, August 1996.

[1]Research Group in Computational Linguistics, University of Wolverhampton, Wolverhampton, UK

E-mail address: C.Orasan@wlv.ac.uk

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 25–32

LARGE SCALE EXPERIMENTS WITH FUNCTION TAGGING

MIHAI LINTEAN [(1)] AND VASILE RUS[(2)]

Abstract. We present in this paper large scale experiments with two Decision Trees based approaches to the task of function tagging. The task of function tagging involves labeling certain nodes in an input parse tree with a set of functional marks such as logical subject, predicate, etc. In the first approach, we consider only nodes that are labeled with a functional tag. In the second approach, all nodes are considered whether they are labeled with function tags or not. The non-labeled nodes are simply considered being labeled with the generic tag NON-F. The results obtained on a standard data set are significantly outperforming baseline approaches when the most frequent tag is assigned.

1. Introduction

Syntactic parsing in its most general definition may be viewed as discovering the underlying syntactic structure of a sentence. The specificities include the types of elements and relations that are retrieved by the parsing process and the way in which they are represented. In this paper we focus on Treebank-style[1] syntactic parsers that retrieve phrases, e.g. NP - noun phrase, VP - verb phrase, S - sentence, and hierarchically organize them in parse trees.

Treebank-style state-of-the-art statistical parsers limit their output to basic structures such as NPs, VPs, PPs (Prepositional Phrases). They are not able to deliver richer syntactic information such as logical subject or predicate although Penn Treebank, the annotated corpus that state of the art parsers use for training, contains annotations for such type of information in the form of function tags and remote dependencies coded as traces. This paper presents experiments with Decision Trees based approaches to augment the output of Treebank-style syntactic parsers with functional information.

In Section 2.2 of Bracketing Guidelines for Treebank II [1], there are 20 function tags grouped in four categories: form/function discrepancies, grammatical role, adverbials, and miscellaneous. Up to 4 function tags can be added to the standard syntactic label (NP, ADVP - Adverbial Phrase, PP, etc.) of each bracket. Those

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.
Key words and phrases. parsing, function tagging.

©2007 Babeş-Bolyai University, Cluj-Napoca

TABLE 1. Categories of Function Tags

Category	Function Tags
Grammatical	DTV, LGS, PRD, PUT, SBJ, VOC
Form/Function	NOM, ADV, BNF, DIR, EXT, LOC, MNR, PRP, TMP
Topicalisation	TPC
Miscellaneous	CLR, CLF, HLN, TTL

tags were necessary to distinguish words or phrases that belong to one syntactic category and are used for some other function or when it plays a role that is not easily identified without special annotation. We rearranged the four categories into four new categories based on corpus evidence, in a way similar to [2]. The new four categories are given in Table 1 and were derived so that no two labels from same new category can be attached to the same bracket.

We present in this paper two approaches to automatically assign function tags to constituents in parse trees. The function tags assignment problem is viewed as a classification problem, where the task is to select the correct tag from a list of candidate tags. In the first approach, we only considered constituents from syntactic parse trees in Treebank that are labeled with functional tags. In the second approach, we considered both labeled and unlabeled constituents. The unlabeled constituents are simply considered being labeled with the generic label *NON-F* (non-functional tag).

This is the first large scale evaluation of Decision Trees based solutions to the task of functional tagging. We used the full data set that Penn Treebank makes available in order to train and test a Decision Trees based functional tagger. In [3], the Decision Trees approach was abandoned (see section *5.2 Why we abandoned decision trees*) due to memory limitations. We addressed the memory issue by using a set of smart preprocessing steps applied to training and test data and by using a High-Performance Computer (IBM AIX System with 64GB of RAM). The preprocessing was necessary in order to reduce the number of distinct values some features have, e.g. lexical based features such as head word. Too many distinct values for these features led to very large Decision Trees that would not fit even in the memory of a High-Performance Computer similar to the one that we used.

The rest of the paper is organized as follows. The next section presents related work in the area of functional tagging and Decision Trees. Section 3 describes the problem we study in this paper while Section 4 presents the generic model we use to solve the problem. Next, the *Experimental Setup and Results* section provides details about the conducted experiments and results. *Conclusions* end the paper.

2. Related Work

Blaheta and Johnson [2] addressed the task of function tagging. They use a statistical algorithm based on a set of features grouped in *trees*, rather than *chains*. The advantage is that features can better contribute to overall performance for cases when several features are sparse. When such features are conditioned in a chain model the sparseness of a feature can have a dilution effect of a ulterior (conditioned) one.

Previous to that, Michael Collins [6] only used function tags to define certain constituents as complements. The technique was used to train an improved parser.

Related work on enriching the output of statistical parsers, with remote dependency information, were exposed in [8] and [7]. Minipar [9] is a dependency parser for English that can identify grammatical dependencies (e.g. *comp1* - first complement, *rel* - relative clause) among words in a sentence. The set of grammatical dependencies in Minipar overlaps at some extent with the functional tags from Treebank.

In the vast literature on Decision Trees, also known as classification trees or hierarchical classifiers, at least two seminal works must be mentioned, those by Breiman et al. [4] and Quinlan [12]. The former originated in the field of statistical pattern recognition and describes a system, named CART (Classification And Regression Trees), which has mainly been applied to medical diagnosis and mass spectra classification. The latter synthesizes the experience gained by people working in the area of machine learning and describes a computer program, called ID3, which has evolved into a new system, named C4.5, by Quinlan.

We opted for Decision Trees for two main reasons. First, Decision Trees can be mapped into rules that are easily interpretable by humans. Second, previous attempts[3] to address the problem of function tagging with Decision Trees was abandoned. We wanted to fill this gap in literature about how suitable Decision Trees are at the function tagging task.

3. The Problem

The task of function tagging is to add extra labels, called function tags, to certain constituents in a parse tree. Let us pick as an illustrative example the sentence *Mr. Hahn rose swiftly through the ranks*[1]. A state-of-the-art syntactic parser will generate the parse tree shown on the left hand side in Figure 1. Each word in the sentence has a corresponding leaf (terminal) node, denoting that word's part of speech. For instance, the word *ranks* has NNS as its part of speech (NNS indicates a common noun in plural form). All the other nodes, called internal or non-terminal nodes, will be labeled with a syntactic tag that marks the grammatical phrase corresponding to the node, such as NP, VP, or S.

[1]This sentence is from Wall Street Journal portion of Penn Treebank.

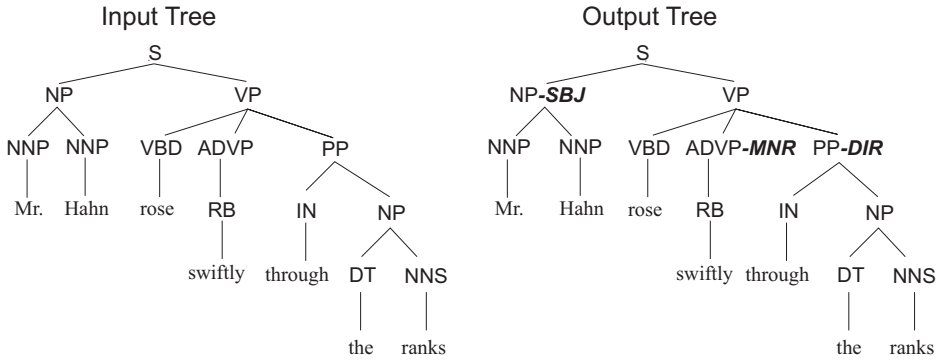


FIGURE 1. A Simple Syntactic Tree

It is not obvious from such syntactic parse trees which nodes are playing the role/function of logical subject for instance. An user of these parse trees needs to develop extra procedures to detect the roles played by various words or phrases. The task of function tagging, the problem addressed in this paper, is to add function tags to nodes in a parse tree.

4. The Model

We modeled the problem of assigning function tags as a classification problem. Classifiers are programs that assign a class from a predefined set of classes to an instance based on the values of attributes used to describe the instance. We defined a set of linguistically motivated attributes/features based on which we characterized the instances.

Let us analyze the set of features and classes we used to build the classifiers. We used a set of features inspired from [2] that includes the following: label, parent's label, right sibling label, left sibling label, parent's head pos (part-of-speech), head's pos, grandparent's head's pos, parent's head, head. We did not use the alternative head's pos and alternative head (for prepositional phrases that would be the head of the prepositional object) as explicit features but rather modified the phrase head rules so that the same effect is captured in pos and head features, respectively.

The set of classes we used corresponds to the set of functional tags in Penn Treebank. The functional tags are grouped in four categories given in Table 1. The four categories were derived so that no two labels from same new category can be attached to the same bracket.

The above features and classes are used to derive Decision Trees classifiers. The next section describes the experiments we conducted to derive and evaluate the classifiers.

5. Experimental Setup And Results

We trained the classifiers on sections 1-21 from Wall Street Journal (WSJ) part of Penn Treebank and used section 23 to evaluate the classifiers. This split is standard in the syntactic parsing community [5]. The evaluation follows a gold standard approach in which the classifiers' output is automatically compared with the correct values, also called gold values.

The performance measures reported are accuracy and kappa statistic. The *accuracy* is defined as the number of correctly tagged instances divided by the number of attempted instances. *Kappa statistic* [13] measures the agreement between predicted and observed classes, while correcting for agreement that occurs by chance. Kappa can have values between -1 and 1 with values greater than 0.60 indicating substantial agreement and values greater than 0.80 showing almost perfect agreement. We also report precision, recall, and F-measure for the second experiment when we also consider the non labeled nodes (viewed as negative instances). Such measures are reported on positive instances (true positives are considered correct). *Precision* is the number of correct guesses of tags from one category over the total number of guesses (correct of incorrect) for tags from that category. *Recall* is the number of correct guesses of tags in some category over the actual number of instances that should have a tag from that category. The *F-measure* value is calculated based on Precision and Recall from the following formula:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{1}$$

To build the classifiers, we used the implementation of Decision Trees in WEKA. WEKA [13] is a comprehensive, open-source (GPL) machine learning and data mining toolkit written in Java. WEKA requires a lot of memory to build the models from large training sets, especially for Decision Trees. A regular machine with 2GB of memory is not sufficient even after the preprocessing steps aimed at reducing the size of data (see below details about preprocessing). We used an IBM AIX High Performance Computer (HPC) system with 64GB of RAM.

5.1. Data Collection.

To build Decision Trees based classifiers, one must collect training data. The data is a set of problem instances. Each instance consists of values for each of the defined features of the underlying model and the corresponding class, i.e. function tag in our case. Instances are automatically created from Penn Treebank parse trees by simply traversing those trees and for each node extracting the values for each feature and for the class attribute.

Since a node can have several tags there are two possible setups for our classification task. We can define a class as a composed tag of all possible combinations of function tags that can be assigned to a node. A single classifier is generated in this case that would assign one composed tag to a node, i.e. one or more individual functional tags at once. We do not use this setup in this work. It was previously

studied by Lintean and Rus [10]. Alternatively, we can try to build four separate classifiers, one for each of the four functional categories described earlier in the paper. Knowing that a node cannot have more that one tag from a given category, each classifier will be used to predict the functional tag from the corresponding category. We focus on this latter setup in this paper.

Some simplifications are necessary to make the task feasible. In those experiments punctuation was mapped to an unique tag PUNCT and traces were left unresolved and replaced with TRACE. Furthermore, two features, **parent's head** and **head**, require special attention. They have as values the words that represent the head word[2] for a given node in the parse tree. Due to lexical diversity, the two features have a very large set of different values, i.e. words. This leads to very large Decision Trees that cannot be handled by regular computers. We applied a set of transformations aimed at reducing the number of possible values for head-words. Due to space constraints we do not specify the transformations (see [10] for details). These transformations reduce the number of distinct values by almost a half for the head and parent's head features from 19,730 to 11,430 and from 14,973 to 8,402, respectively.

5.2. **Results.** We conducted two types of experiments. In both experiments, the training and test data was divided according to the function tag category (Grammatical, Form/Function, Topicalisation, Miscellaneous). An instance from Treebank that has a composed tag such as *LGS-TMP* would lead to one instance for the Grammatical and Function/Form categories each. We generate four different classifiers, one for each category.

In a first experiment, we considered constituents with functional tags. From each parse tree in Treebank only nodes with functional tags were used to generate training and testing instances. Number of instances for the test data set are given in the second column of Table 2. We obtained 118,483 training instances for Grammatical, 66,261 for Form/Function, 3,751 for Topicalization and 16,630 for Miscellaneous. In the second experiment, we considered all internal nodes in parse trees. We did not generate instances for leaf nodes corresponding to part-of-speech tags. Nodes without a functional tag were assigned the new NON-F value indicating no functional tag. A total of 827,193 training instances and 47,333 test instances were generated.

Table 2 presents the results for the first experiment while Table 3 shows the results for the second experiment. The figures represent results on the test data, i.e. section 23 from Treebank. Each table also includes results for a baseline approach. The baseline approach always assigns the most frequent tag from a given category. For instance, for the Grammatical category the SBJ tag is the

[2]The head of a node in a syntactic parse tree is the word that gives most of the meaning of the phrase represented by that node. There is a set of deterministic rules to detect the head word of syntactic phrases [11].

TABLE 2. Performance Measures on Decision Trees (Experiment 1).

Category	# Instances	Errors	Acc./Baseline Acc.	Kappa
Grammatical	6907	21	99.70/81.79%	0.9901
Form/Function	3902	639	83.62/35.93%	0.7841
Topicalisation	261	0	100.00/100.00%	1.0000
Miscellaneous	755	4	99.47/84.45%	0.9807

TABLE 3. Performance Measures on Decision Trees (Experiment 2).

Category	Accuracy	Tag	Precision	Recall	F-Measure	Kappa
Baseline	never tag	always choose most likely tag				
Grammatical	86.93%	SBJ	10.53%	80.63%	18.63%	0
Form/Function	91.79%	TMP	3.10%	37.79%	5.74%	0
Topicalisation	99.41%	TPC	0.59%	100.00%	1.18%	0
Miscellaneous	98.44%	CLR	1.31%	84.21%	2.59%	0
Decision Tree results						
Grammatical	98.45%	-	99.19%	90.14%	94.45%	0.9370
Form/Function	95.15%	-	74.19%	52.54%	61.52%	0.6405
Topicalisation	99.87%	-	86.80%	90.40%	88.56%	0.8849
Miscellaneous	98.54%	-	61.75%	23.19%	33.72%	0.3318

most frequent and thus the baseline approach always assigns this tag. For the first experiment, the baseline performance is shown as the second value in the fourth column *Acc./Baseline Acc.* For the second experiment, since accuracy is computed on both positive and negative instances the most frequent tag is the newly introduced NON-F label that indicates no function tag.

From the tables we noticed high values for Kappa which suggest that Decision Trees offer predictions that are in high agreement with the true, gold values.

The results of Experiment 1 in Table 2 are better than the results of Experiment 1 reported in [10]. Only results of Experiment 1 in this work are directly comparable with results in [10].

6. Conclusions

We presented in this paper successful experiments with building Decision Trees from large data sets. The paper shows how good Decision Trees are at predicting function tags when trained on the whole Treebank data set. The proposed methods significantly outperform a baseline approach. We plan to expand our research to explore the feasibility of building one single Decision Tree that would assign all function tags at once. A set of preprocessing step and a re-engineering of the feature set may be necessary for that.

References

[1] M; Katz K Bies, A; Ferguson and R MacIntyre. Bracketing guidelines for treebank ii style. Penn Treebank Project, 1995.

[2] D Blaheta and M Johnson. Assigning function tags to parsed text. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 234–240, Seattle, May 2000.

[3] Don Blaheta. *Function tagging*. PhD thesis, Brown University, August 2003. Advisor-Eugene Charniak.

[4] J; Olshen R Breiman, L; Friedman and C Stone. A comparative analysis of methods for prunning decision trees. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 19(5):476–491, 1997.

[5] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of North American Chapter of Association for Computational Linguistics (NAACL-2000)*, Seattle, WA, April 29 - May 3 2000.

[6] M Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 1997.

[7] V Jijkoun and M De Rijke. Enriching the output of a parser using memory-based learning. In *Proceedings of the ACL 2004*, 2004.

[8] M Johnson. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[9] D. Lin. Dependency-based evaluation of minipar. In *Proceedings of Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May 1998.

[10] M. Lintean and V. Rus. Naive bayes and decision trees for function tagging. In *Proceedings of the International Conference of the FLAIRS-2007*, Key West, FL, May 2007 (in press).

[11] D.M. Magerman. *Natural Language Parsing as Statistical Pattern Recognition*. PhD thesis, Stanford University, February 1994.

[12] J R Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[13] E Witten, I ; Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann Publishers, 2005.

[1] Department of Computer Science, Institute for Intelligent Systems, FedEx Institute of Technology, The University of Memphis, Memphis, TN 38152, USA
E-mail address: M.Lintean@memphis.edu

[2] Department of Computer Science, Institute for Intelligent Systems, FedEx Institute of Technology, The University of Memphis, Memphis, TN 38152, USA
E-mail address: vrus@memphis.edu

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 33–40

TEXT ENTAILMENT VERIFICATION WITH TEXT SIMILARITIES

DOINA TĂTAR[(1)], GABRIELA ŞERBAN[(2)], AND MIHAIELA LUPEA[(3)]

Abstract. This paper presents a new method for recognizing the text entailment obtained from the text-to-text metric introduced in [3] and from the modified resolution introduced in [12]. In [11], using the directional measure of similarity as presented in [3], which measures the semantic similarity of a text T_1 with respect to a text T_2, some conditions of text entailment are established.

In this paper we present a method based on the results presented in [12] and [11], method which supposes the word sense disambiguation of the two texts T_1 and T_2 (text and hypothesis) and adds some appropriate heuristics. The algorithm is applied to a part of the set of pairs (text-hypothesis) contained in PASCAL RTE-2 data [16].

1. Text entailment verification by logical methods

Establishing entailment relationship between two texts is one of the most complex tasks in Natural Language Understanding. Thus, a very important problem in some computational linguistic applications (as question answering, summarization, information retrieval, and others) is to establish if a text *follows* from another text. The progress on this task is the key to many Natural Language Processing applications. Although the problem is not new, most of the automatic approaches have been proposed only recently, in the framework of RTE challenges events. (This year the on-line contest Pascal RTE Challenge is at the third edition.)

Let us denote the entailment relation between a text T_1 and a text T_2 as $T_1 \Rightarrow T_2$. The implemented methods of different teams participating at RTE events cover domains as Machine Learning ([6], [7]), semantic graphs ([7]), logical form ([9]), theorem proving ([2]) and others.

It is well known that a linguistic text can be represented by a set of logical formulas, called logic forms. From a logical point of view, proving a textual entailment consists of showing that a logical formula is deducible from a set of others formulas. This is a classical (semidecidable) problem in logics. Unfortunately, few

2000 *Mathematics Subject Classification.* 68T50,03H65.

©2007 Babeş-Bolyai University, Cluj-Napoca

sentences can be accurately translated to logical formulas. This is a reason that "pure" logical methods fail to obtain satisfactory results.

In [12] is proposed a new method to solve the problem of establishing if $T_1 \Rightarrow T_2$, a method obtained from the classical resolution refutation method, completing the unification of two atoms with some linguistic considerations. It is used here the method of obtaining logical forms (in fact, logical formulas) from sentences expressed in natural language proposed by [10] . In this method each *open-class* word in a sentence (that means: noun, verb, adjective, adverb) is transformed in a logic predicate (atom). Unification lexical method of two atoms proposed in [12] supposes the use of a lexical knowledge base (as, for example, WordNet) where the similarity between two words is quantified. In the algorithm of lexical unification we consider that $sim(p, p')$ between two words p, p' is that obtained by the Word::similarity interface [8], an on-line interface which calculates the similarity between two words using some different similarity measure. The similarity between two words is used to calculate a score for unifiability of two atoms. The test of quality of modified resolution is that the score is larger than a threshold τ.

The steps of demonstrating by resolution (refutation) that a text T_1 entails the text T_2 with the weight τ consist in:

- translating T_1 into a set of logical formulas T_1' and T_2 into T_2';
- considering the set of formulas $T_1' \cup negT_2'$, where by $negT_2'$ we means the logical negation of formulas T_2';
- finding the set C of disjunctive clauses obtained from the set of formulas T_1' and $negT_2'$;
- verifying if the set C is lexical contradictory with the weight τ'. If $\tau' \geq \tau$ then the text T_1 entails the text T_2.

2. Semantic similarity of texts

In [3] the authors introduce a method that combines word-to-word similarity metrics into a text-to-text metric measure, which indicates the semantic similarity of a text T_1 with respect to a text T_2. For a given pair of texts, they start by creating separated sets of open-class words for nouns, verbs, adjective and adverbs.

The authors define in [3] the similarity between the texts T_1 and T_2 with respect to T_1 as:

$$sim(T_1, T_2)_{T_1} = \frac{\sum_{pos}(\sum_{w_k \in WS_{pos}^{T_1}}(maxSim(w_k) \times idf_{w_k}))}{\sum_{pos}\sum_{w_k \in WS_{pos}^{T_1}} idf_{w_k}} \quad (1)$$

Here the sets of open-class words in each text segment are denoted by $WS_{pos}^{T_1}$ and $WS_{pos}^{T_2}$. The highest similarity of a word w_k with a given *pos* in T_1 with the words of the same *pos* in the other text T_2 is denoted by $maxSim(w_k)$.

This measure, which has a value between 0 and 1, is a measure of the directional similarity, in this case computed with respect to T_1. The authors experiment this

measure of text similarity using as measure of word similarity that of Wu and Palmer. This similarity metric measures the depth of the two concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score: $Sim_{wup} = \frac{2 \times depth(LCS)}{depth(concept1)+depth(concept2)}$.

2.1. **Text entailment verification using similarity of texts.** In this paper we use a simplified definition of similarity of the words. Namely, the single case of similarity is that of identity (which is a symmetric relation) and/or that of the occurrence of a word from a text in the synset of a word in other text (which is not a symmetric relation).

Starting with the measure of text semantic similarity, the textual entailment $T_1 \Rightarrow T_2$ can be derived based on the following theorem, established in the paper [11]. We denoted here by M_{T_1} the set of words from T_1 such that each of them is of maximal similarity with a word in T_2 and by M_{T_2} the set of words of T_2 such that each of them is of maximal similarity with a word in T_1. With these notations the theorem is:

Theorem

$T_1 \Rightarrow T_2$ if the following conditions hold:

$$sim(T_1, T_2)_{T_1} \leq sim(T_2, T_1)_{T_2} \quad (2)$$

$$M_{T_2} \subset M_{T_1} \quad (3)$$

This theorem reduces the verification of entailment relation $T_1 \Rightarrow T_2$ to the calculus of $sim(T_1, T_2)_{T_1}$ and $sim(T_2, T_1)_{T_2}$. The proof is given using the definition of *the demonstration by modified resolution* introduced in [12]. The atom corresponding to the word with a given *pos* in T_1, which has the highest similarity with a word w_k of the same *pos* in the other text T_2 (denoted in (1) by $maxSim(w_k)$), is the most "plausible" atom selected in the modified resolution process. This keeps the quality of the unification in a resolution step high, as this quality depends on the similarity of the two atoms which combine in this step [12].

In order to apply formulas (2) and (3) in our simplified version of similarity of words, we define two sets of words $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$ as follows:

$SYN(T_1)_{T_2}$ = *the set of nouns in* T_1 *such that they are contained in a synset of disambiguated nouns in* T_2 $\cup$ *the set of nouns in* T_1 *which are contained in* $T_2 \cup$ *the set of verbs in* T_1 *such that they are contained in a synset of disambiguated verbs in* T_2 $\cup$ *the set of verbs in* T_1 *which are contained in* T_2. Analogously is defined $SYN(T_2)_{T_1}$.

The value denoted in (1) as $sim(T_i, T_j)_{T_i}$ is $C_1 =\mid SYN(T_1)_{T_2} \mid$ and the value $sim(T_j, T_i)_{T_j}$ is $C_2 =\mid SYN(T_2)_{T_1} \mid$.

Formulas 2 and 3 in these new forms are verified for texts disambiguated by CHAD algorithm of word sense disambiguation [13]. So, in the formula denoted

by 1, we select pos=noun, pos=verb and we define the similarity between two words as 1, if the words are equal or they are situated in the same synset, and as 0 otherwise. In this way we identify (or "align" in the terms of [7]) the words that have the same part of speech and either words are identical, or they belong to the same synset in WordNet.

This identification is completed with a set of heuristics for recognizing false entailment. The false entailment occurs because of lack of monotone character of real texts. Monotonicity supposes that if a a text entails another text, then adding more text to the first, the entailment relation still holds [7].

The heuristics are represented by the bellow condition COND posed in a fixed situation of T_2.

1. $not \in T_1$ and not $not \in T_2$.

In this case the entailment relation does not hold.

For this particular case, $T_2 = NP_2 \cup I_c$, we check if a modal verb is in the following situations:

2. $can \in T_1$ and not $can \in T_2$ (the heuristics shows that "possibility does not entail actuality");
3. $might \in T_1$ and not $might \in T_2$;
4. $should \in T_1$ and not $should \in T_2$;
5. $before \in T_1$ and $after \in T_2$ or $before \in T_2$ and $after \in T_1$;
6. $over \in T_1$ and $under \in T_2$ or $over \in T_2$ and $under \in T_1$.

In all these cases the entailment relation does not hold.

For description of our algorithm, let us make the following notations:

- Named entities in T_1= NE_1 (here we count quantity and time in T_1)
- Named entities in T_2= NE_2 (here we count quantity and time in T_2)
- I_c = non-named entities common in T_1 and T_2

- $SYN(T_1)_{T_2}$ ={words non-NE, non common, in T_1, which are nouns or verbs, and are contained in a synset of T_2} $\cup(NE_1 \cap NE_2) \cup I_c = M_1 \cup (NE_1 \cap NE_2) \cup I_c$
- $SYN(T_2)_{T_1}$ ={words non-NE, non common, in T_2, which are nouns or verbs, and are contained in a synset of T_1} $\cup(NE_1 \cap NE_2) \cup I_c = M_2 \cup (NE_1 \cap NE_2) \cup I_c$
- $C_1 =\mid SYN(T_1)_{T_2} \mid$
- $C_2 =\mid SYN(T_2)_{T_1} \mid$
- $W_{T_1} = NE_1 \cup I_c$ (the named entities and the common words from T_1 are the set of words from T_1 such that each of them is of maximal similarity with a word in T_2, such that $W_{T_1} = M_{T_1}$ in (3))
- $W_{T_2} = NE_2 \cup I_c$ (also = M_{T_2})

The conditions for text entailment obtained from 2 and 3 are:

- a) $C_1 \leq C_2$ (that means $\mid M_1 \mid \leq \mid M_2 \mid$)
- b) $W_{T_2} \subset W_{T_1}$ (that means $NE_2 \subset NE_1$)

For our heuristics an important situation is that T_2 contains only named entities and words also contained in T_1 . In this respect, condition b) is verified first.

Algorithm

if $W_{T_2} \subset W_{T_1}$ /* that means $NE_2 \subset NE_1$
 then
  if $T_2 = NE_2 \cup I_c$
   then
    if COND
     then
      not $(T_1 \Longrightarrow T_2)$
     else
      $T_1 \Longrightarrow T_2$ (case I)
    endif
   else
    if $C_1 \leq C_2$
     then
      $T_1 \Longrightarrow T_2$ (case II)
     else
      not $(T_1 \Longrightarrow T_2)$
    endif
  endif
 else
  not ($T_1 \Longrightarrow T_2$)
endif

For example, if the disambiguated (all nouns are associated with a WordNet synset) texts are:

$$T_1 = w_1\{synset_1\}\, w_2\{synset_2\}\, w_3\{synset_2\}$$

and

$$T_2 = w_4\{synset_2\}\, w_5\{synset_3\}\, w_2\{synset_4\}$$

and if $\{synset_2\} = \{w_2, w_3, w_4\}$ then $\mid SYN(T_1)_{T_2} \mid = \mid \{w_2, w_3, w_2\} \mid = 3$ and $\mid SYN(T_2)_{T_1} \mid = \mid \{w_4, w_2\} \mid = 2$

The conditions 2 and 3 in the above theorem say that, for our example, relation $T_1 \Rightarrow T_2$ does not hold.

2.2. **Implementation and experiments.** The application is written in JDK 1.5.0.and uses *HttpUnit* 1.6.2 API [14]. Written in Java, HttpUnit is a free software that emulates the relevant portions of browser behavior, including form submission, JavaScript, basic http authentication, cookies and automatic page redirection, and allows Java test code to examine returned pages as text, containers of forms, tables, and links [14]. We have used *HttpUnit* in order to search WordNet through the dictionary from [15]. More specifically, *WebConversation*, *WebResponse* and *WebForm* classes from [13] are used. *WebConversation* is used in order to emulate the browser behavior needed to build the test of the web site from [15].

WebResponse class is used in order to obtain the response to a web request from a web server and *WebForm* class is used in order to simulate the submission of a form.

In our system the preprocessing step consists in POS-tagging text and named entity recognizing. The necessary disambiguation for calculating sets $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$ is realized using our CHAD algorithm of disambiguation, based on WordNet [13].

We present the results obtained when the system is applied to a set of 35 pairs (text-hypothesis) from the data set of Pascal RTE-2 Challenge. The data set is balanced to contain equal numbers of *yes* and *no*. Additionally, we considered a set of 7 pairs corresponding to the cases 1 to 6 in condition COND and to the situation *not* ($W_{T_2} \subset W_{T_1}$), which is not illustrated in this data set. The result was of 25 correct evaluations, which corresponds to an accuracy of 71,4%. Remark that the participants in the first Pascal RTE workshop reported accuracy from 49,5% to 58,6%, and in the second Pascal RTE from 50,8% to 75,3%

For example, the pair text-hypothesis:

< pair id="27" entailment="YES" task="IE" >

< t > Responding to Scheuer's comments in La Repubblica, the prime minister's office said the analysts' allegations, "beyond being false, are also absolutely incompatible with the contents of the conversation between Prime Minister Silvio Berlusconi and U.S. Ambassador to Rome Mel Sembler." < /t >

< h >Mel Sembler represents the U.S.< /h > < /pair>.

has as output of POS-tagger and NER the following:

< t >Responding/V to P1/NNP comments/N in P2/NNP, the P3/NNP office/N said/V the analysts'/N allegations/N, "beyond being/V false, are/V also absolutely incompatible with the contents/N of the conversation/N between P3/NNP P4/NNP P5/NNP and P6/NNP P7/NNP to P8/NNP P9/NNP P10/NNP".< /t >

< h >P9/NNP P10/NNP represents/V the P6/NNP.< /h >

The output of algorithm is "YES" (case II).

As another example,

<pair id="84" entailment="YES" task="IE">

$< t >$Salvadoran reporter Mauricio Pineda, a sound technician for the local canal Doce television station, was shot and killed today in Morazan department in the eastern part of the country.$< /t >$

$< h >$Mauricio Pineda was killed in Morazan.$< /h >$ $< /$pair$>$
has the corresponding output

$< t >$P1/NNP reporter/N P2/NNP P3/NNP, a sound/N technician/N for the local canal/N P4/NNP television/N station/N, was shot/V and killed/V today/N in P5/NNP department/N in the eastern part/N of the country/N.$< /t >$

$< h >$P2/NNP P3/NNP was killed/V in P5/NNP.$< /h >$
and the algorithm output is "YES" (case I).

3. Conclusions and further work

There are some issues which impose big limitations to each text entailment system. One of this is the lack of monotonicity of texts in a natural language. The impact of syntactic features is usually positive. We intend to add to our system a part of shallow syntactical analysis and to establish some syntactic heuristics. This will be an advantage especially for recognizing false entailment. For example, at this stage, our system sets a decision "Yes" to the false entailment 1971 from RTE-1:

T_1: "U.N. officials are dismayed that Aristide killed a conference called by Prime Minister Robert Malval" ;

T_2: "Aristide kills Prime Minister Robert Malval".

An analysis of different objects of verb "kill" in each sentence will recognize the false entailment. Also, the recognition and analysis of if-clauses will reject as false entailment some other examples.

We intend also to use a more complex similarity between the words from a pair text-hypothesis T_1, T_2. For example the lexical chain built using WordNet between a verb from T_1 and a verb from T_2 [1] could indicate, for different thresholds, different relations between these: the smaller values mean closer relationships, 0 being the distance between members of the same synset.

Much work remains in recognition of nonmonotonicity effects, by creating additional heuristics to deal with specific patterns.

References

[1] A. Andreevskaia, Z. Li, S. Bergler: "Can shallow predicate argument structures determine entailment?", Proceedings of the First Pascal Recognizing Textual Entailment Challenge, 2005.

[2] J.Bos, K. Markert: "Recognising Textual Entailment with logical inference", Proceedings of HLT/EMNLP. Vancouver, October 2005, pages 628-635.

[3] C. Corley, R. Mihalcea: "Measuring the semantic similarity of texts", Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, June, 2005, pages 13-18.

[4] I. Dagan, O. Glickman and B. Magnini: The PASCAL Recognising Textual Entailment Challenge, Proceedings of the PASCAL Work- shop, 2005.
[5] A. Haghighi, A. Ng, C. Manning: "Robust textual Inference via graph matching", Proceedings of HLT/EMNLP. Vancouver, October 2005, pages 387-394.
[6] D. Inkpeen,D. Kipp, V. Năstase: "Machine Learning Experiments for textual entailment", in Proc. of the Second PASCAL Challenges on Recognising Textual Entailment, Venice, Italy, 2006.
[7] B. MacCartney, T. Grenager, M. de Marneffe, D. Cer, C. Manning: "Lerning to recognize features of valid textual entaiments", Proceedings of the HLT Conference of the North American Chapter of the ACL, peges 41-48, New York, June 2006.
[8] T. Pedersen, S. Patwardhan, and J. Micheelizzi: "Wordnet::similariry-measuring the relatedness of concepts, Proc. of 5th NAACL, Boston, MA, 2004
[9] R. Raina, A. Ng, C. Manning: "Robust textual inference via learning and abductive reasoning", AAAI, Proceedings of the Twentieth National Conference on AI, 2005.
[10] V. Rus: "Logic form transformation for WordNet glosses and its applications". PhD Thesis, Southern Methodist University, CS and Engineering Department, March 2001.
[11] D. Tătar, C. Corley and R. Mihalcea: "Text entailment and semantic similarity of texts.", accepted at Data Mining and Information Engineering, The New Forest, UK, 18 - 20 June, 2007 .
[12] D. Tătar, M. Frenţiu: "Textual inference by theorem proving and linguistic approach", Studia Universitatis "Babeş- Bolyai", Seria Informatics, 2006, nr 2, pages 31-41.
[13] D. Tatar, G. Serban, A. Mihis, M. Lupea, D. Lupsa and M. Frentiu: "Chain algorithm for WSD", submitted to ACL 2007.
[14] http://httpunit.sourceforge.net/, 2006.
[15] http://wordnet.princeton.edu/perl/webwn .
[16] http://ai-nlp.info.uniroma2.it/te/datasets/ RTE/RTE2/

(1) Department of Computer Science, University "Babes-Bolyai" Cluj-Napoca
E-mail address: dtatar@cs.ubbcluj.ro

(2) Department of Computer Science, University "Babes-Bolyai", Cluj-Napoca
E-mail address: gabis@cs.ubbcluj.ro

(3) Department of Computer Science, University "Babes-Bolyai" Cluj-Napoca
E-mail address: lupea@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 41–49

A CHAIN DICTIONARY METHOD FOR WORD SENSE DISAMBIGUATION AND APPLICATIONS

DOINA TĂTAR(1), GABRIELA ŞERBAN(1), ANDREEA MIHIŞ(1), MIHAIELA LUPEA(1), DANA LUPŞA(1), AND MILITON FRENŢIU(1)

Abstract. A large class of unsupervised algorithms for Word Sense Disambiguation (WSD) is that of dictionary-based methods. Various algorithms have as the root Lesk's algorithm, which exploits the sense definitions in the dictionary directly. Our approach uses the lexical base WordNet [3] for a new algorithm originated in Lesk's, namely *chain algorithm for disambiguation* of all words (CHAD). We show how translation from a language into another one and also text entailment verification could be accomplished by this disambiguation.

1. The polysemy

Word sense disambiguation is the process of identifying the correct sense of words in particular contexts. The solving of WSD seems to be AI complete (that means its solution requires a solution to all the general AI problems of representing and reasoning about arbitrary) and it is one of the most important open problems in NLP [5],[6],[7], [10],[12],[13]. In the electronical on-line dictionary WordNet, the most well-developed and widely used lexical database for English, the polysemy of different category of words is presented in order as: the highest for verbs, then for nouns, and the lowest for adjectives and adverbs. Usually, the process of disambiguation is realized for a single, target word. One would expect the words closest to the target word to be of greater semantical importance for it than the other words in the text. The context is hence a source of information to identify the meaning of the polysemous words. The contexts may be used in two ways: a) as *bag of words*, without consideration of relationships with the target word in terms of distance, grammatical relations, etc.; b) with relational information. The *bag of words* approach works better for nouns than verbs but is less effective than methods that take other relations in consideration. Studies about syntactic relations determined some interesting conclusions: verbs derive more disambiguation

2000 *Mathematics Subject Classification.* 68T50,03H65.
Key words and phrases. WSD, machine translation, text entailment.

©2007 Babeş-Bolyai University, Cluj-Napoca

information from their objects than from their subjects, adjectives derive almost all disambiguation information from the nouns they modify, and nouns are best disambiguated by directly adjacent adjectives or nouns [5]. All these advocate that a global approach (disambiguation of all words) helps to disambiguate each POS.

In this paper we propose a global disambiguation algorithm called **chain algorithm** for disambiguation, CHAD, which presents elements of both points of view about a context: because this algorithm is *order sensitive* it belongs to the class of algorithms which depend of relational information; in the same time it doesn't require syntactic analysis and syntactic parsing.

In section 2 of this paper we review Lesk's algorithm for WSD. In section 3 we present "triplet" algorithm for three words and CHAD algorithm. In section 4 we describe some experiments and evaluations with CHAD. Section 5 introduces some conclusions of using the CHAD for translation (here from Romanian language to English) and for text entailment verification. Section 6 draws some conclusions and further work.

2. Dictionary-based methods

Work in WSD reached a turning point in the 1980s when large-scale lexical resources, such as machine readable dictionaries, became widely available. One of the best known dictionary-based method is that of Lesk (1986). It starts from the idea that a word's dictionary definition is a good indicator for the senses of this word and uses the definition in the dictionary directly.

Let us remember basic algorithm of Lesk [8]:

Suppose that for a polysemic target word w there are in a dictionary Ns senses $s_1, s_2, \cdots, s_{Ns}$ given in an equal number of definitions $D_1, D_2, \cdots, D_{Ns}$. Here we mean by D_i the set of words contained in the i-th definition.

Consider that the new context to be disambiguated is c_{new}. The **reduced form** of Lesk's algorithm is:

for $k = 1, Ns$ *do*
  $score(s_k) =\mid D_k \cap (\cup_{v_j \in c_{new}} \{v_j\}) \mid$
endfor
Calculate $s' = argmax_k score(s_k)$

The score of a sense is the number of words that are shared by the different sense definitions (glosses) and the context. A target word is assigned that sense whose gloss shares the largest number of words.

The algorithm of Lesk was successfully developed in [2] by using WordNet dictionary for English. It was created by hand in 1990s and includes definitions (glosses) for individual senses of words, as in a dictionary. Additionally it defines groups of synonymous words representing the same lexical concept (synset) and organizes them into a conceptual hierarchy. The paper [2] uses this conceptual hierarchy

for improving the original Lesk's method by augmenting the definitions with non-gloss information: synonyms, examples and glosses of related words (hypernyms, hyponyms). Also, the authors introduced a novel overlap measure between glosses which favorites multi-word matching.

3. Chain algorithm for word sense disambiguation - CHAD.

First of all we present an algorithm for disambiguation of a triplet. In a sense, our triplet algorithm is similar with global disambiguation algorithm for a window of two words around a target word given [2]. Instead, our CHAD realizes disambiguation of all-words in a text with any length, ignoring the notion of "window" and "target word" and target word in similar studies, all that without increasing the computational complexity.

The algorithm for disambiguation of a triplet of words $w_1w_2w_3$ for Dice measure is the following:

begin
  for each sense $s^i_{w_1}$ do
    for each sense $s^j_{w_2}$ do
      for each sense $s^k_{w_3}$ do
        $score(i,j,k) = 3 \times \frac{|D_{w_1} \cap D_{w_2} \cap D_{w_3}|}{|D_{w_1}|+|D_{w_2}|+|D_{w_3}|}$
      endfor
    endfor
  endfor
  $(i^*, j^*, k^*) = argmax_{(i,j,k)} score(i,j,k)$ /* sense of w_1 is $s^{i^*}_{w_1}$, sense of w_2 is $s^{j^*}_{w_2}$, sense of w_3 is $s^{k^*}_{w_3}$ */
end

For the overlap measure the score is calculated as: $score(i,j,k) = \frac{|D_{w_1} \cap D_{w_2} \cap D_{w_3}|}{min(|D_{w_1}|,|D_{w_2}|,|D_{w_3}|)}$

For the Jaccard measure the score is calculates as: $score(i,j,k) = \frac{|D_{w_1} \cap D_{w_2} \cap D_{w_3}|}{|D_{w_1} \cup D_{w_2} \cup D_{w_3}|}$

Shortly, CHAD begins with the disambiguation of a triplet $w_1w_2w_3$ and then adds to the right the following word to be disambiguated. Hence it disambiguates at a time a new triplet, where first two words are already associated with the best senses and the disambiguation of the third word depends on these first two words. CHAD algorithm for disambiguation of the sentence $w_1w_2...w_N$ is:

begin
  Disambiguate triplet $w_1w_2w_3$
  $i = 4$
  while $i \leq N$ do
    Calculate $score(s_i) = 3 \times \frac{|D^*_{w_{i-2}} \cap D^*_{w_{i-1}} \cap D^{s_i}_{w_i}|}{|D^*_{w_{i-2}}|+|D^*_{w_{i-1}}|+|D^{s_i}_{w_i}|}$
    Calculate $s^*_i := argmax_{s_i} score(s_i)$
    $i := i + 1$
  endwhile
end

Due to the brevity of definitions in WN many values of $| D^*_{w_{i-2}} \cap D^*_{w_{i-1}} \cap D^{s_i}_{w_i} |$ are 0. We attributed the first sense in WN for s^*_i in this cases.

4. Some experiments with chain algorithm. Experimental evaluation of **CHAD**

In this section we shortly describe some experiments that we have made in order to validate the proposed chain algorithm **CHAD**.

4.1. **Implementation details.** We have developed an application that implements **CHAD** and can be used to:

- disambiguate words (4.2);
- translate words into Romanian language (5.1);
- text entailment verification (5.2).

The application is written in JDK 1.5.0. and uses *HttpUnit* 1.6.2 API [15]. Written in Java, HttpUnit is a free software that emulates the relevant portions of browser behavior, including form submission, JavaScript, basic http authentication, cookies and automatic page redirection, and allows Java test code to examine returned pages either as text, an XML DOM, or containers of forms, tables, and links [15].

We have used *HttpUnit* in order to search WordNet through the dictionary from [16]. More specifically, the following Java classes from [15] are used:

- *WebConversation*. It represents the context for a series of HTTP requests. This class manages cookies used to maintain session context, computes relative URLs, and generally emulates the browser behavior needed to build an automated test of a web site.
- *WebResponse*. This class represents a response to a web request from a web server.
- *WebForm*. This class represents a form in an HTML page. Using this class we can examine the parameters defined for the form, the structure of the form (as a DOM), and the text of the form. We have used *WebForm* class in order to simulate the submission of the form with corresponding parameters.

4.2. **Results.** We tested our CHAD on 10 files of Brown corpus, which are POS tagged. Recall that WN stores only stems of words. So, we first preprocessed the glosses and the input files, replacing inflected words with their stems.

The reason for choosing Brown corpus was the possibility offered by SemCor corpus (the best known publicly available corpus hand tagged with WN senses) to evaluate the results. The correct disambiguated words means the disambiguated words as in SemCor. We ran separately CHAD for: 1. nouns, 2. verbs, and 3. nouns, verbs, adjectives and adverbs. In the case of CHAD addressed to nouns, the output is the sequence of nouns tagged with senses. The tag *noun#n#i*

means that for noun *noun* the WN sense i was found. Analogously for the case of disambiguation on verbs and of all POS. The results are presented in tables 1 and 2. As our CHAD algorithm is dependent on the length of glosses, and as nouns have the longest glosses, the highest precision is obtained for nouns. In Figure 3, the Precision Progress can be traced. By dropping and rising, the precision finally stabilizes to value 0.767 (for the file Br-a01). The most interesting part of this graph is that he shows how this Chain Algorithm works and how the correct or incorrect disambiguation of first two words from the first triplet influences the disambiguation of the next words.

It is known that, at Senseval 2 contest, only 2 out of the 7 teams (with the unsupervised methods) achieved higher precision than the WordNet 1^{st} sense baseline. We compared in figures 1 , 2 and 3 the precision of CHAD for 10 files in Brown corpus, for Dice, Overlap and Jaccard measures with WordNet 1^{st} sense.

Comparing the precision obtained with the Overlap Measure and the precision given by the WordNet 1^{st} sense for 10 files of Brown corpus (Br-a01, Br-a02, Br-11, Br-12, Br-13, Br-14, Br-a15, Br-b13, Br-b20 and Br-c01), we obtained the following results:

- for Nouns, the minimum difference was 0.0077, the maximum difference was 0.0706, the average difference was 0.0338;
- as a whole, for 4 files difference was greater or equal to 0.04, and for 6 files was lower;
- in case of all Parts of Speech, the minimum difference was 0.0313, the maximum difference was 0.0681, the average difference was 0.0491;
- as a whole, for 7 files difference was greater or equal to 0.04, and for 3 files was lower;
- relatively to Verbs, the minimum difference was 0.0078, the maximum difference was 0.0591, the average difference was 0.0340;
- as a whole, for 4 files difference was greater or equal to 0.04, and for 6 files was lower.

Let us remark that in our CHAD the standard concept of windows better size parameter [2] is not working: simply, a window is the variable space between the previous and the following word in respect to the current word.

5. Applications of CHAD algorithm

5.1. **Application to Romanian-English translation.** WSD is only an intermediate task in NLP. In Machine Translation WSD is required for lexical choise for words that have different translation for different senses and that are potentially ambiguous within a given document. However, most Machine Translation models do not use explicit WSD [1] (in Introduction). The algorithm implemented by us consists in the translation word by word of a Romanian text (using dictionary at http://lit.csci.unt.edu/≈rada/downloads/RoNLP/R.E. tralexand), then the application of chain algorithm to the English text. As the translation of a

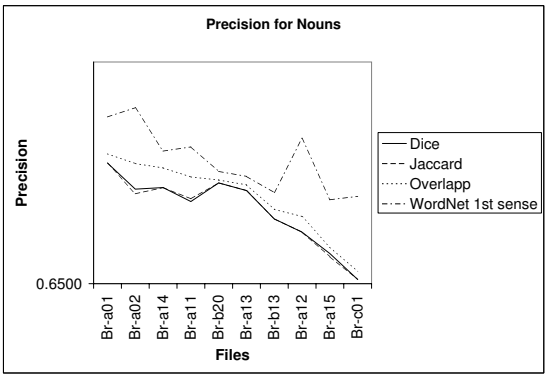


FIGURE 1. Noun Precision

File	Words	Dice	Jaccard	Overlap	WN1
Bra01	486	0.758	0.758	0.767	0.800
Bra02	479	0.735	0.731	0.758	0.808
Bra14	401	0.736	0.736	0.754	0.769
Bra11	413	0.724	0.726	0.746	0.773
Brb20	394	0.740	0.740	0.743	0.751
Bra13	399	0.734	0.734	0.739	0.746
Brb13	467	0.708	0.708	0.717	0.732
Bra12	433	0.696	0.696	0.710	0.781
Bra15	354	0.677	0.674	0.682	0.725
Brc01	434	0.653	0.653	0.661	0.728

TABLE 1. Precision for Nouns, sorted descending by the precision of Overlap measure

Romanian word in English is multiple, the disambiguation of a triplet is modified as following. Let be the word w_1 with k_1 translations $t^m_{w_1}$, the word w_2 with k_2 translations $t^n_{w_2}$ and the word w_3 with k_3 translations $t^p_{w_3}$. Each triplet $t^m_{w_1} t^n_{w_2} t^p_{w_3}$ is disambiguated with the triplet disambiguation algorithm and then the triplet with the maxim score is selected:

begin
 for $m = 1, k_1$ do
  for $n = 1, k_2$ do
   for $p = 1, k_3$ do
    Disambiguate triplet $t^m_{w_1} t^n_{w_2} t^p_{w_3}$ in $(t^m_{w_1})^*(t^n_{w_2})^*(t^p_{w_3})^*$
    Calculate $score((t^m_{w_1})^*(t^n_{w_2})^*(t^p_{w_3})^*)$
   endfor
  endfor
 endfor

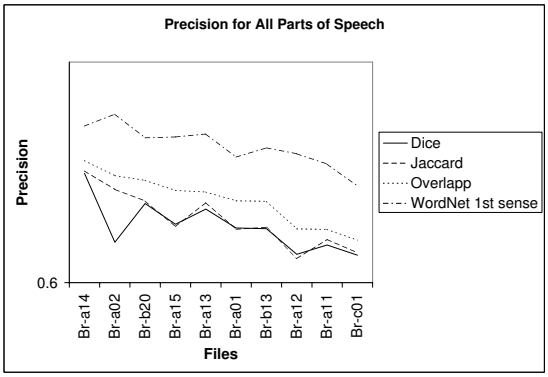


FIGURE 2. All Parts of Speech Precision

File	Words	Dice	Jaccard	Overlap	WN1
Bra14	931	0.699	0.701	0.711	0.742
Bra02	959	0.637	0.685	0.697	0.753
Brb20	930	0.672	0.674	0.693	0.731
Bra15	1071	0.653	0.651	0.684	0.732
Bra13	924	0.667	0.673	0.682	0.735
Bra01	1033	0.650	0.648	0.674	0.714
Brb13	947	0.649	0.650	0.674	0.722
Bra12	1163	0.626	0.622	0.649	0.717
Bra11	1043	0.634	0.639	0.648	0.708
Brc01	1100	0.625	0.627	0.638	0.688

TABLE 2. Precision for all POS, sorted descending by the precision of Overlap measure

Calculate $(m*, n*, p*) = argmax_{(m,n,p)} score((t^m_{w_1})^*(t^n_{w_2})^*(t^p_{w_3})^*)$
Optimal translation of triplet is $(t^{m*}_{w_1})^*(t^{n*}_{w_2})^*(t^{p*}_{w_3})^*$
end

Let us remark that $(t^{m*}_{w_1})^*$, for example, is a synset which corresponds to the best translation for w_1 produced by CHAD algorithm. However, since in Romanian are used many words linked by different spelling signs, these composed words are not found in the Romanian-English dictionary. Accordingly, not each Romanian word produces an English correspondent as output of the above algorithm. However, many translations are still correct. For example, the translation of expression *vreme trece* (in the poem "Glossa" of our national poet Mihai Eminescu), is *Word: (Rom)vreme (Eng)Age#n#4 , Word: (Rom)trece (Eng)Flow#v#1* . As another example from the same poem, where the synset of a word occurs (as an output of our application), *ţine toate minte*, is translated in *Word: (Rom) tine (Eng)*

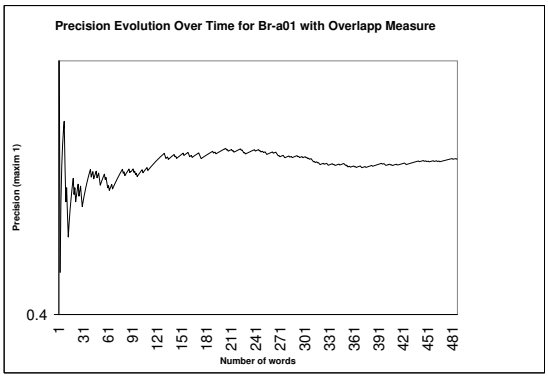


FIGURE 3. Precision in progress

Keep#v#8 :{keep, maintain}, Word: (Rom) toate (Eng) All#adv#3 :{wholly, entirely, completely, totally, all, altogether, whole}, Word: (Rom) minte (Eng) Judgment#n#2 :{judgment, judgement, assessment}.

5.2. **Application to text entailment verification.** The recognition of text entailment is one of the most complex task in Natural Language Understanding [14]. Thus, a very important problem in some computational linguistic applications (as question answering, summarization, segmentation of discourse, and others) is to establish if a text *follows* from another text. For example, a QA system has to identify texts that entail the expected answer. Similarly, in IR the concept denoted by a query expression should be entailed from relevant retrieved documents. In summarization, a redundant sentence should be entailed from other sentences in the summary. The application of WSD to text entailment verification is treated by authors in the paper "Text entailment verification with text similarity" in this Volume.

6. Conclusions and further work

In this paper we presented a new algorithm of word sense disambiguation. The algorithm is parametrized for: 1. all words (that means nouns, verbs, adjectives, adverbs); 2. all nouns; 3. all verbs. Some experiments with this algorithm for ten files of Brown corpus are presented in section 4.2. The stemming was realized using the list from http://snowball.tartarus.org/algorithms/porter/diffs.txt. The precision is calculated relative to the corresponding annotated files in SemCor corpus. Some details of implementation are given in 4.1.

We showed in section 5 how the disambiguation of a text helps in automated translation of a text from a language into another language: each word in the first text is translated into the most appropriated word in the second text. This

appropriateness is considered from two points of view: 1. the point of view of possible translation and 2. the point of view of the real sense (disambiguated sense) of the second text. Some experiments with Romanian - English translations and text entailment verification are given (section 5).

Another problem which we intend to address in the further work is that of optimization of a query in Information Retrieval. Finding whether a particular sense is connected with an instance of a word is likely the IR task of finding whether a document is relevant to a query. It is established that a good WSD program can improve performance of retrieval. As IR is used by millions of users, an average of some percentages of improvement could be seen as very significant.

References

[1] E. Agirre and P. Edmonds (editors). 2006. *WSD: Algorithms and Applications*. Springer.
[2] S. Banarjee and T. Pedersen. 2003. *Extended Gloss Overlaps as a Measure of Semantic Relatedness*. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, August 9-15, Acapulco, Mexico.
[3] C. Fellbaum (editor). 1998. *WordNet An Electronic Lexical Database*. The MIT Press.
[4] S. Harabagiu and D.Moldovan. 1999. *A parallel system for Textual Inference*. IEEE Transactions parallel and distributed systems, 10(11), 254–270.
[5] N. Ide and J. Veronis. 1998. *Introduction to the special issue on WSD: the state of the art*. Computational Linguistics, 24(1):1–40.
[6] D. Jurafsky and J. Martin. 2000. *Speech and language processing*. Prentice Hall.
[7] A. Kilgarriff. 1997. *What is WSD good for?* ITRI Technical Report Series- August.
[8] C. Manning and H. Schutze. 1999. *Foundation of statistical natural language processing*. MIT.
[9] T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. *Wordnet::Similarity-measuring the relatedness of concepts*. 1024–1025.
[10] P. Resnik and D. Yarowsky. 1998. *Distinguishing Systems and Distinguishing sense: new evaluation methods for WSD*. Natural Language Engineering, 1(1).
[11] V. Rus. 2001. *Logic form transformation for WordNet glosses and its applications*. PhD Thesis, Southern Methodist University, CS and Engineering Department.
[12] G. Serban and D. Tatar. 2004. *UBB system at Senseval3*. Proceedings of Workshop in Word Disambiguation, ACL 2004, Barcelona, July, 226–229.
[13] D. Tatar and G. Serban. 2001. *A new algorithm for WSD*. Studia Univ. Babes-Bolyai, Informatica, 2, 99–108.
[14] D. Tătar and M. Frenţiu. 2006. *Textual inference by theorem proving and linguistic approach*. Studia Universitatis Babes-Bolyai, Informatica, LI(2), 31–41.
[15] http://httpunit.sourceforge.net/, 2006.
[16] http://wordnet.princeton.edu/perl/webwn, 2006.

(1) Babes-Bolyai University, Cluj-Napoca

E-mail address: dtatar@cs.ubbcluj.ro, gabis@cs.ubbcluj.ro
E-mail address: mihis@cs.ubbcluj.ro, mlupea@cs.ubbcluj.ro
E-mail address: dlupsa@cs.ubbcluj.ro, mfrentiu@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 50–57

SYNTAGMA PROCESSING FOR INCOMPLETE ANSWERS

ADRIAN ONET[(1)]

Abstract. By trying to find a solution to incomplete answer processing, answers that are very frequent in a usual communication scenario based upon question-answer pattern, we developed an algorithm able to reconstruct the incomplete answer by using the question syntactical environment. Thus, one of the problem related to natural answers are the syntagmas. We call syntagma an incomplete answer that resumes to a phrase not to a grammatically correct sentence based on a subject and a verb. For example, if we consider the question "*What is your favorite color?*", most of the answers will be of the following form "green". Unfortunately, such an answer can't usually be processed by using an English grammar. In our SPEL (Syntactic Parser for English Language) system, we have introduced an algorithm that is able to reconstruct the answers from the given syntagma and the initial question, without affecting the semantic information given by the answer.

1. Introduction

In a usual communication scenario that necessarily involves a question-answer pattern the most common situation that we have to resolve is the syntagma answers. This means that all the incomplete answers that are received to a given number of questions must be reconstructed by using the specific syntactical structure of the question.

In order to eliminate the irrecognizability of this kind of incomplete sentences, we will present in this paper an algorithm for syntagma reconstruction by using the user's incomplete answer to a question and the respective question. The algorithm is capable to create an answer that is syntactically correct. It will consequently have a subject and a verb that respect the basic syntactical pattern. The presented algorithm is used as part of the SPEL system and it was tested on more than 40000 answers with promising result. However, the algorithm is not fully proved but it has a good rating of reconstructing the correct answer. Another benefit of this algorithm is the fast processing: it uses only some of the semantic and syntactic

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.

Key words and phrases. Syntagma Processing, Natural Language Processing, Incomplete Sentence Reconstruction.

©2007 Babeş-Bolyai University, Cluj-Napoca

information from the question and from the answer. First we will do a short introduction of the SPEL system that incorporated this algorithm, afterwards, we will present the necessary steps to implement the algorithm itself and its further improvements.

2. SPEL Description

The SPEL is designed to be able to syntactically parse English complete and incomplete sentences. The system is based on a DCG grammar and an extension of the Wordnet [1] dictionary. Before we go any further into the description of our system, we have to enhance the fact that there are many syntactic parsers in the literature, from these we can mention:

(1) AGFL [2] that is based on a two layer gramar;
(2) *Link Grammar Parser*, based on linked grammars [3];
(3) RASP (*Robust Accurate Statistical Parsing*) system, based on a statistical analysis of the lexical information [4];
(4) *Connexor* [5] which is also based on a statistical parsing model.

As opposed to these systems, SPEL is based on a strict grammar that is unable to recognize syntactically incorrect sentences or correct sentences that do not have their corresponding rules in its grammar. Its efficiency strictly depends on the grammar that we have built which allows us to detect the syntactical correctness of the processed sentences that are processed. However, despite these, our system presents the following advantages, by providing:

- the deep analysis of the syntactically correct sentences;
- the extensibility and the modularity of the grammar;
- the possibility of inserting semantic rules over the existing rules for a semantic parsing;
- a flexible adaptability of the grammar and the possibility of constructing a new system for automatic sentence translation.

The disadvantages of SPEL over the existing systems concern mostly, on one hand, the processing time and, on the other hand, the morphological, syntactical correctness of the words in the sentence. But some of these disadvantages were already considered to be incorporated and will be eliminated in future releases. As we already stipulated, one of the problem that SPEL may encounter is that the processing time for some complex sentences can be very long. This usually happens for sentences that contain polymorphic words, such as a verb which has the same form as the corresponding noun (*to work / work*) or an adjective and an -ing verb form (such as *interesting*). This problem occurs mainly because of the size of the dictionary. On the other hand, SPEL will not recognize syntactically incorrect sentences. That is sentences that do not respect the static rules from its DCG Grammar. This, in fact, ensures the fact that the system depends strongly on the syntactical correctness of the sentence. Also, the system doesn't accept

elliptic words or ortographically incorrect words. As the grammar is implemented in Prolog, the order of the rules is the order the sentence will be matched. Thus, the first match will be considered the desiderate parse. But this is not always the case if we consider the polymorphic words, where we can not decide, even by statistical choice, which morphological value of a word is to be considered first.

As an improvement for SPEL, we want to combine the existing grammar with the linked grammars, in such a way that it will also introduce semantic elements in the syntactic parsing. This will improve the system by choosing the morphologic occurrence of a polymorphic word (for example, the word "living" can be adjective, noun and -ing verb) that is most semantically appropriate with the sentence context. Another benefit of such an approach is that the parsing time can be considerably reduced as the grammar will use only one morphological value of a word, avoiding checking irrelevant paths.

Also, The SPEL architecture is based on English grammar written in a DCG form, the grammar is interpreted by a Prolog engine. The dictionary is an extension of the Wordnet dictionary and is stored in a relational form. The extension from the Wordnet is the adding of more syntactic information for the words contained in Wordnet. To analyze a certain sentence, SPEL first selects from the dictionary the words that may contribute to the sentence. The next step is to call the Prolog engine with the given words from the dictionary and the sentence to be evaluated to try to do the matching. If the match is successful, SPEL is able to draw the resulted syntactic parse. As regarding the incorrect sentences, the system is capable to recognize the incorrect words.

One of the system usages is to parse users' answers to psychological tests and return statistics of the morphological parts discovered in the users' answers. One of the issues with these answers is that the users tend not to answer in a sentence to the given question (for example, *What is the emotion that you feel when looking at the inkblot?*) or question-task (of the form *Describe what activity could be taken place in this sequence*), but rather to give only a syntagma. For example, for the following question "*What did you have for lunch?*" the user will answer in a syntagma like "*a donut*". Such a syntagma will not be able to be parsed by the system. In the following, we will give an algorithm able to transform these syntagmas in correct sentences using semantic and syntactic information from the question and syntactic information from the answer.

3. Syntagma processing

3.1. **Syntagma problem.** As we mentioned in the previous paragraph, the users tend to answer to a question in sentences that are not syntactically complete, most of the time the user answer is a very short syntagma that answers to the question. The goal for the SPEL system is to provide a system which could be able to recognize the syntagmas and to be able to reconstruct a syntactically correct

sentence from them without affecting the semantic information which remains intact.

3.2. **Identifying syntagmas.** One of the challenging problems regarding syntagma processing is, in fact, the syntagma identification. As for now the SPEL system considers syntagmas all the sentences that are not correctly identified by the grammar. The problem with this approach is that the system will consider as syntagma even the answers that are not syntactically correct (according to the given grammar). To avoid considering all the failed sentences as syntagma the system eliminates from syntagma the answers that could not be categorized in the Syntagma Categorization step (see 3.4). In the current state, the SPEL system first applies the answers against the grammar. In the case that the sentence is not recognized as being correct, the system will try to categorize the answer as a syntagma category. If the syntagma can be categorized than the sentence is considered as syntagma and the syntagma resolving step will occur that will reconstruct the sentence from the syntagma and the information from the question deconstruction step, finally the new reconstructed is applied again against the grammar. If the new reconstructed answer is recognized by the grammar, then the sentence is considered correctly reconstructed, otherwise the sentence is not recognized as a syntagma. The downside of this approach is that an answer has to be processed twice against the grammar doubling the processing time.

3.3. **Question Deconstruction.** In order to process the recognized syntagmas from the previous step, we need to determine some syntactic and semantic information for each question, information that was involved in the syntagma answer. To do this we will construct an array of pairs of properties of the form attribute:value. Depending on the scope of the questions, there are properties that need to be included. In our case, we considered the following attributes for each question:

- **Syntactical Subject**: representing the subject to which the question refers;
- **Verb involved**: representing the verb that contributes to the answer construction (most of the time it is part of the question);
- **Verb preposition**: sometimes the verb that contributes to the answer needs a preposition, as for example "of". This preposition will be given by the value of this attribute;
- **Logical Subject**: this attribute represents the object in the question. The object in the question becomes the action agent in the answer. The answer may have a syntactical subject but the real agent will be the value given by this parameter;
- **Original syntactical subject**: this is usually the second subject from the question;
- **Question verb**: the exact form of the verb that is also part of the answer;

Depending on the questions domain (for example, psychological test related question), there can be added other specific properties.

Let us consider the following question-task: *Describe what activity could be taken place in this sequence.* In this case, the syntactical subject is "*it*" resuming "*the activity*" expressed in the question, as the subject to which the question refers; the verb involved in this case is "*could be*"; the logical subject will be "*someone*" ("*someone*" is the agent of the action involved in the question); the original syntactical subject is "*I*"; the question verb in our question is "*could*"; the verb preposition is missing in this question. Finally, the properties array for the question will be represented as follows:

$$[syntactical\_subject(it), verb\_involved(couldbe), logical\_subject(someone), \\ original_{s}yntactical\_subject(I), question\_verb(could), verb\_preposition()]$$

In the following part of the article we will show how this information can be used in the sentence reconstruction from the syntagma that answers to the question.

3.4. **Syntagma categorization.** In order to be able to reconstruct the sentence using the syntagma, we will need, beside the question deconstruction, to categorize the syntagmas. This step is necessary in order to be able to apply specific rules for each kind of syntagma. We will present here only a partial question qualification that applies to psychological test answers. Thus, depending on the domain of the question, the syntagma classification involves several categories. Here are the main categories used by the SPELL system:

- **Participial syntagmas** - these are the syntagmas composed by a present participle (the forms in -ing). The syntagmas are considered part of this category if they start with a present participle verb. For example *having fun*, *looking at the sky* and *climbing a mountin*;
- **Subject elliptical verb syntagmas** - are the syntagmas constructed around a regular verb. To recognize these syntagmas, these are the ones that start directly with a verb, for example *work hard to get where I want*;
- **Noun phrase syntagmas** - are the syntagmas that represent a noun phrase. To identify these syntagmas, we have to pay attention to the structures that, if alone, are recognized as a noun phrase, for example: *a yellow building*;
- **Auxiliary verb elliptical syntagmas** - are the syntagmas that contain a present participle verb or a past participle verb and, also, where the previous word is not an auxiliary verb. There is a problem in order to do the classification of these syntagmas, because the verb form in the past participle is the same with the preterit form of the verb, so the confusion may occur between a subject elliptical verb or participial and auxiliary verb elliptical syntagmas;

All the answers that could not be classified here are not considered as syntagmas, but rather as syntactically incorrect sentences. This method is not an exhaustive method, but it can give very good results for domain specified questions (where most of the syntagmas tend to respect the existing rules). Another problem that arises here is the time needed to determine the syntagma category. In most of the cases, except for the noun phrase syntagmas, there is only a word lookup in the dictionary. And even more, if we consider that the sentences were previously checked against the grammar and the word retrieved from the dictionary, we already have the words morphological value, so the dictionary access is not needed in order to make the categorization. The same applies for the noun phrase syntagmas, as we can use the previous parsing phase to determine if the answer is actually a noun phrase. By using both this classification and the question deconstruction, we are now able to rebuild the sentences resumed by the syntagmas.

3.5. **Resolving syntagmas.** In order to reconstruct the sentences by using the question deconstruction and the syntagma characterization, we will create rules that apply for each syntagma category. As in the previous cases, these rules depend on the question domain and can't be used as a general rule for a particular syntagma category. Also, because of this, our solution doesn't provide a precise sentence reconstruction. Still from our practical result this algorithm gives a good ratio of well constructed sentences. Another problem represents the fact that the reconstructed sentence needs to be again applied against the grammar to check if it is a syntactically correct sentence. However, by building a sentence using the elliptical structures that we call here "syntagmas", we have the possibility to include in our parsing structures that usually are considered to be syntactically incorrect because elliptical. In the following we will present the rules used by the SPEL system in order to reconstruct the sentence from the syntagmas.

a) In the case of **participial syntagmas**, the sentence will be reconstructed using the following formula:

$sentence = syntactical\_subject + verb + logical\_verb + verb\_preposition + syntagma$

To demonstrate this, we consider the question-task from the section 3.3: *Describe what activity could be taken place in this sequence?* The answer that was given to this kind of question: *Having on a costume going to a Halloween party.* By using the given rule and the question deconstruction, the new recognized sentence will be: *It could be someone having on a costume going to a Halloween party.*

b) In the case of **subject elliptical verb syntagmas**, we use the following formula:

$$sentence = syntactical\_subject + syntagma$$

In order to exemplify this situation, we could have as an answer a subject elliptical verb syntagma, a construction of the type *looks like someone is crying.* According to our formula, our syntactical_subjet is it, so the rebuilt sentence will be: *It looks like someone is crying.*

c) For **participial syntagmas**, the formula to be applied will have the following form:

$$sentence = syntactical\_subject + verb + logical\_verb + syntagma$$

As an example of such syntagma, let us assume that for our question-task *Describe what activity could be taken place in this sequence?* the answer is *cared away by his emotions*. In this case, the reconstructed sentence will be: *It could be someone cared away by his emotions*.

d) To pursue our syntagma reconstruction examples, in the situation where the answer qualifies as a noun phrase syntagmas, the formula to be applied wil be:

$$sentence = syntactical\_subject + verb + logical\_verb + syntagma$$

Thus, as an answer to the same question-task as previously, the answer could be of the form: *a war scene with guns*. In order to reconstruct a syntactically correct sentence, the noun phrase syntagma will be consequently transformed as: *It could be a war scene with guns*.

e) For the situation where the answer is an auxiliary verb elliptic syntagmas, the formula will be:

$$sentence = syntagma\_subject + verb + logical\_verb + remaining\_syntagma$$

If we consider the response: *someone having a crisis*, the syntagma subject is *someone* and the remaining_syntagma is *having a crisis*. Here, the noun phrase preceding the participle becomes the actual subject of the reconstructed sentence, the verb involved in the question (in our case could be) becomes the main verb. Since the verb to be is an auxiliary verb, the participle in our syntagma is going to complete the verb, thus the solution: *Someone could be having a crisis*.

4. Conclusion

As we mentioned in this article, the given solution is not a precise solution, but it gives good results for domain specific sentences. As an improvement we can use multiple rule assignations for each syntagma category, each rule with a probability value assigned to it. In this case the algorithm will be changed in the sense that, instead of trying only one rule for the sentence reconstruction, it will try all the rules and select the one assigned with the highest probability. Another aspect that was not discussed in detail is the cost of this algorithm. The cost is a significant point as this kind of algorithms are mostly used in fields where there are a few questions answered by thousands of students. As it can be noticed, the SPEL steps do not involve a high cost, compared to the grammar application against the sentence. Still, a big cost comes from a second checking of the reconstructed sentence against the grammar to be sure that the reconstructed sentence respects the grammar. We are developing at the present time another version of the algorithm that involves all the processes to be fulfilled in the first grammar checking phase, by adding extra information to each participating word.

But in this case we have to consider not increasing too much the cost for the syntactically correct sentences. As mentioned before, this solution is used in the current SPEL implementation with very promising results.

References

[1] Ch. Fellbaum, Wordnet. An electronic lexical database, The MIT Press, Massachusetts, 1999
[2] C. H. A. Koster, E. Verbruggen, "The AGFL Grammar Work Lab", in Proceedings FREENIX/Usenix, 2002, pp 13-18
[3] D. Bechet, "K-valued link grammars are learnable from strings", in Proccedings of the 8th conference on Formal Grammar (FGVienna), Viena, 2003
[4] E. Briscoe, J. Carroll, "Robust Accurate Statistical Annotation of General Text", in Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas, 2002, pp. 1499-1504
[5] The connexor parser wbe page: http://www.connexor.com/demo/syntax/
[6] P. Blackburn, J. Bos, Representation and Inference for Natural Language. A first Course in Compuational Semantics. Volume I Working with first order logic. Computerlinguistik, Universitt des Saarlandes, 1999
[7] A. Onet, D. Tatar, "The semantic representation of Natural Language sentences. A theoretical and practical approach", in PC 132 God, nr.1797, 2001, Budapesta pp.195-204

(1) Babes-Bolyai University Cluj-Napoca, Faculty of Mathematics and Computer Science

E-mail address: adrian@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 58–65

A TEXT ANALYSIS BASED APPROACH FOR THE COMPLIANCE BETWEEN THE SPECIFICATION AND THE SOFTWARE PRODUCT

DANA LUPŞA[1] AND ADRIANA TARŢA [2]

Abstract. Nowadays, the success or failure of a software product depends on its quality. An essential component of software quality is its functionality. In this paper we propose a new approach in evaluating the compliance between software documentation (expressed on natural language) and the final software product. We define two evaluation measures and present some case studies.

1. Introduction

The continuous development of computer science and the increased expansion of application areas of software products has raised more and more frequent the question: *What makes a good project?*

According to ISO-9126 [**?**] the factors of the software product quality are functionality, reliability, usability, efficiency, maintainability, and portability. *Functionality* is the ability to satisfy stated or implied needs. It assumes the program is correct. Correctness is strongly connected to good specifications and good design [**?**]. *Reliability* refers to the capability of software to maintain its level of performance under stated conditions for a stated period of time. *Usability* refers to the effort needed to use the software product. *Efficiency* is related to the relationship between the level of performance of the software and the amount of resources used, under stated conditions. *Maintainability* refers to the effort needed to make specified modifications and *portability* is related to the ability of software to be transferred from one environment to another.

Software developers are interested in saving time and costs along with minimizing the risks associated with non-compliance between user needs and final program functionality. Researchers agree that, for achieving these goals, it is necessary to develop and implement an effective standard management solution that

2000 *Mathematics Subject Classification.* 68Q60.
Key words and phrases. software quality, software specification, requirements.

©2007 Babeş-Bolyai University, Cluj-Napoca

will result in the engineers and project staff actually using the standards and specifications [**?**].

In this paper we propose a new approach in studying the quality of a software product based on the analysis of its functionality. We will acquire knowledge about the software functionality from requirements documents. In order to evaluate the quality of the final product we propose two measures that indicates its compliance to the specification documents.

The paper is structured as follows: Section 2 gives a short presentation of the specification documents. Section 3 presents some approaches in the domain of writing good requirements specification. Section 4 proposes an original view on the automatic analyse of the compliance between the requirements and the final product. Section 5 describes some real case studies. The paper ends with conclusions and identifies some key issues for future work in this area.

2. Specification documents

To create a software application, the description of the problem and the requirements are needed, i.e. what the problem is about and what the system must do. Specification documents are the results of an investigation of the problem rather than how a solution is defined [**?**]. One important principle that must be followed when developing a complex software system is: *Think first, program later* [**?**]. The first step in applying this principle consists of defining the problem completely [**?**], [**?**]. A good software specification is the first step toward a successful software product.

The description of the problem is usually called the *problem specification*. It contains a short description of where the user needs support from the program. It is usually informal and it can be considered as a blueprint for the problem analysis.

Requirements are description of needs or desires for a product. The primary goal of the requirements phase is to identify and document what is really needed. The documents must be easily understood by the clients and the development team.

Requirements are typically classified into three categories:

- functional requirements - describe system features or what the system must do;
- non-functional requirements - describe properties the system must have (e.g. performance, availability, accessibility);
- constraints - limits the development in some way. For example, a constraint can be the operating system the system must run on, or the programming language that must be used to implement the system.

The description of functional requirements is named *functional specifications* (FS).

Essential features of requirements are:

- Necessary - contains elements that must be included and for which other system components will not be able to compensate.
- Unambiguous - susceptible to only one interpretation.
- Concise - stated in language that is brief and easy to read, yet conveys the essence of what is required.
- Consistent - does not contradict other stated requirements nor is it contradicted by other requirements. In addition, the specification must use terms that have the same meaning in all statements of the documents.
- Complete - stated entirely in one place and in a manner that does not force the reader to look at additional text to know what the requirements means.
- Reachable - a realistic capability that can be implemented for the available money, with the available resources, in the available time.
- Verifiable - must be able to determine that the requirements have been met through one of four possible methods: inspection, analysis, demonstration, or test.

During analysis review, comparison of application domain model with client's reality may result in changes to each. Specifications are most important for external interfaces that must remain stable [**?**]. One of the main problems of requirements elicitation is expressing customer requirements in a form that can be understood not only by requirements engineers but also by noncomputer professional customers and users. The usual choice for expressing elicited requirements is natural language, since it is frequently the only common language to all participants.

Our approach, presented in Section 4, analyses the correspondence between specification documents, expressed in natural language, and the visible linguistic components of the final product, which is the user interface. We propose two measures that indicate the degree of consistency between specifications and the user interface (UI). We have applied our measures to problem specifications, functional specification and user manuals (UM) and we will present a short comparison of the results. We focus on functional specification because it is the most detailed specification document.

3. Related work

The previous work in the domain of writing good requirements specification focuses on finding solutions to build the appropriate requirements based on some automated processes or some patterns. Two approaches will be presented in the following.

In [**?**], the authors present requirements templates that can improve requirements elicitation and expression. They use two categories of patterns: linguistic patterns (very used sentences in natural language requirements descriptions), and

requirements patterns (generic requirements templates that are found very often during the requirements elicitation).

In [**?**] the authors present a system that automatically verifies some desired quality properties of software requirements, for example the unambiguity and completness. Their approach is based on the representation of software requirements in XML and the usage of the XSLT language .

As far as we know, there is no approach that investigates the correspondance between software specification documents and the software product, based on text processing.

4. Our approach

We propose a method to evaluate the compliance between requirements specification and the final software product. Our approach deals with specifications in natural language.

A good functional requirements specification is a key step towards a successful software product. Obviously, this is true only if the software product follows the specification. We propose two measures to evaluate the correlation between the specification in natural language and the "natural language" part of a software product, which is the user interface. They are useful to verify the degree to which a software product respects specifications (in the case of a good quality specification) or to evaluate the completeness of a specification (in the case of a program that meets the quality standards).

According to [**?**], a functional specification is "a formal document used to describe in detail for software developers a product's intended capabilities, appearance, and interactions with users". This means that the words chosen to appear in the user interface (UI) illustrate the concepts described in functional specification. The compliance between the specifications and the UI grows with the number of common words. Based on this idea, we propose two measures: $\mathcal{CW}$ and $\mathcal{CWT}$, that are based on the number of words that appear both in the UI and in the specification. Their values are scaled in order to obtain real values from the interval $[0, 1]$. They reach the maximum (value 1) when all the words from the UI are present in the specification document . This is also the ideal case.

The first measure, denoted by $\mathcal{CW}$, is based on the number of *Common Words* from FS and UI.

$$\mathcal{CW} = \frac{common - words}{words - in - the - user - interface}$$

The second measure, denoted by $\mathcal{CWT}$, is based on counting the *Common Words Truncated to first k letters*:

$$\mathcal{CWT} = \frac{first - k - letter - from - common - words}{first - k - letter - from - words - in - the - user - interface}$$

From all the specifications, the functional specification contains the most detailed description of the application functionalities. It is the most related to the user interface. That is why we consider that the two measures are appropriate to evaluate the compliance between FS and UI.

The problem specification makes a short description of user needs. Parts of the problem specification should be reflected on the UI, if the implementation is compliant with the problem specification. The concepts presented in the problem specification are expected to be present in the UI. We expect that, in this case, the values of $\mathcal{CW}$ and $\mathcal{CWT}$ will be lower than in the case of the functional specifications.

The user manual (UM) describes and explains all the items in the UI. We consider that the $\mathcal{CW}$ and $\mathcal{CWT}$ measures should also be a good indicator of the quality of the UM document.

In the next section a detailed study of some functional specifications is described. A comparative view of the correspondance between problem specification, functional requirements and user manual, on one side, and the user interface, on the other side, is also presented.

5. The Experiments

We have studied the $\mathcal{CW}$ and $\mathcal{CWT}$ measures for four cases of applications developed for the points based jobs evaluation method. The main functionalities of the applications are: job management, job evaluation sessions management, job evaluation factors configuration and job evaluation results management. The applications were developed using Borland Delphi 7 environment.

The words from the user interface were automatically extracted. A Java application was developed for this purpose. It parses the **.dfm* files from the project and extracts the words associated to the widgets displayed on the user interface. The output generated by this application is a file containing the words extracted from the user interface.

For those applications, we have one problem specification and four triplets of functional specification, user interface and user manual. All documents were written in Romanian.

Truncation is usually used for languages for which a stemmer is missing. The number of characters the word is truncated to is the truncation parameter. A common practice for choosing the truncation parameter is to use a value smaller, but close to, the average length of the words in that language. In our case, we have extracted the Romanian words from the freely available Romanian-English dictionary from [**?**]. The average length of the words from this dictionary is 7.65. We choose the value 6 for the truncation parameter.

FS	$\mathcal{CW}$	$\mathcal{CWT}$
FS1	0.31	0.47
FS2	0.73	0.86
FS3	0.77	0.82
FS4	0.77	0.80

TABLE 1. Values of $\mathcal{CW}$ and $\mathcal{CWT}$

5.1. **FS and UI Compliance.** We have computed the $\mathcal{CW}$ and $\mathcal{CWT}$ measures on four good quality functional specifications. We have chosen k=6 for evaluating $\mathcal{CWT}$. The results are presented in Table 1.

We expect that $\mathcal{CWT}$ makes a more accurate estimation of the common concepts because it approximates better the number of common word roots. In order to determine the origin of the differences, we focus on the functional specification that has the lowest $\mathcal{CW}$ and $\mathcal{CWT}$ scores. We have considered the words that appear in the UI but are missing from the functional specification. We classified them in two sets as presented in Table 2. The words in the first set can be characterized as being specific for working with computers and they do not describe the functional logic of the application. From this point of view, their missing is insignificant. The second set contains more meaningful words. If we take a look at the functional specification, we can see that, for most of them, different words derived from the same word root appear. For example, we can not find the words like *expert (expert)*, *evaluatori (evaluators)*, but words like *expertii (the experts)*, *evaluatorii (the evaluators)* are present. This issue is solved by using the second proposed measure, $\mathcal{CWT}$.

Set	**Words**
Set 1	*accepta (accept), adauga (add), adaugare (addition), iesire (exit), intra (login/enter), salvare (save), selectate (selected), selectati (select), sterge (delete), stergere (deletion), ...*
Set 2	*expert (expert), evaluatori (evaluators), ...*

TABLE 2. Some words that appear in the UI and do not appear in FS

The values computed for $\mathcal{CWT}$ measure increase because the number of differences decreases. For example, in this case, for two separate words present in UI: *sterge (delete)* and *stergere (deletion)*, $\mathcal{CWT}$ (with $k = 6$) "sees" only one truncated word component, that is *sterge*.

A possible drawback of the $\mathcal{CWT}$ measure is generated by the truncation to a fixed number of characters. It is possible that two non-related words to be truncated to the same k characters. We manually verified our data and we found that there are no such situations.

5.2. **Problem Specification, FS, UM and UI Compliance.** We have evaluated the $\mathcal{CW}$ and $\mathcal{CWT}$ measures for a set of: one problem specification, four functional specifications (FS) and four user manuals (UM). For $\mathcal{CWT}$, we have chosen the parameter $k = 6$.

The results are presented in Figure 1. The first bar set indicates the values of $\mathcal{CW}$ scores, and the second bar set indicates the values of $\mathcal{CWT}$ scores. Each set represents the $\mathcal{CW}$ or $\mathcal{CWT}$ scores for program specification, functional specification and user manual. For FS and UM we have represented the minimum and maximum values of $\mathcal{CW}$ and $\mathcal{CWT}$ obtained for the four specifications. Minimum is depicted in dark grey and maximum in light grey.

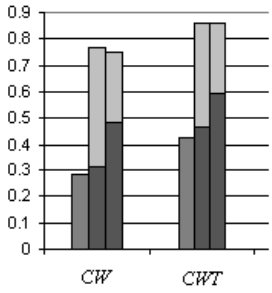


FIGURE 1. A comparative view over the values of $\mathcal{CW}$ and $\mathcal{CWT}$ for problem specification, FS and UM

Problem specification is a more general specification while functional specification is a more detailed one. That is why we expect lower values for $\mathcal{CW}$ and $\mathcal{CWT}$ for the problem specification.

Usually, the user manual refers to every detail in the UI. That is why the highest scores for $\mathcal{CW}$ and $\mathcal{CWT}$ are expected to be achieved for UM. On the other hand, a UM can contain screen shots from the application, that is why it is possible that the text in the UM does not contain every word from the UI.

The results depicted in Figure 1 confirm our expectations: the lowest values for $\mathcal{CW}$ and $\mathcal{CWT}$ are achieved for the problem specification and the highest ones for the functional specification and the user manual.

6. Conclusions and further work

A good functional requirements specification is a key step towards a successful software product. In this article, we have proposed two measures, $\mathcal{CW}$ and $\mathcal{CWT}$, to evaluate the compliance between product specification in natural language and the user interface. In our experiments, we have obtained high scores for some good quality specifications.

In the future, our intention is to improve the measures we have introduced in this paper. The starting point could be the use a stemmer instead of a blind truncation of words to a fixed number of characters. We also plan to further investigate the relations induced by the presence of the concepts, not only by the presence of the words. This can be done using some sort of semantic analysis.

(1) Babeş-Bolyai University, Str. M. Kogălniceanu nr. 1, Cluj-Napoca, Romania
E-mail address: dana@cs.ubbcluj.ro

(2) Babeş-Bolyai University, Str. M. Kogălniceanu nr. 1, Cluj-Napoca, Romania
E-mail address: adriana@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 66–72

TEXT CATEGORIZATION EXPERIMENTS USING WIKIPEDIA

ZALÁN BODÓ, ZSOLT MINIER, AND LEHEL CSATÓ[(1)]

Abstract. Over the years many models had been proposed for text categorization. One of the most widely applied is the vector space model, assuming independence between indexing terms. Since training corpora sizes are relatively small – compared to ∞ – the generalization power of the learning algorithms is relatively low. Using a bigger unannotated text corpus can boost the representation and hence the learning process. Based on the work of Gabrilovich and Markovitch we use Wikipedia articles to give word distributional representation for documents. Since this causes dimensionality increase, some feature clustering is needed. For this end we use LSA.

1. Introduction

Text categorization is one of the more profoundly examined task of information retrieval. The amount of textual information available these days makes more and more necessary the intelligent and *efficient* methods that help navigating in this virtual space.

It is a "classical" categorization or classification task: given a function $f : D \to 2^C$ (by training examples), where D is the set of documents and C is the set of categories, find $\widehat{f}$, which best approximates f. The training examples $(\mathbf{d}_i, y_i)$, $i = 1, \ldots, |D|$, compose the training corpus, where $\mathbf{d}_i$ and y_i denotes the document and the associated label respectively. The problem is evidently a multi-class and multi-label task, since usually there are more than two classes, and a document can belong to many classes.

The solution of the problem is usually separated into two phases: term selection and machine learning. Both of these two are important parts of a text categorization system. A good term selection is very useful, because usually the data contain noise, irrelevant features can lead the system into wrong direction, etc. Some machine learning techniques like SVMs and Rocchio's method can also filter out irrelevant terms.

2000 *Mathematics Subject Classification.* 68P20, 68T30.

Key words and phrases. text categorization, document representation, feature selection, latent semantic analysis.

©2007 Babeş-Bolyai University, Cluj-Napoca

The most widely used model in information retrieval and hence in text categorization is the vector space model (VSM) introduced in [9]. The index terms are words or stems taken from the training corpus, which constitute the basis of the vector space, therefore they are assumed to be independent – although words are evidently not semantically independent. Because of its simplicity and good performance it is the most popular one used in the IR community, since other, more sophisticated models do not provide significantly better performance. Meaningful, descriptive terms should get a higher weight in the representation, so term weighting is also an important factor of the system. The term frequency×inverse document frequency (tfidf) is defined as

$$w_{ij} = \mathrm{tr}(i,j) \cdot \log \frac{|D|}{n_j}$$

which gives higher weights for more descriptive (frequent) terms in a document (tf), and also higher weights for terms which have more discriminative power (idf). Here w_{ij} denotes the weight of word j in document i, $freq(i,j)$ is the frequency of word j in the ith document, $|D|$ is the total number of documents and n_j is the number of documents in which word j appears through the corpus.

For a good comparative study on the basic feature selection techniques in text categorization see [11], while [10] gives a broad overview on different machine learning techniques applied to the problem.

In this paper we study the use of Wikipedia derived knowledge in enhancing text categorization. The next section constitutes the main part of the article describing the steps of the proposed method. We describe the experiments in section 3 and discuss the results in section 4.

2. Wikipedia-based text categorization

2.1. **Wikipedia.** The Wikipedia is the largest encyclopedia edited collaboratively on the internet comprising of ≈ 1.6 million concepts totalling ≈ 8 gigabytes of textual data. It is written in a clear and coherent style with many concepts explained sufficiently deeply thus making it a wonderful resource for natural language research. Our need for a semantic relatedness metric between words could be easily approached using the distribution of words in the different Wikipedia concepts.

2.2. **Document representation.** Document representations can be very sparse in the vector space model, because they are indexed by a few word taken from the training corpus.

Through the article any vector is considered a row vector.

Gabrilovich and Markovitch [6] use word distributional representation for measuring word and text *relatedness*, using Wikipedia articles. We adopt their technique to represent documents in this concept space.

Given a document $\mathbf{d}$ containing $|W|$ terms it can be transformed to the Wikipedia concept space by

$$\underbrace{\begin{pmatrix} w_1 & w_2 & \dots & w_{|W|} \end{pmatrix}}_{\mathbf{d}} \cdot \underbrace{\begin{pmatrix} c_{11} & c_{12} & \dots & c_{1|C|} \\ c_{21} & c_{22} & \dots & c_{2|C|} \\ \vdots & \vdots & \ddots & \vdots \\ c_{|W|1} & c_{|W|2} & \dots & c_{|W||C|} \end{pmatrix}}_{\mathbf{W}}$$

where the w's are the weights of the words (e.g. tfidf, idf calculated upon the categorization corpus or Wikipedia) and c_{ij} represents the weight of the word i in Wikipedia concept j. It is easy to observe that the matrix $\mathbf{W}$ is a term×concept (document) matrix, which transforms the document to the concept space, therefore by calculating the *similarity* of two text gives us the well-known GVSM-kernel [12], [3]

$$K_{\mathrm{GVSM}}(\mathbf{d_1}, \mathbf{d_2}) = \mathbf{d_1}\mathbf{W}(\mathbf{d_2}\mathbf{W})^{\mathrm{T}} = \mathbf{d_1}\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{d_2}^{\mathrm{T}}$$

where $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ is a term×term correlation matrix, now built upon external information.

We have used the same representation for documents, that is we transformed documents from the term space to the concept space by multiplying the document×term matrix ($\mathbf{X}$) by $\mathbf{W}$. Although both $\mathbf{X}$ and $\mathbf{W}$ are sparse matrices (with a density of 0.67% and 0.2%, respectively), their product results in a much more denser structure. Though – similarly to [6] – we filtered out Wikipedia articles considered less important, the resulting matrix is still huge and dense, thus making difficult to store and actually use the constructed document vectors.

2.3. **Dimensionality reduction.** To address the high dimensionality of vector spaces in natural language processing, dimensionality reduction techniques are often used. Dimensionality reduction helps at making feature vectors small enough to be handled by machine learning methods and many times has the beneficial effect of removing noise and thus slightly increasing classification accuracy and reducing overfitting.

2.3.1. *Latent Semantic Analysis.* Latent Semantic Analysis was introduced by Deerwester et al. [4] in information retrieval. It is a dimensionality reduction technique based on singular value decomposition. Given the term×document matrix $\mathbf{A}$, it is decomposed as $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$ where $\mathbf{U}$ and $\mathbf{V}$ are matrices of orthonormal columns and $\mathbf{S}$ is diagonal. By controlling the number of singular values in $\mathbf{S}$ one can achieve a dimensionality reduction in both the term×term matrix $\mathbf{U}$ and in the document×document matrix $\mathbf{V}$.

We used this method to reduce the dimensionality of the word×concept matrix that we built from Wikipedia.

2.4. **Learning over training data.** For learning the decision boundaries we applied support vector machines using the LIBSVM [1] library with linear kernels. Support vector machines (SVMs) were introduced by Vapnik for binary classification. Although the formulation of the problem can be extended in some ways to handle multi-class classification, in most cases – because of the lower computational cost – binarization techniques like one-vs-rest, one-vs-one, error correcting output coding, etc. are used for supporting non-binary classification.

In its simplest form the SVM maximizes the margin $(2/\|\mathbf{w}\|)$ of the hyperplane which separates positive and negative examples, that is under the following condition:

$$y_i \cdot (\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \geq 1, \quad \forall \mathbf{x}_i \in P \cup N$$

where P and N denotes the set of positive and negative samples respectively. The decision function is simply

$$f(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$

The separating hyperplane of maximal margin depends only on the vectors supporting the marginal hyperplanes, and this is the reason why is called support vector machine.

Joachims [7] experimentally proved that the classes induced by the documents from the Reuters corpus are more or less linearly separable using the vector space model. We assumed the same holds for our model.

3. Experimental methodology and results

The first step constituted building an inverted index for the words appearing in Wikipedia, excluding stop words and also the first 300 most frequent words. Because the large amount of Wikipedia articles turns the problem into one of unmanageable size, as we have mentioned earlier, we filtered out some of them, namely we eliminated those containing less than 500 words or having less or equal than 5 forward links to other Wikipedia articles. In this way we processed 327 652 articles.

For testing the model we used the Reuters-21578 text categorization corpus, ModApté split with $90 + 1$ categories. The Reuters collection contains documents which appeared on the Reuters newswire in 1987. These documents were manually categorized by the personnel from Reuters Ltd. and Carnegie Group Inc. in 1987. The collection was made available for scientific research in 1990. Originally, there were 21 578 documents, but some of them, namely 8681 were unused in the split, moreover, some of it were not categorized – they were put in the unknown category. Removing this "virtual" class, the training and the test corpus contains 9583 and 3744 documents respectively, defining 90 classes. Some of the documents are assigned to more than one category, the average number of classes per document being 1.24 [2].

In the preprocessing step we selected the top 5209 word stems of the Reuters corpus using the χ^2 term ranking method [11]. For these 5209 terms the inverted index was built based on the Wikipedia, that is, each term is represented as a vector of occurences in the vector space of Wikipedia concepts. The number of dimensions of this vector space is 327 652.

With this data we performed four experiments on the Reuters corpus.

In the first experiment (χ^2[5209]) we measured the performance of the system with the terms extracted from the corpus itself. We used these results as a baseline for another term selection method [8] and we used them for baseline here as well. In [8] we proposed a term selection method based on segmenting the textual data for categorization, then clustering these text segments in each category to obtain the largest clusters and using the terms (stems) from the merged clusters as features. The results obtained for the Reuters corpus were similar to the performance of the χ^2 term ranking method, however in our method there is no need to determine the optimal number of features. Therefore we used χ^2 term selection with our feature threshold.

In the second experiment (χ^2[5209]+LSA) we expressed the documents of the Reuters corpus with Wikipedia concepts through transforming the words in each document into Wikipedia concept space. However, as the dimensionality of the Wikipedia concept space is prohibitively large, we reduced it using the LSA method to an arbitrarily chosen 2000 number of dimensions. Effectively, in the transformation $\widehat{\mathbf{X}} = \mathbf{XW}$ where $\mathbf{X}$ is the data of the corpus (document×word) and $\mathbf{W}$ is the Wikipedia matrix (word×concept) we replace $\mathbf{W}$ with $\mathbf{U}$ from the singular value decomposition of $\mathbf{W} \approx \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^{\mathrm{T}}$ keeping only the first $k = 2000$ columns in $\mathbf{U}$. We did not actually perform this singular value decomposition, because it is not needed to find $\mathbf{U}$ which is actually the matrix containing the eigenvectors of $\mathbf{WW}^{\mathrm{T}}$ which can be found by principal component analysis. This way the final number of dimensions of each document in the corpus is 2000 and the document vectors are dense.

In the third experiment (χ^2[2000]) we selected only the first 2000 top ranking word stems from the corpus as terms using the χ^2 method to compare our dimensionality reduction based on semantic relatedness of words to term ranking. The documents are represented as sparse vectors with 2000 dimensions.

In the fourth experiment (χ^2[5209]+noLSA) we did not perform a principal component analysis of $\mathbf{WW}^{\mathrm{T}}$ so we transformed the corpus with $\mathbf{WW}^{\mathrm{T}}$ to use all the semantic relatedness between words obtained from Wikipedia. Document vectors became dense with their original dimension of 5209.

The results we obtained are shown on figure 1.

The performance is measured using the common precision-recall breakeven point – the intersection of the precision and recall curves if such a point exists – and the F_1 measure.

	mP	mR	mBEP	mF1	MP	MR	MBEP	MF1
χ^2[5209]	88.46	84.59	86.52	86.48	71.61	61.25	66.43	66.02
χ^2[5209]+LSA	86.68	82.59	84.63	84.58	62.97	53.61	58.29	57.91
χ^2[2000]	87.34	84.21	85.77	85.75	64.76	58.14	61.45	61.27
χ^2[5209]+noLSA	48.95	35.42	42.18	41.10	8.79	5.35	7.07	6.65

FIGURE 1. Performance results obtained for the Reuters corpus given in percentage. Notation: mP=micro-precision, mR=micro-recall, mBEP=micro-breakeven, mF1=micro-F_1, MP=macro-precision, MR=macro-recall, MBEP=macro-breakeven, MF1=macro-F_1

4. Discussion

As the results show, using Wikipedia in this way did not help classification performance. We also tried to use nonlinear kernels as inhomogeneous polynomial and RBF with optimized parameters by cross validation, but none of them produced a significant improvement, so we did not insert these results in the paper. Including the semantic relatedness or cooccurrence information in the document vectors, the performance drops, meaning that it only shows up as noise in the data. It appears that Wikipedia-based semantic relatedness does not model well the similarity between the documents from the same Reuters class. However, Gabrilovich and Markovich [5] showed that by augmenting the features of the bag-of-words model with closely related Wikipedia concepts results in significantly better performance.

Using the Wikipedia-based representation for word similarity Gabrilovich and Markovich [6] obtained a much more higher correllation with human judgement as for the cosine similarity. This means that the document representation as a weighted sum of the contained words' vectors is inappropriate, otherwise the kernel $\mathbf{X}\mathbf{U}_k\mathbf{U}_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}$ would improve categorization performance.

One possible improvement could be to cut off extremities from the transformation matrix to increase the sparseness of document representations and decrease possible noise.

The method should be tested on other corpora, to verify that semantic relatedness – derived from Wikipedia – can help categorization. The Reuters collection is very unbalanced, causing serious difficulties for text categorization systems.

Our experiments show that still the bag-of-words model performs better than the semantic space model of documents.

5. Acknowledgements

We would like to acknowledge the financial support of the grant CEEX/1474 by the Romanian Ministry of Education and special thanks to the anonymous reviewers of our paper for the supporting critics.

References

[1] Chih-chung Chang and Chih-jen Lin. LIBSVM: a library for support vector machines (version 2.31), September 07 2001.

[2] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In *SIGIR*, pages 151–158. ACM, 2002.

[3] Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. Latent semantic kernels. *J. Intell. Inf. Syst*, 18(2-3):127–152, 2002.

[4] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, June 1990.

[5] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*. AAAI Press, 2006.

[6] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI)*, January 2007.

[7] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In Claire Nedellec and Céline Rouveirol, editors, *ECML*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998.

[8] Zsolt Minier, Zalán Bodó, and Lehel Csató. Segmentation-based feature selection for text categorization. In *Proceedings of the 2nd International Conference on Intelligent Computer Communication and Processing*, pages 53–59. IEEE, September 2006.

[9] G. Salton, A. Wong, and A. C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:229–237, 1975.

[10] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[11] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, 1997.

[12] Yiming Yang, Jaime G. Carbonell, Ralf D. Brown, and Robert E. Frederking. Translingual information retrieval: Learning from bilingual corpora. *Artificial Intelligence*, 103(1–2):323–345, 1998.

[1] Department of Mathematics and Computer Science, Babeş-Bolyai University, RO-400084, Cluj-Napoca, Romania

E-mail address: {zbodo, minier, lehel.csato}@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 73–80

THE 'INTEGRAL' MODEL OF LANGUAGE FUNCTIONING (E. COŞERIU)

EMMA TĂMÂIANU-MORITA[1], CORNEL VÎLCU[2],
AND MAGDALENA CIUBĂNCAN[3]

Abstract. The paper explores the framework of Coseriu's "integral linguistics", focusing mainly on the three planes of language and their corresponding "linguistics" - the three directions in language investigation that Coseriu postulated. It is argued that, in the panorama of contemporary approaches to language, Coseriu's integral linguistics offers one of the most comprehensive and finely articulated frameworks for investigating the functioning of language in a dynamic perspective.

1. INTRODUCTION: THE THREEFOLD STRUCTURE OF "INTEGRAL LINGUISTICS"

There are three types of linguistic content and, as long as we intend to represent the reality of language functioning, in a meaningful and accurate manner, these should be submitted to three wholly different methods of formal treatment: undoubtedly this would be the most important contribution that Integral Linguistics, the theory of language founded and developed by Eugeniu Coseriu beginning with the 5th decade of the 20th century, can bring to any scientific debate concerning Computational Linguistics. Also, the theory of language that will be briefly presented here brings into focus wider, epistemological and philosophical issues concerning, primarily, the very problem of where the general science of Linguistics should be situated within the disciplinary field of sciences seen as a fully articulated ensemble [1]. What follows is an outline of Coserius comprehensive model of language functioning, devised by the authors on the basis of numerous Coserian sources, and also drawing upon our own investigations into specific issues within

2000 *Mathematics Subject Classification.* 91F20.

Key words and phrases. Coserian semantics, elocutional, idiomatic and textual competence.

[1]Extensive debates on the difference, at the level of the grounding principles, between "integral" linguistics and other theoretical orientations are undertaken in [19], [20], [21], [22], [17], [18]. It should be noted therefore that the term "integral linguistics" is used here as the specific denomination of Coserian linguistics, a denomination explicitly selected and substatiated by Coseriu himself in the later years of his scientific activity

©2007 Babeş-Bolyai University, Cluj-Napoca

this model. Integral linguistics claims that there is a clear distinction between sciences of nature and sciences of culture and that human language cannot be described and dealt with by using the same apparatus which is used for natural sciences, since the objects of the two types of sciences are governed by different laws: while natural objects belong to the world of necessity, which is governed by causes that produce certain effects, cultural objects, on the other hand, belong to a world that is specific to humanity, namely that of freedom ([6], chapter 3).

VIEWPOINT LEVEL	**Activity** Energeia	**Knowledge** (Competence) Dynamis	**Product** Ergon
Universal Speaking in general (universally-human activity)	Speech in general	Elocutional	Empirically infinite totality of utterances
Historical Particular languages (idiomatic traditions)	Concrete language	Idiomatic	Abstract language
Individual Discourse / Text (individual speech)	Discourse	Expressive	Text

The backbone of Coserius outlook on how language works is his well-known triad of language planes or levels of manifestation (Ebenen des Sprachlichen), outlined in what follows. In a definition that appears in the early fundamental study Determinacin y entorno, el objeto de la lingstica (?ciencia del lenguaje?) slo puede ser el lenguaje, en todos sus aspectos. Y el lenguaje se da concretamente como actividad, o sea, como hablar [...]. Ms an: slo porque se da como activitad, puede estudiarse tambin como producto?" ([2]: 285-286).

An objectively grounded theory of language will start from two general observations ([2]: 285-287, [3],[13]: 74):

(A) that language is (1) a generally-human activity (Ttigkeit), exerted by individuals (2) as representatives of communitary traditions of speech competence (Sprechen-knnen) (3) at an individual level; (B) that any activity, including the activity of speaking, can be regarded (a) as activity as such (enrgeia), (b) as the knowledge or competence underlying the activity (dynamis), and (c) as the product of that activity (ergon). The two triads (3 levels of manifestation and 3 points of view) delineate nine aspects of language as a creative cultural activity, aspects which can also be found as such in the intuitive knowledge of speakers ([13]: 59, 72, 75).

Integral linguistics posits itself as the epistemic exploration of language in all of these forms [2] and only in these forms (i.e. integral linguistics will not take as its specific object the biological abilities that underlie speech activity, or the external, social-institutional environments of speech): although admitting the fact that human beings' general capacity for expression also comprises 'non-verbal', as well as physiological/neurological aspects, Coseriu restricts the focus of linguistic inquiry to the domain of cultural linguistic competence, comprising the three levels of organization indicated above (cf. cite13: 65). By virtue of its specific object of study, any realistic approach to language structure and functioning, formal approaches included, will therefore fall into one of these nine aspects, and will have to define its goals and methodology accordingly. The speaker is understood to have an intuitive knowledge of three kinds of entities/ procedures, and to be able to convey, simultaneously, within the same act of speech, three kinds of linguistic contents (meaning). The types of meaning corresponding to each level are correlated with peculiar evaluations of adequacy that can be suspended bottom to top [3]. In the end, integral linguistics as a science can be conceived as a threefold system of knowledge:

level plane of language	type of content	adequacy judgement	science
universal	designatum	congruence	'designational' linguistics
historical	significatum	correctness	'significational' linguistics
individual	sense	appropriateness	linguistics of sense

A decisive fact should nevertheless be stressed once again: the most important plane is the individual one - the perpetual generation of sense, which constitutes the actual essence of linguistic activity. Thus, all the components (norms, devices, configurations, units etc.) belonging to the universal and idiomatic planes are taken up in individual speech as raw materials for text-constitution (Textkonstitution) and sense-construction (Aufbau des Sinns).

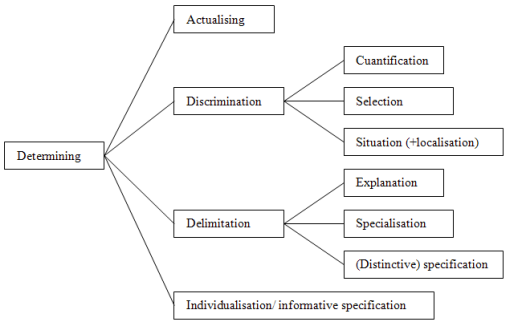
2. The Historical Level And Idiomatic Competence

The basic plane of intuitive linguistic knowledge is the historical plane. Idiomatic competence comprises all the possibilities for actual uses of language: these potentialities appear as significata. According to the Romanian linguist, these can be classified into five types: lexical, categorial, instrumental, syntactic

[2] This is how the phrase "todos sus aspectos" from the passage quoted above should be understood.

[3] For a detailed discussion and references, see [17]: 25-29, 131-133

and ontic significata ([8]: 247-249). Although the term significatum may suggest that the linguistic approach to idiomatic knowledge of the speakers should be structural, this is not always the case. Coseriu distinguishes several types of phenomena that should be discussed in 'significational' linguistics, yet cannot be treated by methods specific to structuralism (for an ouline of Coserius lexematics developed es early as the 1960s, see [15]: 47-55):



Thus, knowledge of things (pertaining to the universal level), meta-linguistic uses of speech, fixed expressions specific to idioms, as well as speakers' intuitive knowledge on the historical evolution of their language cannot be submitted to a structuralist approach [4]. Also, 'historical' language is a mixture of different dialects, socio-cultural idiomatic levels and style traditions and as a 'diatopic', 'diastratic and 'diaphasic' language, cannot be subjected to rigorous classifications. In the end, only functional language, which is not only synchronic, but also 'syntopic', 'synstratic' and 'synphasic' can be the subject matter of a taxonomic science called grammar [5]. At the level of the functional language, for instance, Coserius conception on the Gestaltung of lexical significata, manifested in the lexematic structures [6], remains to this day one of the most coherent and refined semantic models, whose explanatory and descriptive potential is far from being fully exploited.

[4]For the main principles of structuralist linguistics (functionality, opposition, systematicity and neutralising) see [6], esp. chapters VII and VIII

[5]This would comprise what is usually known as phonology, semantics, morphology and syntax (including its 'trans-phrasal' domain).

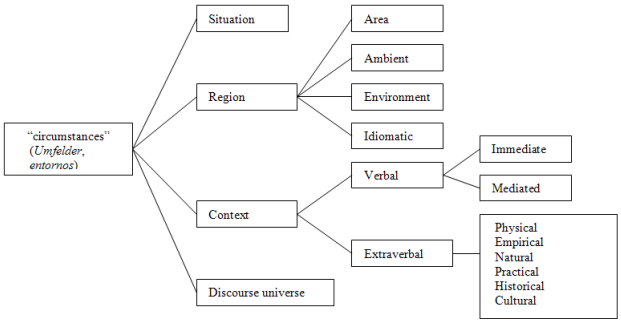
[6]Paradigmatic (oppositional) structures are classified into two subtypes: (a) primary structures (lexical field and lexical class); (b) secondary structures (modification, development, composition). Syntagmatic (combinatorial) structures or "lexical solidarities" comprise three types of relations (affinity, selection, implication).

3. The Universal Level And Elocutional Competence

The main function of language on its universal level is a cognitive one: the orientation towards 'reality' and the use of idiomatic potentialities and contextual means in order to refer to things in Coseriu's words, a 'designational' task. But we have to understand that designata can be both real objects to which we refer when using language, and objects 'projected' as real by linguistic means. This means that the distinction between 'actual' and 'fictional' things does not affect the status of a designatum: it is rather an intentional object in the phenomenological (Husserlian) sense of the word (cf. [20]: 151). There are two main devices that are used in order to orient language towards the expression of reality.

DETERMINATION

On the one hand, we can 'actualize' words from the language and combine them in order to reduce their conceptual range to particular (or even individual) things or states of affairs. Coseriu ([2]: 291-308) calls this procedure determining and identifies the following means/ categories for nominal determining:



CIRCUMSTANCES (sp. entornos, germ. Umfelder).

On the other hand, we can use the complex circumstances of speech for restraining and fixing the area of designata. Again, it is important to understand that we do not deal here with an 'adaptation' of discourse to reality; on the contrary, it is the data of reality that are used as 'raw' material for the construction of designata. Coseriu ([2]: 310-319) classifies circumstances as follows [7]:

To the universal plane of language corresponds a specific type of intuitive knowledge (elocutional linguistic competence), comprising two dimensions. The first one consists of the universal principles of thought (identity, non

[7]For a later reworking of the concept and subcategorization of the "universe of discourse", see also [14].

contradiction, non-tautology etc. cf. Coseriu [7]: 20-23, [5]: 242-243, [13]: 90-96). The second one is the general knowledge of (the normality of) things (cf. [13]: 97-107).

4. THE TEXTUAL LEVEL AND EXPRESSIVE COMPETENCE

The starting-point tenet of integral text linguistics, i.e. of Textlinguistik als Linguistik des Sinns, is the double semiotic articulation in texts ([10]: 48-51). Put simply, significata and designata of linguistic signs present or implied [8] in the text become signifiers of the second degree, signifiers for textual sense, by virtue of processes that manifest the functional autonomy of the textual plane. Beyond the simplicity of its straightforward formulation in Coserian sources, this principle has ample consequences, both theoretical and methodological. For one, let us emphasize the following: what is meant is not merely an extension or enhancement of what is actually said; it is also not mere development or explicitation of the implicit. The two are not situated on the same semantic level. What is said is a signifiant for a signifi raised to the power of two: textual sense. Also, in integral linguistics, by textual signifiant (Textkonstitution) we do not understand material units present in the text as such, but semantic functions, devices and strategies. Among the varied functions that form the textual signifiant, we have argued elsewhere ([17]: 124-133) that the highest degree of relevance for recognising the modality of sense-construction in a given text should be attributed to the following three: (1) evocative functions (relations) of signs in the text, (2) textual functions, and (3) forms of suspending (Aufhebung) incongruence and incorrectness through the value of appropriateness.

4.1. **Evocative relations.** The main types of evocative functions put forward by Coseriu ([4]: 202, [10]: 68-101, [11]: 25-29) are the following: (a) Relations of the sign to other signs, either taken individually, or as pertaining to certain categories/ groups of signs, or as sign-systems seen holistically; (b) Relations of the sign in the given text with signs from other texts (evocation of well-known texts, which belong to the linguistic and cultural tradition of a community); (c) Relations between signs and things (iconic relationship of the given textual sign with the designated object) [9] ; (d) Relations between signs and the knowledge of things, activated when the

[8]We use the term "implied textual signs" in the sense of the "Ausdruckslcke als Ausdrucksverfahren" (see [12]).

[9]We have in mind something analogous to the peculiar type of formal "imitation" of a Zen principle through the narrative configuration of a literary text, analysed in [16] and [18]: 130-141.

designata themselves are already invested with semiotic (symbolic) value in a certain cultural space, prior to their usage as text-constitutive units in the given text.

4.2. **Textual functions.** Textual functions reflect the purport of speech in a determined situation. A tentative list includes explicit and implicit functions such as assertion, question, insinuation, joke, illustration, denial etc. ([10]: 45-47, 170-174).

4.3. **Suspension of incongruence and incorrectness.** The very distinction between congruent and incongruent utterances, as well as the one between correct and incorrect sentences, can be rendered inactive if the speaker so wishes in order to achieve a defined goal in his/her individual use of language. For instance, the speaker can willingly simplify rules of his/ her language in order to be more easily understood by a foreigner; or he/she can give a give a metaphoric, meta-linguistic or even extravagant sense to the references of his/her text-discourse:

5. Conclusion

The focal underlying idea of the present paper is that, in the highly heterogeneous panorama of contemporary approaches to language, Coserius integral linguistics offers one of, if not the most comprehensive and finely articulated conceptual framework for investigating the structured totality of linguistic phenomena from the standpoint of speech activity This framework allows for an accurate positioning of each particular investigation in relation to the diverse aspects and levels of linguistic organization, on a unitary basis. At the same time, the conceptual distinctions this framework provides, grounded in the reality of the speakers intuitive knowledge of language, can prove of interest for the area of formal approaches to language as well.

References

[1] M. Ciubancan: Japanese Causative Constructions - Where to Find Them?, in "Studia Universitatis Babes-Bolyai", Philologia, nr. 1, 2006, pp. 87-97.
[2] E. Coseriu: Determinacin y entorno. Dos problemas de una lingistica del hablar, 1955-56, in Coseriu 1962, pp. 282-323.
[3] E. Coseriu: Teoria del lenguaje y lingistica general. Cinco estudios, Madrid: Gredos, 1962.
[4] E. Coseriu: Tesis sobre el tema "lenguaje y poesa", 1971/1977, in Coseriu 1977, pp. 201-207.
[5] E. Coseriu: La situacin en la lingistica, 1973/1977, in Coseriu 1977, pp. 240-256.
[6] E. Coseriu: Lezioni di linguistica generale, Torino, 1973; extended and revised Spanish version: Lecciones de lingstica general, Madrid, 1981.

[7] E. Coseriu: Lgica del lenguaje y lgica de la gramtica, 1976, in Coseriu 1978, pp. 15-49.
[8] E. Coseriu: El hombre y su lenguaje. Estudios de teora y metodologa lingstica, Madrid: Gredos, 1977.
[9] E. Coseriu: Gramtica, semntica, universales. Estudios de lingstica funcional, Madrid: Gredos, 1978.
[10] E. Coseriu: Textlinguistik. Eine Einfhrung, Tbingen: Narr, 1981.
[11] E. Coseriu: Acerca del sentido de la enseanza de la lengua y literatura, in Innovacin en la enseanza de la lengua y la literatura, Madrid, Ministerio de Educacion y Ciencia, 1987, pp. 13-32.
[12] E. Coseriu: Die Ausdruckslcke als Ausdrucksverfahren (Textlinguistische bung zu einem Gedicht von Kavafis), in Stuttgarter Arbeiten zur Germanistik, nr. 189, "Sinnlichkeit in Bild und Klang". Festschrift fr Paul Hoffman zum 70. Geburstag, 1987, pp. 373-383.
[13] E. Coseriu: Sprachkompetenz. Grundzge der Theorie des Sprechens, Tbingen: Francke, 1988.
[14] E. Coseriu. La preghiera come testo, in I quattro universi di discorso. Atti del Congresso Internazionale Orationis Millenniu
[15] E. Coseriu. and Geckeler, H: Trends in Structural Semantics, Tbingen: Narr, 1981.
[16] Tamianu-Morita, E: On a Sense-Constitutive Sign Relation: Evocation of Japanese Significations in an English Text, in "Studia Universitatis Babes-Bolyai", Philologia, XLIII, 4, 1988, pp. 75-84.
[17] Tamianu, E: Fundamentele tipologiei textuale. O abordare n lumina lingvisticii integrale, Cluj-Napoca: Clusium, 2001.
[18] Tamianu-Morita, E: Integralismul n lingvistica japoneza. Dimensiuni - impact - perspective, Cluj-Napoca: Clusium, 2002.
[19] Vlcu, C: Eugeniu Coseriu si 'rasturnarea lingvistica': o (noua) deschidere spre postmodernitate, in "Studia Universitatis Babes-Bolyai", Philologia, XLVI, 4/2001, pp. 117-128.
[20] Vlcu, C: De la semnificat la designat. Excurs despre Logos semantikos in "Dacoromania", serie noua, VII-VIII, 2002-2003, pp.141-157.
[21] Vlcu, D: Integralism vs. generativism - o dezbatere metodologica, in "Studia Universitatis Babes-Bolyai", Philologia, XLVI, 4/2001, p. 35-45.
[22] Vlcu, D: Integralism vs. generativism - schita a unei confruntari, in Un lingvist pentru secolul XX, Ed. Stiinta, Chisinau, 2002, p. 56-62.

[1] DEPT. OF GENERAL LINGUISTICS AND SEMIOTICS, UNIVERSITY "BABES-BOLYAI" CLUJ-NAPOCA
E-mail address: etamaian@lett.ubbcluj.ro

[2] DEPT. OF GENERAL LINGUISTICS AND SEMIOTICS, UNIVERSITY "BABES-BOLYAI" CLUJ-NAPOCA
E-mail address: cornel_vilcu@yahoo.co.uk

[3] DEPT. OF GENERAL LINGUISTICS AND SEMIOTICS, UNIVERSITY "BABES-BOLYAI" CLUJ-NAPOCA
E-mail address: mciubancan17@yahoo.com

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 81–88

ENHANCING THE INVISIBLE WEB

LUCIAN HANCU[(1)]

Abstract. In recent years, a large amount of information has been placed in databases across the globe, and published through dynamically generated Web pages. The evolution of the so-called Invisible (or Hidden) Web constitutes both an opportunity and an issue for Web-based information extractors. This article describes the architecture of an Invisible-Web Extractor, whose primal goal is to enhance the value of the hidden Web data. We consider three main issues of the tool: how to access the Invisible Web information, how to extract information from the gathered data and how to create new knowledge from it.

1. Introduction

During the last decade, the Web has become a primal source of information, which exhibits various forms of content: personal or business Web pages, news aggregators, large collections of music or videos. The more its content evolves and varries, the most difficult becomes the design and implementation of automatic tools that discover it, in order to index it and make that content available by use of search interfaces or to extract page snippets with the purpose of creating a more valuable Web material.

In [10], the authors classify the various portions of the Web, by considering two dimensions: whether the pages are public or private and whether the pages are static or dynamic (automatically generated by a script). Today's search engines index only public static pages and public dynamic pages, whose parameters are known or not required, thus leaving undiscovered a large amount of potential indexable information.

The total amount of indexed material is only a small fraction of the entire available Web data. As mentioned in [Table 1], private pages are not easily trackable and indexable, as they require login credentials [10]. Discovering the appropriate parameters (where they are required to correctly gather the Web pages) is a crucial task, as the missing or misleading of only one of the expected parameters can

2000 *Mathematics Subject Classification.* 68U15, 68U16.
Key words and phrases. Data Mining, Information Extraction, Invisible Web.

©2007 Babeş-Bolyai University, Cluj-Napoca

Page availability	**Page producer**		
	Static pages	Dynamic pages	
		Params known	Params unknown
Private	*Requires login*		
Public	Indexable by search engines		Requires domain-specific data

TABLE 1. Indexable Web - a small fraction of the entire Web contents

cause undesirable behaviour of the script which materialize the Web page, making impossible its correct collecting by the Web agent.

In this paper, we discuss all the steps of the roadmap to the exploitation of the Invisible Web material. We begin by describing various approaches to the discovering of the information hidden behind search forms and present our technique together with the motivation of applying it. The third section investigates the structure of the collected information and propose a solution to extract valuable information from the Web pages. The fourth section examines the extracted material and describes how to create new knowledge based on the hidden Web data and its practical usage. We conclude by presenting a number of issues we found during our experiments and propose alternative methods to be explored in future work.

2. Discovering the hidden material

In the previous works [1, 3], we have investigated two approaches for discovering and providing the appropriate parameters to be filled in a Web form: the first is a semi-automatic tool for specifying input parameters for the URLs of the dynamic pages that need to be downloaded and indexed, which relies on the information from local databases to instantiate the parameters and produce the pages [3]. A second approach consists in applying program analysis techniques on the source code of the scripts which generate Web pages in order to derive dependencies between Web page's input parameters and columns from the data repositories. After extracting these dependencies, an automated tool could simply collect all the possible values for each input parameter - as a finite set of values - and materialize all the possible Web pages obtained by instantiating the parameters with those values [1].

These two approaches illustrate a participatory vision, in which the tool responsible with the gathering of the Web material has access to local databases and the source of the scripts in order to retrieve information from them. In contrast, the black box model considers that the Web agent which collects the Web pages does not have the credentials to access local databases or the source of the scripts. Theese black-box Hidden Web crawlers apply form analysis tools, discover

common filling patterns or make use of heuristics to collect pages hidden behind search forms [5, 7, 10].

In this article, we consider the case of both private and public dynamically generated Web pages which can be gathered by the use of background knowledge. Our model consists in extracting information from an Invisible Web source, then using that information for instantiating parameters to a second source. Both sources of information are not indexable (i.e. invisible) by classical Web agents.

In our model, the first Web source contains identification data on Romanian business entities (such as fiscal ID, name of the entity, location, status), whereas the latter Web source contains financial data (financial statements on the last financial years). The sources are invisibles to the search engines, as the accession of the first one requires login credentials (we use a limited guest account which displays the minimum required information to be used in gathering content from the second Web source), while the accession to the latter Web source expects the input of the financial ID of each entity. The purpose of performing these steps is building new knowledge based on the data extracted from the Invisible Web sources.

Here are the basic steps our tool performs:

1. Collect and extract information from the first source
   1.1. Authenticate to the Web server using background knowlegde (login credentials)
   1.2. Extract the first page (comprising the total number of results)
   1.3. Navigate through the results of the query
   1.4. Extract information from the Web pages collected during the previous step (1.3).
2. Apply extracted information to the instantiation of the second source parameters, then gather data.
3. Create new knowledge from both invisible Web sources.

In the discovery process, the access to the pages is crucial, thus we use a semi-automatic approach, which consists in: a manual visit of *the login page* (for extracting login credentials), of *the query page* (for configuring the discovery tool) and of *the first result page*, which contains the number of results and links to the subsequent results pages, followed by the launch of the tool. The manual configuration of the tool is preferable to any automatic approach, as the information extracted from the first (or *base*) source shall be used in gathering Web material from the second (or *target*) source.

A fully automatic composition of input parameters could imply errors in downloading information from the first Web source, then missing downloadable pages in the case of the *target* Invisible Web site. The *error propagation* comes out as we apply information captured from the base Web source, then build the required

list of input parameters for the downloadable scripts, and finally gather the pages from the target Web source.

3. Extracting Information from the Invisible Web

Once a large amount of information is collected in a local data store, automated tools index it and republish it with the purpose of easily find that information as response to user queries through Web search forms. Instead of only index it (as in [1, 3]), we intend to extract data from both sources and create new knowledge that would enhance the value of the Invisible Web.

The aim of information extraction is to find relevant text in a document, that is a text segment and its associated attributes [6] or to find relationships between two distinct items of text [2]. As suggested above, this comes in contrast with the aim of information retrieval, which deals with the issue of finding relevant documents in a collection [4]. While multiple difficulties arise when extracting text from unstructured text, Web data has the advantage of comprising HTML tags which can be treated as text separators or can provide us with additional information on the data (for instance, tags like *<B>*, *<H1>* .. *<H6>* usually contain data as article titles, section of articles).

Invisible Web pages have an additional advantage of being automatically generated by a Web script, with useful material from columns of databases. We have manually investigated the material extracted from the two Invisible Web sources mentioned in the previous section and classified two different situations, in which the source renders *a single row* of the database and *multiple rows of the database*.

In the former case, the Web page is structured as follows:
$< TR >$
$< TD >$ Description of first column $< /TD >$
$< TD >$ Content of first column $< /TD >$
$< /TR >$
...
$< TR >$
$< TD >$ Description of n-th column $< /TD >$
$< TD >$ Content of n-th column $< /TD >$
$< /TR >$
whereas, in the latter case,
$< TR >$ *[Header row with description of columns]*
$< TD >$ Description of first column $< /TD >$
...
$< TD >$ Description of m-th column $< /TD >$
$< /TR >$
$< TR >$ *[Content of the first row]*
$< TD >$ Content of first column $< /TD >$
$< TD >$ Content of 2nd column $< /TD >$

...
$< TD >$ Content of m-th column $< /TD >$
$< /TR >$
...
$< TR >$ *[Content of the n-th (last) row]*
$< TD >$ Content of first column $< /TD >$
...
$< TD >$ Content of m-th column $< /TD >$
$< /TR >$

The discovery that Web pages from the same site share the same structure conducted us to applying pattern matching for extracting useful data from the documents. The patterns are manually constructed and make intense use of $< TD >$ and $< /TD >$ tags to delimit two columns of the table, or to delimit the description of one column from its contents.

4. Creating new Knowledge

Building new knowledge is the subsequent step after extracting useful material from the Invisible Web pages. We have investigated the Web sources from which to collect the pages and figured out that interesting information could be generated after inspecting related data.

We consider two Web pages P_A and P_B that contain financial information on companies C_A and C_B. We say that P_A is related to P_B if C_A *competes* with C_B, that is C_A and C_B have the same activity code (described in [8]). The *competition* is either *local* (when the two companies also share the same county of residence) or *national* (when the two companies do not share the county of residence).

Our goal in creating new information is to build the list of the first competitors (either local or national) which share the same activity code. The list also varies on a criteria like the total number of company's employees or the turnover on a specified year. Creating such synthetical information is similar to the classical Strengths, Weaknesses, Opportunities and Threatenings analysis [9]. This type of analysis can be useful for competitors in discovering the tough and weak points of the companies in the same activity domain; it can also be useful for clients of those companies in analysing their position on the local or national market, and it also provide the entrepreneurs with interesting investing opportunities (like finding sectors with weak competition, or counties with available work force). We present an example of such an analysis that is automatically generated after extracting the useful information from the available Invisible Web pages.

In [Table 2] we outline the results of quering our tool with the keywords *7221*, *employees*, *Cluj*, which returns the first 10 entities from the Cluj county whose activity code is 7221 (*The editing of software programs*). We obtain the list of the entities in descending order by the number of employees and render it considering

Entity	*Em*	Aim	Ac	Ct	Dat	Ca	Ve	Pb	Sal	Rpr	Pca
Intellisync	1.00	6			3	5	5		1	9	
Nivis	0.70	3	3	4	1	2	2	2	2		
Transart	0.65		4	5		6	6	8	3		
EBS	0.60	5	9		4	4	4		4		
ISDC	0.45	9	5	8	8	8	8	6	5		
Alfa Global	0.40		6	6	5	7	7	4	6		
Montran	0.38					1	1		7		
Recognos	0.33					9	9		8		
Arobs	0.31								9		
Ro planet	0.29										
Fortech	0.22		7	7				3			
Nethrom	0.15										
Api	0.09	7			7						
Vectorsoft	0.08										
Transylvan	0.08										
Q soft	0.08										
BNW	0.08										
Depart	0.08										
I I Studio	0.07										
Arxia	0.07										

TABLE 2. Information obtained from the Invisible Web sources

the relative number of employees (the number of employees of the current entity divided by the number of employees of the top entity).

We also render the positions of each entity by considering ten distinct criteria: *AIM* (Tangible assets), *AC* (Intangible assets), *CT* (Total capitals), *DAT* (Total debits), *CA* (Turnover), *VE* (Total income), *PB* (Gross profit), *SAL* (Employees), *RPR* (GrossProfit/TotalCapitals) and *PCA* (GrossProfit/Turnover).

This second classification orders the top nine entities on each one of the mentioned criteria. For instance, the *SAL* column points out all the nine positions of the top, whereas the other columns do not necessarily display all positions. This happens because entity's strength in a category does not guarantee a good position in any other category, with the exception of the *Turnover* and *Total income* columns.

In the illustrated example, the *Turnover* and *Total income* columns generate the same order for the listed entities. The result is expectable, as the *total income* includes the *turnover*. Furthermore, companies in various sectors do not output financial or extraordinary income, making the two cited columns publish almost the same order on companies.

5. Issues

In this section, we discuss some of the difficulties we found during our experiments and propose solutions to be explored in future work.

Information freshness: We conducted our experiment on the companies' financial data found at the end of 2004, which were published on the Internet in the late 2005. There is almost half-year delay between the availability of the information at the companies and the publishing of that information on the Internet and almost one year delay between the end of the financial exercice. Even so, the results provide the user with interesting knowledge, such as *the top companies on a certain activity domain*, *the strengths and weaknesses analysis* [9] *of the top companies*. A solution to the freshness of data would be the implementation of a participatory system in which companies upload their financial results as soon as they release them. A success of such an architecture would require a very large number of collaborating participants (almost all of the active companies) to provide us with useful material.

Extracting data on entities: We have explained in the *Information Extraction* section our approach to obtain information on entities from both the base and target sources of Web material. This approach considers that the structure of a hidden Web page is persistent over different invokes with input parameters. From this point of view, we can easily apply regular expressions in order to obtain the needed information. The problem appears when pages change their structure, making impossible the extraction of the data by use of the initial regular expressions. Introducing named-entity recognition techniques would imply a correct extraction of the places where companies reside (easily found in dictionaries), leaving uncertain the possibility to extract the name of the companies (as there is a vast variety for those names).

Extending the model to a larger scale: The primal goal of our work was to build a semi-automated tool for extracting content from the Invisible Web, considering the value of the information hidden behind search HTML forms and the possibilities to enhance it. We have investigated several Invisible Web sites and built a model formed of two Web data sources. One of the directions for future work would be to extend our model to perform the extraction from a more complex Web structure, such as a group of tens of Invisible Web sites. To extract valuable information from them, these sites have to be related, that is some part of the data gathered from one site must be used in another site (for instance, the Unique Fiscal ID of one company or the Private Numeric ID of one person). A hypothetical extention of our architecture would be a third site having the list of employees for every active company, then another site publishing personal information on people (like complete address, phone numbers). This type of extention raises privacy concerns, but it also poses other questions like how to obtain access to such sources of information. The fact that we can easily find Invisible Web sites

which contain data on companies does not guarantee the success in finding hidden Web information on persons, nor the existance of the sources in the near future. We believe that information on persons should remain private, thus limiting the possibilities of extending our model. However, we intend to extend the depicted tool to periodically collect and extract information from the sources, by applying the same techniques for the gathering, the extraction of valuable information and the creation of new knowledge as the information on both invisible Web sources changes.

6. Conclusions

We have described a model of gathering, extracting and enhancing the information from the Web pages whose content is kept in large databases and that are automatically generated as response to user queries. The results highlight the value of the information hidden behind search forms and how new information can be generated by applying pattern matching techniques on already existing Web material.

The techniques we have experimented are easily applicable to other Invisible Web sources. We are investigating the possibility to extend our model to include several related Web sources whose content is not trackable by traditional Web agents. We are also examining various data mining techiques for discovering valuable patterns or correlations on the gathered material. This would significantly improve the value of the Invisible Web, contributing to the creation of (what we call) an *Enhanced Invisible Web*.

References

[1] G. Attardi, A. Esuli, L. Hancu, M. Simi, 2004, *Participatory Search*, Proceedings of the IADIS International Conference WWW/Internet 2004, Madrid, Spain.
[2] S. Chakrabarti, *Mining the Web, Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Publishers, 2003, pp 290-295.
[3] L. Hancu, *Discovering Hidden Web Content*, Babes-Bolyai University, Graduation Paper, 2002.
[4] D. Hand, H. Manilla, P. Smyth, *Principles of Data Mining*, MIT Press, 2003, pp. 456-470.
[5] K.I. Lin, H. Chen, 2002, *Automatic Information Discovery From The Invisible Web*, International Conference on Information Technology: Coding and Computing, Nevada, USA.
[6] Manu Konchady, *Text Mining Application Programming*, Charles River Media, 2006, pp. 155-182.
[7] S. Raghavan, H. Garcia-Molina, 2001, *Crawling the Hidden Web*, Proceedings of the 27th Conference on Very Large Databases, Rome, Italy.
[8] Romanian Registry of Commerce, Nomenclator CAEN, *http://recom.onrc.ro/obco.htm*
[9] Wikipedia, *SWOT analysis*, *http://en.wikipedia.org/wiki/SWOT_analysis*.
[10] Ricardo Baeza-Yates, Carlos Castillo, 2005, *Crawling the Infinite Web*, Journal of Web Engineering.

[1] SoftProEuro Cluj-Napoca

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 89–95

CHAIN ALGORITHM USED FOR PART OF SPEECH RECOGNITION

ANDREEA-DIANA MIHIS[1]

Abstract. Dictionary base methods have the advantage that they can be applied to texts written in different languages if there exist an electronically dictionary for that specific language. As a consequence, these kinds of methods can be used to identify the part of speech of words from a text written in a single language. The principal advantage of using a dictionary base approach in part of speech recognition is that it can be applied to different languages, because it does not use a specific grammar. In this paper such a method is used to identify the parts of speech of words from a text using a chain algorithm [7] that disambiguates a text.

1. Introduction

When learning a language, the use of a dictionary is essential, even though grammar skills are not fully developed. When translating a text from a foreign language, the first step consists in identifying the corresponding word, followed by applying mostly grammar skills of the familiar language in trying to obtain the correct translation. When translating from a familiar language to a foreign language the most used method is word by word (the french mot-a-mot). From beginners point of view the dictionary is a powerfull tool, that sometimes can be used successfully to overcome most dificulties. This should be appliable to a computer. Like the beginner, the computer can use the dictionary explanation to choose the correct form of a word and to decide their speech part in a text. This method can be used to many languages, as long as there is an appropiate electronical dictionary.

Pertaining to Natural language processing, this method uses only a dictionary to identify the part of speech of a given word. The basic idea is that in the dictionary, every word will have different definitions, one or more for every part of speech of the specified word. Trying to identify the correct part of speech from

2000 *Mathematics Subject Classification.* 68T35, 68T50, 91F20.
Key words and phrases. chain algorithm, part of speech recognition.

©2007 Babeş-Bolyai University, Cluj-Napoca

the list, is almost equivalent with trying to match the definitions. And matching the definitions is done using the Chain Algorithm.

2. Chain algorithm for part of speech recognition

The chain algorithm for part of speech recognition is based on and includes the chain algorithm presented in [7].

The idea of the algorithm is to try to disambiguate all the words, in groups of three, but, because the part of speech (POS) is unknown, the word is searched in dictionary with all POS(s). If a word is not found with a POS, then the senses of that word for that POS does not exists. Finally, the POS and the sense merge from the greatest score.

The algorithm for POS recognition by disambiguating a triplet of words $w_1w_2w_3$ for Dice measure is the following:

begin
  for each POS p^{w_1} do
    for each POS r^{w_2} do
      for each POS q^{w_3} do
        begin
          for each sense $s^i_{p^{w_1}}$ do
            for each sense $s^j_{r^{w_2}}$ do
              for each sense $s^k_{q^{w_3}}$ do
                $score(i,j,k) = 3 \times \frac{|D_{p^{w_1}} \cap D_{r^{w_2}} \cap D_{q^{w_3}}|}{|D_{p^{w_1}}| + |D_{r^{w_2}}| + |D_{q^{w_3}}|}$
              endfor
            endfor
          endfor
        end
      endfor
    endfor
  endfor
  $(P, R, Q, i^*, j^*, k^*) = argmax_{(p,r,q,i,j,k)} score(i,j,k)$
  /* POS of w_1 is P, sense of P^{w_1} is $s^{i^*}_{P^{w_1}}$
  POS of w_2 is R, sense of R^{w_2} is $s^{j^*}_{R^{w_2}}$
  POS of w_3 is Q, sense of Q^{w_3} is $s^{k^*}_{Q^{w_3}}$ */
end

Here, α^w denotes the part of speech α of word w, and $s^i_{\alpha^w}$ is the sense with number i for the word w with POS α. D_{α^w} is the complete dictionary definition for word w with POS α.

For the overlap measure the score is calculated as:

$score(i,j,k) = \frac{|D_{p^{w_1}} \cap D_{r^{w_2}} \cap D_{q^{w_3}}|}{min(|D_{p^{w_1}}|, |D_{r^{w_2}}|, |D_{q^{w_3}}|)}$

For the Jaccard measure the score is calculated as:

$score(i,j,k) = \frac{|D_p w_1 \cap D_r w_2 \cap D_q w_3|}{|D_p w_1| \cup |D_r w_2| \cup |D_q w_3|}$

The POS recognition algorithm is a chain algorithm. This means that for the first triplet it is used to identify the POS(s) of all tree words, and for the following ones, only for the last word, because the POS(s) of first two are already identified. As a consequence, the complexity of the algorithm decreases:

begin
  for each POS q^{w_3} do
    begin
      for each sense $s^k_{q^{w_3}}$ do
        $score(i*, j*, k) = 3 \times \frac{|D_P w_1 \cap D_R w_2 \cap D_q w_3|}{|D_P w_1| + |D_R w_2| + |D_q w_3|}$
      endfor
    end
  endfor
  $(P, R, Q, i^*, j^*, k^*) = argmax_{(p,r,q,i,j,k)} score(i,j,k)$
  /*POS of w_3 is Q, sense of Q^{w_3} is $s^{k^*}_{Q^{w_3}}$ */
end

As it can be seen in detail in [7], some "stop" words were used to eliminate the trivial cases of glosses intersection.

3. Experimental evaluation

To test the POS recognition algorithm were used ten files from Brown corpus thanks to the possibility offered by SemCor corpus to evaluate the results. The input files were not tagged, and they contain compound words, underscored.

The algorithm was tested only for English Language, having WordNet as base dictionary, implying the following POS(s): Nouns, Verbs, Adjectives and Adverbs.

The application that was implemented to test the algorithm has as output the list of words tagged with theirs POS(s) and theirs corresponding sense, as follows: *noun#n#i*. Here the noun *noun* has the POS noun and the sense i from WordNet. If #..# were missing, than the algorithm failed to find the POS and sense of the corresponding word.

Because in SemCor exist some words without tag (Notag) in computing the precision, they and the stop words are ignored.

The final results can be seen in Figure 1 and in detail in annex.

In the upper part of the figure is displayed the precision for the whole file, ascending. The precision for every part of speech is displayed in the lower part of the figure, using the same order as for the first one. The precision for a specific POS is computed thus if a word has a specific POS, it will be recognized with the precision found in the graph. There are some words that are identified to have a specific POS, even if in that case they have another POS, but these words are not considered in computing the precision shown in the Figure 1.

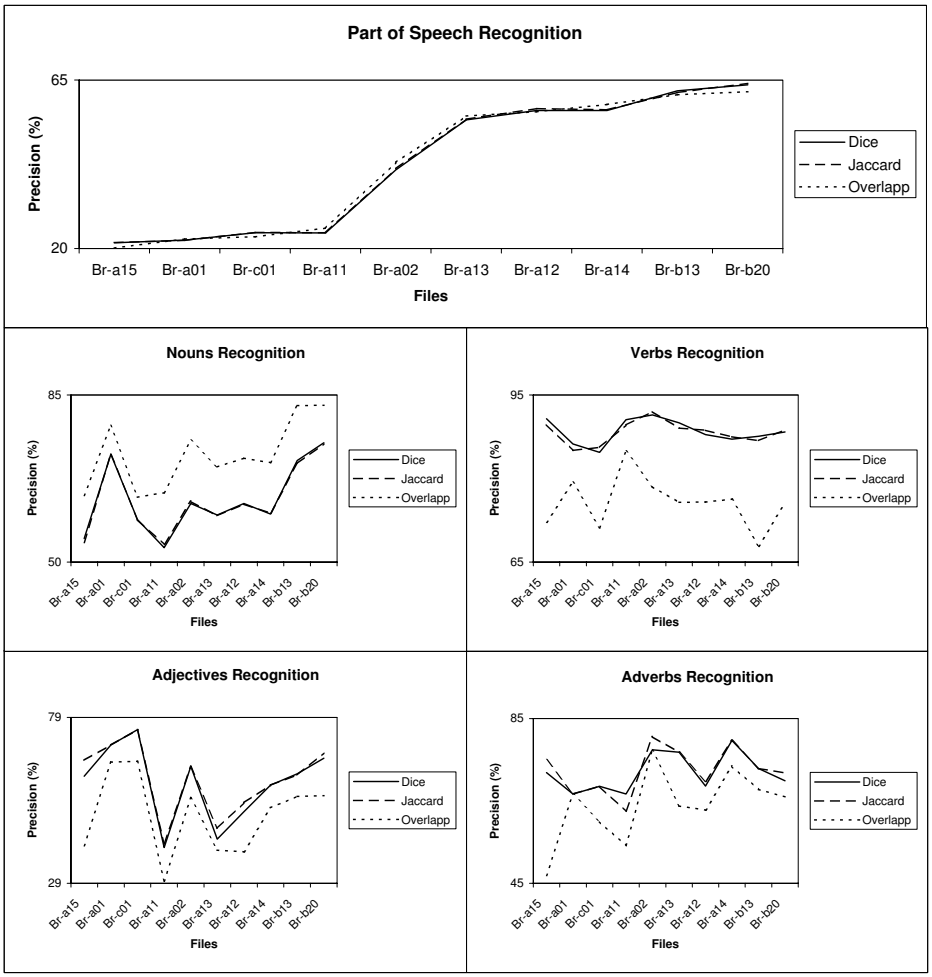


FIGURE 1. Precision of POS Recognition sorted ascending by the global precision

Because not all POS(s) were considered in applying the POS recognition algorithm, the precision for the whole file is lower then the precision for every POS. It seems that the highest precision is for verbs and the lowest one is for adjectives. In case of Nouns, the highest results were obtained using Overlapp measure, but in the other cases, with Overlapp measure the lowest results were obtained.

Table 1 summarizes the results.

Precision	Minimum Value	Maximum Value	Average Value
POS Recognition	20.16	64.13	42.78
Disambiguation	7.00	41.43	21.78
Nouns	53.04	82.83	66.12
Verbs	67.83	92.02	84.45
Adjectives	29.55	75.34	56.39
Adverbs	46.87	80.65	69.41

TABLE 1. Minimum, Maximum and Average Value for Precision

The Precision for disambiguating the whole text is not great, only 21,78% in average. But the precision for a specific POS is by average more then 68%.

4. Application of POS recognition

An application of POS recognition algorithm is in Entity-Relation diagrams automatic construction.

To create an Entity-Relation diagram from a specification written in a specified natural language, first step is to identify the nouns - Entities and verbs - Relations.

For instance, for specification: *A driver drives cars* the following result is obtained:

$Word : driver\ Driver\#n\#1 : driver$

$Word : drives\ Drives\#v\#28 : drive, take$

$Word : cars\ Cars\#n\#2 : car, railcar, railwaycar, railroadcar$

The corresponding Entity-Relation Diagram will have as Entities nouns: *driver* and *car*, and as Relation the verb *drives*.

Next step is to identify Attributes for the Entities. This can be done by searching for the adjectives which determine the noun, and by searching for the nouns bounded to the Entity noun with an membership verb, as in the following: $APeoplehasanAddress$. *Address* is an Attribute for the Entity *Pupil*.

5. A short compression to another related methods

Most of the methods found in the literature that deals with part of speech identification, or word tagging, are designed only for a single language. They use grammatical notions, corpuses and artificial intelligence techniques such as training on a part of the target text. They have very good results: more than 90% precision.

The advantage of the proposed method is that it is not using grammatical concepts, and so, it can be used for more than one language. And because of this, if a bilingual dictionary is available, the algorithm can be used to identify the language of every word.

6. Conclusion and further work

POS recognition algorithm can be used to identify the POS(s) of words from a text written in a specified language if there is an electronically dictionary, without using grammar notions or another sources.

The precision for POS recognition can be improved. For instance, if at the beginning all the words with only one POS and only one sense are annotated, and than the CHAIN algorithm is applied ascending and descending, using the words with only one POS and sense as anchors. Another way is by not ignorring some POS(s), and this can be easily done by considering all the POS(s) found in the dictionary for the corresponding language. Another posibility is to start with the word with the longest gloss. And of course, combining the Chain algorithm with an artificial intelligence technique, to maximize the sum of all scores.

The POS recognition algorithm is a consequence of CHAIN algorithm for word sense disambiguation found in [7]. I believe that the CHAIN algorithm can have some more interesting consequences.

7. Annex

See Figure 2.

8. References

[1] S. Banarjee and T. Pedersen. 2003. *"Extended Gloss Overlaps as a Measure of Semantic Relatedness."* Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, August 9-15, Acapulco, Mexico.

[2] E. Agirre and P. Edmonds (editors). 2006. *"WSD: Algorithms and Applications."* Springer.

[3] C. Fellbaum (editor). 1998. *"WordNet An Electronic Lexical Database."* The MIT Press.

[4] D. Jurafsky and J. Martin. 2000. *"Speech and language processing."* Prentice Hall.

[5] C. Manning and H. Schutze. 1999. *"Foundation of statistical natural language processing."* MIT.

[6] D. Tatar and G. Serban. 2001. *"A new algorithm for WSD."* Studia Univ. Babes-Bolyai, Informatica, 2, 99108.

[7] D. Tatar, G. Serban, A. Mihis, M. Lupea and M. Frentiu. *"A chain dictionary method for Word Sense Disambiguation and applications"*, Proceedings of KEPT2007, to appear.

[8] http://wordnet.princeton.edu/perl/webwn

[1] Computer Science Department, Babes-Bolyai University, Kogalniceanu Street nr. 1, RO-400084, Cluj-Napoca, Romania

E-mail address: mihis@nessie.cs.ubbcluj.ro

	Br-a01			Br-a02			Br-a11			Br-a12			Br-a13			Br-a14			Br-a15			Br-b13			Br-b20			Br-c01		
	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.	Dice	Jacc.	Over.
Number of Words:	1816	1816	1816	1765	1765	1765	1843	1843	1843	1922	1922	1922	1856	1856	1856	1844	1844	1844	1908	1908	1908	1881	1881	1881	1860	1860	1860	1967	1967	1967
Correct Disambiguated:	128	128	128	217	217	342	117	118	211	319	316	502	288	295	479	287	281	456	90	94	176	334	334	514	339	338	500	128	127	191
Correct POS:	280	279	284	488	492	512	304	305	320	731	738	727	665	668	678	663	665	682	277	278	259	784	778	771	769	774	747	312	313	298
Notag in WordNet:	30	30	30	61	61	61	37	37	37	110	110	110	76	76	76	99	99	99	69	69	69	93	93	93	130	130	130	77	77	77
Nouns:	215	215	215	408	408	408	296	296	296	570	570	570	578	578	578	541	541	541	233	233	233	505	505	505	460	460	460	236	236	236
Correct nouns:	156	156	169	254	256	308	157	159	191	355	354	409	346	345	404	325	326	383	128	126	149	360	357	418	345	344	381	139	139	150
False nouns:	31	32	43	59	60	89	69	70	99	131	128	213	130	127	204	125	124	187	42	42	84	121	120	210	107	102	187	37	37	75
Verbs:	87	87	87	163	163	193	106	106	196	256	256	294	210	210	222	203	203	232	87	87	87	263	263	298	253	253	258	111	111	111
Correct verbs:	75	74	87	149	150	128	96	95	90	225	227	194	189	187	159	202	203	177	87	87	70	226	224	175	228	229	193	94	95	79
False verbs:	45	45	29	101	98	50	71	71	44	123	123	62	123	123	124	130	131	65	53	55	30	152	155	93	102	102	63	50	50	31
Adjectives:	58	58	58	95	95	95	88	88	88	182	182	182	182	182	182	145	145	145	62	62	62	250	250	250	186	186	186	73	73	73
Correct adjectives:	41	41	38	61	61	52	35	36	26	92	97	70	77	83	71	85	85	75	38	41	25	155	154	138	124	127	103	55	55	48
False adjectives:	9	9	11	21	18	13	26	25	16	50	47	46	45	45	37	45	42	38	9	8	15	30	34	21	52	50	42	28	27	23
Adverbs:	12	12	12	31	31	31	24	24	24	86	86	86	69	69	69	64	64	64	32	32	32	59	59	59	103	103	103	35	35	35
Correct adverbs:	8	8	8	24	25	24	16	15	13	59	60	54	53	53	44	51	51	47	23	24	15	43	43	40	72	74	68	24	24	21
False adverbs:	4	4	2	15	16	10	26	25	17	35	34	22	44	43	25	27	28	18	20	18	9	32	32	24	18	20	9	2	2	2
Ignored:	528	528	528	514	514	514	548	548	548	526	526	526	556	556	556	579	579	579	554	554	554	526	526	526	523	523	523	603	603	603
Statistics:																														
Words Not Ignored:	1258	1258	1258	1190	1190	1190	1258	1258	1258	1286	1286	1286	1224	1224	1224	1166	1166	1166	1285	1285	1285	1262	1262	1262	1207	1207	1207	1287	1287	1287
Precision for POS Recognition (%):	22.3	22.2	22.6	41	41.3	43	24.2	24.2	25.4	56.8	57.4	56.5	54.3	54.6	55.4	56.9	57	58.5	21.6	21.6	20.2	62.1	61.6	61.1	63.7	64.1	61.9	24.2	24.3	23.2
Precision for Disambiguation (%):	10.2	10.2	15.3	18.2	18.2	28.7	9.3	9.38	16.8	24.8	24.6	39	23.6	24.1	39.1	24.6	24.1	39.1	7	7.32	13.7	26.5	26.5	40.7	28.1	28	41.4	9.95	9.87	14.8
Precision for Nouns (%):	72.6	72.6	78.6	62.3	62.7	75.5	53	53.7	64.5	62.3	62.1	71.8	59.9	59.7	69.9	60.1	60.3	70.8	54.9	54.1	63.9	71.3	70.7	82.8	75	74.8	82.8	58.9	58.9	63.6
Precision for Verbs (%):	86.2	85.1	79.3	91.4	92	78.5	90.6	89.6	84.9	87.9	88.7	75.8	90	89	75.7	87.1	87.5	76.3	90.7	89.7	72.2	87.6	86.8	67.8	88.4	88.8	75.6	84.7	85.6	71.2
Precision for Adjectives (%):	70.7	70.7	65.5	64.2	64.2	54.7	39.8	40.9	29.5	50.5	53.3	38.5	42.3	45.6	39	58.6	58.6	51.7	61.3	66.1	40.3	62	61.6	55.2	66.7	68.3	55.4	75.3	75.3	65.8
Precision for Adverbs (%):	66.7	66.7	66.7	77.4	80.6	77.4	66.7	62.5	54.2	68.6	69.8	62.8	76.8	76.8	63.8	78.7	79.7	73.4	71.9	75	46.9	72.9	72.9	67.8	69.9	71.8	66	68.6	68.6	60

	MIN	MAX	AVG
Statistics:			
Words Not Ignored:			
Precision for POS Recognition (%):	20.2	64.1	42.8
Precision for Disambiguation (%):	7	41.4	21.8
Precision for Nouns (%):	53	82.8	66.1
Precision for Verbs (%):	67.8	92	84.2
Precision for Adjectives (%):	29.5	75.3	56.4
Precision for Adverbs (%):	46.9	80.6	69.4

FIGURE 2. Results

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 96–103

NATURAL LANGUAGE GENERATION: APPLICATIONS FOR ROMANIAN LANGUAGE

CIPRIAN-IONUT DUDUIALĂ [(1)]

Abstract. Natural Language Generation (NLG) and Foreign Language Writing Aid (FLWA) are two important tasks of Natural Language Processing (NLP), which deal with obtaining natural language from a machine representation system and building computer programs that assists a non-native language user in writing decently in a target language, respectively. This paper uses both NLG and FLWA. Suppose a person wants to translate a sentence from English to Romanian language, but he/she does not speak Romanian. The first thing to be done is to take a dictionary, find the corresponding words, put them together and form the sentence, but a lot of disambiguities might arise. Using an affix grammar to construct the Romanian language grammar and a semantic which gives us information about the words we use to build a sentence, we can construct, starting from a set of words, correct sentences from sintactic and semantic point of view. This paper deals only with short sentences.

1. Introduction

Translating a sentence from English to Romanian means more than translating word with word using a dictionary . We risk to obtain phrases that make no sense.

First of all, the words order in a phrase is different. *"Is Dory online?"* is translated *"Dory este online?"* due to the fact that in Romanian language the subject appears before the predicate even when a question is constructed. Next, in English, for some tenses of verbs, the same form is used for different persons, while Romanian language uses different forms. In addition, a dictionary does not contain the verb forms for every tense and person. Another ambiguity can be generated by an adjective that determines a noun, since in English the adjectives have only one form. They do not depend on the genre or number of the substantive, but Romanian language uses different forms for adjectives.

This paper takes into consideration only short propositions containing a subject, a predicate and, if needed, some complements. The subject and the complements can be followed by at most one attribute. Conjuctions like *"si"* or *"sau"* and

Key words and phrases. natural language generation, affix grammar.

©2007 Babeş-Bolyai University, Cluj-Napoca

punctuation marks will be omitted. This means the propositions cannot have a multiple subject.

2. Theoretical Support

The main idea comes from [1], where models for source code generators are created. Starting from a language grammar and having some logical relations and simple statements as input, valid source code for a given programming language is obtained. In what follows, the simplest model from that paper is analyzed.

Let $G = \{\{S, C\}, \{\Delta, |-, \varphi, \Omega, a, b, c_1, c_2, (,), ", "\}, P, S\}$ be the proper grammar with the set of productions P:

$$\begin{aligned} S &\rightarrow (S,S)||-(C,S)|\Delta(C,S,S)|\varphi(C,S)|\Omega(S,C)|a|b \\ C &\rightarrow c1|c2 \end{aligned}$$

In the grammar definiton, the fundamental control structures are encoded: *"(,)"* means *"concatenation"*, *"|−"* means *"if whith one branch"*, *"Δ"* means *"if with two branches"*, *"φ"* means *"while"* and *"Ω"* means *"repeat"*. Let a and b be two fundamental structures and logical expressions. Then:

- $a < b$, if a always appears before b in the generated programs;
- $a > b$, if a always appears after b in the generated programs.

In the following example, a and b are fundamental structures, while *c1* and *c2* are logical expressions:

	a	b	c_1	c_2
a	-	$<$	$<$	$<$
b	$>$	-	$<$	$<$
c_1	$>$	$>$	-	$<$
c_2	$>$	$>$	$>$	$>$

Table 1. Source code generators - semantic example.

The matrix is anti-symmetric, thus the defined model is consistent. For the above example, the set of words produced by applying exactly one production which does not involve only terminal symbols is reduced to S={(a, b), (a, c1), (b, c1), (a, c2), (b, c2)}. The programs are equivalent with the Pascal programs:

```
a;  REPEAT a   REPEAT b   REPEAT a   REPEAT b
b;  UNTIL c1;  UNTIL c1;  UNTIL c2;  UNTIL c2;
```

A similar model solves our problem. An appropriate grammar helps building correct sentences from syntactical point of view. More than that, keeping records about the forms of the words (substantives, adjectives, verbs, pronouns, adverbs and prepositions) for singular and plural and for all three genres is useful to solve the ambiguities that might appear.

3. Romanian Language Grammar

This section introduces the Romanian language grammar, which is a simple grammar used to illustrate some examples. It can be enriched with other productions to reflect all forms of the sentences that can be created using Romanian words. An affix grammar with restrictions [2] is used, which is in fact a context free grammar with finite set-valued features, acceptable to linguists. A simple example is $G = \{\{A, B, C\}, \{a, b, c\}, P1, A\}$, with the set of productions *P1*:

$$\begin{aligned}
param &:: one; two. \\
A &\rightarrow (\{param :: one\}|\{param :: two\}), B(param) \\
A &\rightarrow (\{param :: one\}|\{param :: two\}), C(param) \\
B(one) &\rightarrow a \\
B(two) &\rightarrow b \\
C(param) &\rightarrow c
\end{aligned}$$

The restrictions are introduced through some parameters. In our case, *"param"* is the parameter, while *"one"* and *"two"* are the parameter values and are introduced by *"::"*. For grammar G, the conditions appear before rewriting and the following expressions are generated:

$$\begin{aligned}
A \rightarrow B(one) \rightarrow a \qquad & A \rightarrow B(two) \rightarrow b \\
A \rightarrow C(one) \rightarrow c \qquad & A \rightarrow C(two) \rightarrow c
\end{aligned}$$

which means that *C(param)* can be applied for all values of *param*. Suppose now that the conditions are after the rewriting, that is:

$$\begin{aligned}
param &:: one; two. \\
A &\rightarrow B(param), (\{param :: one\}|\{param :: two\}) \\
A &\rightarrow C(param), (\{param :: one\}|\{param :: two\}) \\
B(one) &\rightarrow a \\
B(two) &\rightarrow b \\
C(param) &\rightarrow c
\end{aligned}$$

In this case, $A \rightarrow B(param)$ is applied first. For all the values that param can take according to the condition imposed to this production, we will rewrite *B(param)*. Applying the same principle for $A \rightarrow C(param)$ we obtain:

$$\begin{aligned}
A &\rightarrow B(param) \rightarrow B(one) \rightarrow a \\
A &\rightarrow B(param) \rightarrow B(two) \rightarrow b \\
A &\rightarrow C(param) \rightarrow C(one) \rightarrow c \\
A &\rightarrow C(param) \rightarrow C(two) \rightarrow c
\end{aligned}$$

Next, the affix grammar for the Romanian language is created to simplify things a little from notation point of view. Let

$$\begin{aligned} GR \;=\; & \{\{SR, Prop, Substantiv, Subst\_Atr, Atribut, Predicat, Adverb, \\ & ListComplAdv, ListCompl, Compl\}, \{adjectiv(gen, nr), prepozitie, \\ & substantiv(gen, nr, caz, tip\_articol), pron\_pos(pers, genp, nrp, gen, nr), \\ & pronume(pers, gen, nr, caz), verb(pers, nr, timp), adverb, \\ & verb\_gerunziu, verb\_infinitiv\}, PR, SR\} \end{aligned}$$

be the Romanian language grammar. The parameters and their values for GR are:

$$\begin{aligned} nr,\ nrp &:: sing, pl. \\ gen,\ genp &:: masc, fem, neutru. \\ pers &:: I, II, III. \\ timp &:: prezent, viitor, perf\_comp, imperf, mmcperf, conj, cond\_opt. \\ tip\_subiect &:: subst, pron. \\ tip\_articol &:: hot, nehot. \\ caz &:: N, Ac, D, G. \end{aligned}$$

The set of productions PR is defined as follows:

$$SR \;\rightarrow\; (\{tip\_subiect :: subst\}|\{tip\_subiect :: pron\}), Prop(tip\_subiect)$$

$$\begin{aligned} Prop(subst) \;\rightarrow\; & ((\{gen :: masc\}|\{gen :: fem\}|\{gen :: neutru\})\&(\{nr :: sing\}| \\ & \{nr :: pl\})\&(\{tip\_articol :: hot\}|\{tip\_articol :: nehot\})), \\ & substantiv(gen, nr, N, tip\_articol)\ Predicat(III, gen, nr) \end{aligned}$$

$$\begin{aligned} Prop(subst) \;\rightarrow\; & ((\{gen :: masc\}|\{gen :: fem\}|\{gen :: neutru\})\&(\{nr :: sing\}| \\ & \{nr :: pl\})), Subst\_Atr(gen, nr, N) \\ & Predicat(III, gen, nr) \end{aligned}$$

$$\begin{aligned} Prop(pron) \;\rightarrow\; & ((\{gen :: masc\}|\{gen :: fem\})\&(\{nr :: sing\}|\{nr :: pl\})\& \\ & (\{pers :: I\}|\{pers :: II\}|\{pers :: III\})), \\ & pronume(pers, gen, nr, N)\ Predicat(pers, gen, nr) \end{aligned}$$

$$\begin{aligned} Subst\_Atr(gen, nr, caz) \;\rightarrow\; & (\{tip\_articol :: hot\}|\{tip\_articol :: nehot\}), \\ & substantiv(gen, nr, caz, tip\_articol)\ Atribut \end{aligned}$$

$$\begin{aligned} Subst\_Atr(gen, nr, caz) \;\rightarrow\; & (\{tip\_articol :: hot\}|\{tip\_articol :: nehot\}), \\ & substantiv(gen, nr, caz, tip\_articol) \\ & adjectiv(gen, nr) \end{aligned}$$

$$Subst\_Atr(gen, nr, caz) \rightarrow (((\{tip\_articol :: hot\}|\{tip\_articol :: nehot\})\&$$
$$(\{genp :: masc\}|\{genp :: fem\})\&$$
$$(\{nrp :: sing\}|\{nrp :: pl\})\&$$
$$(\{pers :: I\}|\{pers :: II\}|\{pers :: III\})),$$
$$substantiv(gen, nr, caz, tip\_articol)$$
$$pron\_posesiv(pers, genp, nrp, gen, nr)$$

$$Atribut \rightarrow (((\{gen :: masc\}|\{gen :: fem\}|\{gen :: neutru\})\&(\{nr :: sing\}|$$
$$\{nr :: pl\})\&(\{caz :: Ac\}|\{caz :: G\})\&(\{tip\_articol :: hot\}|$$
$$\{tip\_articol :: nehot\})), [prepozitie]$$
$$substantiv(gen, nr, caz, tip\_articol)$$

$$Atribut \rightarrow (((\{gen :: masc\}|\{gen :: fem\})\&(\{nr :: sing\}|\{nr :: pl\})\&$$
$$(\{pers :: I\}|\{pers :: II\}|\{pers :: III\})), prepozitie$$
$$pronume(pers, gen, nr, Ac)$$

$$Predicat(pers, gen, nr) \rightarrow (\{timp :: prezent\}|\{timp :: viitor\}|\{timp :: conj\}|$$
$$\{timp :: perf\_comp\}|\{timp :: mmcperf\}|$$
$$\{timp :: imperf\}|\{timp :: cond\_opt\}),$$
$$verb(pers, nr, timp)\ [ListComplAdv(gen, nr)]$$

$$ListComplAdv(gen, nr) \rightarrow [ListComplAdv(gen, nr)]Adverb(gen, nr)$$
$$[ListComplAdv(gen, nr)]|ListCompl$$

$$Adverb(gen, nr) \rightarrow adjectiv(gen, nr)|adverb|verb\_gerunziu$$

$$ListCompl \rightarrow Compl|Compl\ ListCompl$$

$$Compl \rightarrow prepozitie\ verb\_infiniv$$

$$Compl \rightarrow (((\{gen :: masc\}|\{gen :: fem\}|\{gen :: neutru\})\&(\{nr :: sing\}|$$
$$\{nr :: pl\})\&(\{caz :: Ac\}|\{caz :: D\})\&(\{tip\_articol :: hot\}|$$
$$\{tip\_articol :: nehot\})), [prepozitie]$$
$$substantiv(gen, nr, caz, tip\_articol)$$

$$Compl \rightarrow (((\{gen :: masc\}|\{gen :: fem\}|\{gen :: neutru\})\&(\{nr :: sing\}|$$
$$\{nr :: pl\})\&(\{caz :: Ac\}|\{caz :: D\})), [prepozitie]$$
$$Subst\_Atr(nr, caz)$$

$$Compl \rightarrow (\{gen :: masc\}|\{gen :: fem\})\&(\{nr :: sing\}|\{nr :: pl\})\& (\{pers :: I\}|\{pers :: II\}|\{pers :: III\})\&(\{caz :: Ac\}|\{caz :: D\})), [prepozitie]pronume(pers, gen, nr, caz)$$

4. Simple Model for NLG Using Romanian Language Grammar

Using a semantic the amount of computations needed to determine a correct proposition is reduced. Each word used is considered an element of a set and a relation between the words is defined. The phrase: *"I always read the weather forecast in the newspapers"* is translated into Romanian *"Eu citesc intotdeauna previziunile meteo in ziare"*. Translating word by word, using a dictionary, the user obtains the set of words $S = \{eu, intotdeauna, a\ citi, vremea, previziuni, ziare\}$ and establishes, for example, the following relations between them:

	eu	a citi	intotdeauna	vremea	previziuni	ziare
eu	-	-	-	-	-	-
a citi	-	-	Pr	-	Pr	Pr
intotdeauna	-	C	-	-	-	-
vremea	-	-	-	-	A	-
previziuni	-	C	-	Wd	-	-
ziare	-	C	-	-	-	-

Table 2. NLG - simple model example.

In the above table, each entry has the following meaning:

- '-' → the 2 elements are not related
- 'Pr' → the word from the line is the predicate determined by the complement from the column
- 'C' → the word from the line is the complement which determines the verb from the column
- 'A' → the word from the line is the attribute which determines the word from the column
- 'Wd' → the word from the line is the word determined by the attribute from the column

Observe that the predicate of the proposition can be easily determined, since for sure one line has one or several of the entries with value *'Pr'*. Note also that a semantic has at least two words due to the fact that a subject and a predicate are needed for any proposition. Anyway, another question needs an answer: how can be determined the person and the number of a pronoun, for example? For that a database - like a dictionary (DEX) - that gives us information about words is used.

5. Extension of the Simple Model

The paper purpose is not to find new methods for NLG. It proves that using the simple idea of the models created in [1] and given a set of words correct sentences can be constructed using only those words. In our case, not only the semantic is important: creating propositions that make sense cannot be done at random. Not any adjective can be used to describe a substantive and not any adverb can be used together with a certain verb. Thus, the database needs some links between words showing if it makes sense to use them together - for example, specifying that *"munte"* can be used with the adjective *"imens"*, but makes no sense to be used with adjectives like *"scund"*. For the models defined in this paper, some problems appear for substantives in *dative* case if the links are not defined. All the forms of a substantive or adjective are also need - for different genres and numbers - and all the forms of a verb - for different tenses and persons.

Next, to reduce the number of propositions generated, the words order in the phrase is specified. We can say: *"Raul curge lin la vale"* or *"Raul curge la vale lin"* and so on. A semantic of the form of [Table 3] indicates which word can be in front of another and which can not be.

Definition: Let a and b be two words. Then:

- $a < b$, if a always appears before b in the generated propositions;
- $a > b$, if a always appears after b in the generated propositions;
- $a = b$, if a and b can appear in any order in the generated propositions.

Definition: A **simple semantic** is a table defining the order of words in a phrase and having the form of [Table 3].

		raul	a curge	la	vale	lin
sg	raul	-	$<$	$<$	$<$	$<$
prez	curge	$>$	-	$=$	$=$	$=$
-	la	$>$	$=$	-	$<$	$>$
sg	vale	$>$	$=$	$>$	-	$>$
adj	lin	$>$	$=$	$<$	$<$	-

Table 3. NLG - simple semantic example.

The above table shows that *"raul"* is always the first word, while *"lin"* always appears before *"vale"*. So, only *"Raul lin curge la vale"* is generated. Not specifying that *"lin"* is an adjective means it can also be considered an adverb, thus *"Raul curge lin la vale"* could also be generated. Two important issues can be deduced: (i) the prepositions can be introduced into the model, which helps avoiding inappropriate use of prepositions and (ii) some adjectives can be used as adverbs and the sense of the proposition is changed, depending on the interpretation given

to that adjective. Also, for example, the word "muncitor" can be used as adjective or substantive. Thus, specifying its function might be useful.

Another semantic that specifies which word is determined by another one can be introduced:

		raul	a curge	la	vale	lin
sg	raul	-	$<$	$<$	$<$	$< d$
prez	curge	$>$	-	$=$	=d	$=$
-	la	$>$	$=$	-	$< d$	$>$
sg	vale	$>$	d=	$> d$	-	$>$
adj	lin	$d >$	$=$	$<$	$<$	-

TABLE 4. NLG - semantic example.

In [Table 4] the letter *"d"* appears before the order sign. *"a d* b"*, with *"*"* from $\{>, <, =\}$ means a determines b, while *"a *d b"* means a is determined by b. Several combinations are possible: a substantive determined by an adjective or another substantive (attributes), for example. Moreover, a preposition appears before the word determined. Obviously, preposition means before a position, thus in the line corresponding to each preposition we have exactly once " $< d$", while in the corresponding column we have exactly once " $> d$". If this condition is not satisfied, we say that the semantic is *inconsistent*. If a word determines more than one other word the semantic is also considered *inconsistent*. The probability that in a short proposition a word is used twice is very small, but if this happens that word is added twice in the semantic.

6. Conclusions

This paper presents a way of generating senteces using a given set of Romanian words. An affix grammar for the Romanian language is introduced to ensure that the sentences are correct from syntactical point of view. Finally, in order to reduce the number of sentences generated and and to indicate the words order in a sentence, three different semantics are defined.

References

[1] Vasile Cioban, Ciprian Duduiala - *"A Case Tool Proposal for Source Code Generators"*, Proceedings of the Symposium *"Colocviul Academic Clujean de Informatica"*, pag. 147-153, 1-2 June 2006.

[2] http://www.cs.ru.nl/agfl/ - official site of AGFL formalism developed between 1991 and 1996 by the Computer Science Department of the Radboud University of Nijmegen.

(1) Centre for Mathematical Medicine and Biology, School of Mathematical Sciences, University of Nottingham, UK

E-mail address: cipriduduiala@yahoo.com

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 107–113

A HYBRID INCREMENTAL/MONTE CARLO SEARCHING TECHNIQUE FOR THE "SMOOTHING" PARAMETER OF PROBABILISTIC NEURAL NETWORKS

FLORIN GORUNESCU[(1)], MARINA GORUNESCU[(2)], KENNETH REVETT[(3)], AND MARIUS ENE [(4)]

Abstract. The only control factor that needs to be selected for Probabilistic Neural Network training to cause a reasonable amount of overlap is the smoothing parameter σ. The shape of the decision boundary can be made as complex as necessary by choosing an appropriate value of σ. It has been shown that the decision boundary varies continuously from a hyperplane to a very nonlinear surface according to σ and it has also been suggested several techniques to choose this parameter. The aim of this paper is to introduce a hybrid technique, sequentially using an incremental search as a first raw approach, followed by a Monte Carlo search to fine tune the first one. An application to a medical data concerning the hepatic cancer is also considered.

1. Introduction

The probabilistic neural networks (PNN), introduced by Specht [4], represent supervised neural networks (NN) widely used in the area of classification, pattern recognition, nonlinear mapping etc. PNN are essentially based on the well-known Bayesian classification technique, that is a strategy allowing the minimization of the expected risk. They constitute a class of NN combining some of the best attributes of statistical pattern recognition and feedforward NN, representing the neural network implementation of kernel discriminant analysis. The greatest advantages of PNN are the fact that the output is probabilistic (easy interpretation of output) and the training speed. PNN requires small training time, training PNN actually consisting mostly of copying training cases into the network. On the other hand, the main criticism of PNN is the very rapid increase in memory and computing time when the input sample dimension and the training set size increase ("curse" of dimensionality).

2000 *Mathematics Subject Classification.* 68T05, 90B15.

Key words and phrases. Probabilistic neural network, smoothing parameter, incremental search, Monte Carlo search, hybrid search.

©2007 Babeş-Bolyai University, Cluj-Napoca

The classification performance of PNN is largely influenced by the smoothing parameter σ (i.e. the radial deviation of the Gaussian functions). In this paper we propose a hybrid two-step sequentially algorithm to optimize the smoothing factor of the PNN. This technique consists in using, as a first step, an incremental search to estimate local optima of the cost function given by the percentage of well classified patterns. The second step, the fine tuning process, consists in using a Monte Carlo technique to estimate the best value of σ, in order to maximize the classification accuracy. This hybrid searching model is then applied to a medical data set, concerning the hepatic cancer.
The paper is organized as follows: in the following Section the basic concepts of PNN are described. Next, the proposed searching approach is presented. Subsequently, the experimental setup including the description of the data set and the sampling technique used are presented. In the next Section, a presentation of the obtained results is reported. The paper ends with conclusions.

2. Probabilistic Neural Networks

The PNN paradigm is based on the Bayes strategy for pattern recognition. Consider a q-category situation in which the state of nature θ is known to be θ_k, k=1, 2,..., q. If it desired to decide whether $\theta = \theta_k$ based on a set of measurements represented by the p-dimensional samples (vectors $\mathbf{x}$=$(x_1, x_2, ..., x_p)$), the Bayes decision rule formally becomes:

- Decision θ_k: "State of nature is θ_k";
- Given measurement $\mathbf{x}$ if the decision is θ_k then the error is $P(error|x) = 1 - P(\theta_k|x)$;
- Minimize the probability error;
- Bayes decision rule: "*Decide* θ_k *if* $P(\theta_k|x) > P(\theta_j|x),\ \forall j \neq k$" or, equivalently, "*Decide* θ_k *if* $P(x|\theta_k)P(\theta_k) > P(x|\theta_j)P(\theta_j),\ \forall j \neq k$"

To illustrate the way the Bayes decision rule is applied to PNN, consider the general case of a q-category classification problem, in which the states of nature are denoted by $\Omega_1, \Omega_2, ..., \Omega_q$. The goal is to determine the class (category) membership of a multivariate sample data (i.e. a p-dimensional random vector $\mathbf{x}$) into one of the q possible categories $\Omega_1, \Omega_2, ..., \Omega_q$, that is, we have to make the decision $D(\mathbf{x}) = \Omega_i, i = 1, 2, ..., q$, where $\mathbf{x}$ represents the sample (data vector). If we know the (multivariate) probability density functions $f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_q(\mathbf{x})$, associated with the categories $\Omega_1, \Omega_2, ..., \Omega_q$, the *a priori* probabilities $h_i = P(\Omega_i)$ of occurrence of patterns from categories Ω_i and the *loss* (or *cost*) parameters l_i, associated with all incorrect decisions given $\Omega = \Omega_i$, then, according to the Bayes decision rule, we classify $\mathbf{x}$ into the category Ω_i if the following inequality holds

true:

$$l_i h_i f_i(\mathbf{x}) > l_j h_j f_j(\mathbf{x}), i \neq j.$$

The boundaries between every two decision classes Ω_i and Ω_j, $i \neq j$, are given by the hypersurfaces:

$$l_i h_i f_i(\mathbf{x}) = l_j h_j f_j(\mathbf{x}), i \neq j.$$

and the accuracy of the decision depends on the accuracy of estimating the corresponding p.d.f's. only.

The key to using the Bayes decision rule to PNN is represented by the technique chosen to estimate the p.d.f's $f_i(\mathbf{x})$ corresponding to each decision class Ω_i, based upon the training samples set. The classical approach uses a sum of small multivariate Gaussian distributions, centered at each training sample, that is:

$$f_i(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}\sigma^p} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp\left(-\frac{\|x-x_j\|^2}{2\sigma^2}\right), i = 1, 2, ..., q,$$

where m_i is the total number of training patterns in Ω_i, $\mathbf{x}_j$ is the j-th training pattern from category Ω_i, p is the input space dimension and σ is the adjustable "*smoothing*" parameter using the training procedure.

Bayes decision rule: For each $\mathbf{x} \in \Omega_i$ compare $f_i(\mathbf{x})$ and $f_j(\mathbf{x})$ for all $i \neq j$, following the algorithm:

"IF $l_i h_i f_i(\mathbf{x}) > l_j h_j f_j(\mathbf{x})$ (for all $j \neq i$) THEN $\mathbf{x} \in \Omega_i$ ELSE IF $l_i h_i f_i(\mathbf{x}) \leq l_j h_j f_j(\mathbf{x})$ (for some $j \neq i$) THEN $\mathbf{x} \notin \Omega_i$"

The standard PNN training procedure requires a single pass over all the samples of the training set, rendering PNN faster to train compared to feedforward NN.

Basically, the architecture of PNN might be limited to three layers: the *input/pattern* layer, the *summation* layer and the *output* layer. Each input/pattern node forms a product of the input pattern vector $\mathbf{x}$ with a weight vector W_i and then performs a nonlinear operation, that is $\exp\left[-(W_i - \mathbf{x})(W_i - \mathbf{x})^\tau/(2\sigma^2)\right]$ (assuming that both $\mathbf{x}$ and W_i are normalized to unit length), before outputting its activation level to the summation node. Each summation node receives the outputs from the input/pattern nodes associated with a given class and simply sums the inputs from the pattern units that correspond to the category from which the training pattern was selected, that is $\sum_i \exp\left[-(W_i - \mathbf{x})(W_i - \mathbf{x})^\tau/(2\sigma^2)\right]$. The output nodes produce binary outputs by using the inequality:

$\sum_i \exp\left[-(W_i-\mathbf{x})(W_i-\mathbf{x})^\tau/(2\sigma^2)\right] > \sum_j \exp\left[-(W_j-\mathbf{x})(W_j-\mathbf{x})^\tau/(2\sigma^2)\right]$

related to two different categories Ω_i and Ω_j.

Note. Since the PNN paradigm is based on the Bayes decision rule, the binary outputs above are based on finding the maximum of all sums.

3. Hybrid incremental/Monte Carlo searching technique

The key factor in PNN is therefore the way to determine the value of σ, since this parameter needs to be estimated to cause reasonable amount of overlap. Commonly, the smoothing factor is chosen heuristically. If σ is too large or too small the corresponding probability density functions will lead to the increase in misclassification rate. Thus, too small deviations cause a very spiky approximation which cannot generalize and, on the other hand, too large deviations smooth out the details.

Although an appropriate figure is easily chosen by experiment, by selecting a number which produces a low selection error, and fortunately PNN are not too sensitive to the precise choice of smoothing factor, the smoothing parameter σ is critical for the classification accuracy. Therefore, there are some approaches to assess this important PNN issue [1], [2], [3], [5]. This work deals with the estimation of a (near) optimum value of σ using a two-step method, combining both a deterministic raw detection technique (incremental search) and a stochastic fine detection (Monte Carlo search).

The complete hybrid searching algorithm consists in three steps (sub-algorithms):

- Algorithm for estimating the searching domain D_σ, using statistical tools.
- Algorithm for raw estimating of local optima of the cost function, using an incremental search.
- Algorithm for fine estimating of the optimum value of σ, using a Monte Carlo search.

We synthesize below the three algorithms.

A. Algorithm to estimate D_σ

Input. Consider q classes of objects (p-dimensional vectors) $\Omega_1, \Omega_2, ..., \Omega_q$. Each decision class Ω_i contains a number of m_i vectors (or training patterns), that is $\Omega_i = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{m_i}\}$.

1) For each class $\Omega_i, i = 1, 2, ..., q$, compute the (Euclidian) distance between any pair of vectors and denote these distances by $d_1, d_2, ..., d_{r_i}$, where $r_i = C^2_{m_i}$.

2) For each class $\Omega_i, i = 1, 2, ..., q$, compute the corresponding average distances and standard deviations $D_i = \frac{\sum_{j=1}^{r_i} d_j}{r_i}$, $SD_i = \sqrt[2]{\frac{\sum_{j=1}^{r_i}(d_j - D_i)^2}{r_i}}$.

3) For each class $\Omega_i, i = 1, 2, ..., q$, consider the corresponding 99.7% confidence interval $I_{\Omega_i} = (D_i - 3SD_i, D_i + 3SD_i)$ for the average distances.

Output. $D_\sigma = (\cup I_{\Omega_i}) \cap R_+$ represents the searching domain for the smoothing parameter σ.

B. Algorithm for incremental search

Input. The searching domain D_σ.

1) Divide the searching domain D_σ by N dividing knots $\Delta_j, j = 1, 2, ..., N$ into $(N + 1)$ equal sectors.

2) Repeat Bayes decision rule algorithm by assigning $\sigma = \Delta_j$.

3) Compute the maximum value of the cost function.

Output. The values σ's corresponding to the local optima of the cost function.

C. Algorithm for Monte Carlo search

Input. The values σ's corresponding to the local optima of the cost function.

1) Consider heuristically a neighborhood for each σ (i.e. an interval centered in σ).

2) Generate in each interval a number M of random dividing points $\{P_1, P_2, ..., P_M\}$, uniformly distributed.

3) Repeat Bayes decision rule algorithm by assigning $\sigma = P_k, k = 1, 2, ..., M$.

4) Compute the maximum value of the cost function in each case.

Output. The value σ corresponding to the global optimum of the cost function represents the optimal value of the smoothing parameter.

Note. In the incremental search, the number of dividing knots N was chosen heuristically. It has been experimentally proven that for $N > 300$, the accuracy graph became flat, ending thus a further search. The number of training patterns that are classified in the right way represents the cost function of the PNN algorithm.

4. Experimental results

The model was fitted to real data consisting of 299 individuals (both patients and healthy people) from the Department of Internal Medicine, Division of Gastroenterology, University Emergency Hospital of Craiova, Romania. This group of individuals consists of 60 patients with chronic hepatitis (CH), 179 patients with liver cirrhosis (LC), 30 patients with hepatocellular carcinoma (HCC) and 30 healthy people (HP).

The hybrid PNN algorithm has been applied to data in order to classify the group of individuals into two categories, depending on the diagnosis type: Ω_1 = hepatic cancer (HCC) and Ω_2 = non-hepatic cancer. In this way, the physician benefits from an efficient tool, provided by this approach, seen as a computer-aided diagnostic of the hepatic cancer.

Each individual in the data set is represented by a 15-dimensional vector $\mathbf{x} = (x_1, x_2, ..., x_{15})$, where the components represent some of the most important characteristics (serum enzymes) leading to the right medical diagnosis. Concretely, x_1 = TB (total bilirubin), x_2 = DB (direct bilirubin), x_3 = IB (indirect bilirubin), x_4 = AP (alkaline phosphatase), x_5 = GGT (gamma glutamyl transpeptidase), x_6 = LAP (leucine amino peptidase), x_7 = AST (aspartate amino transferase), x_8 = ALT (alanine amino transferase), x_9 = LDH (lactic dehydrogenase), x_{10} = PI (prothrombin index), x_{11} = Gamma, x_{12} = Albumin, x_{13} = Glycemia, x_{14} = Cholesterol, x_{15} = Age.

The first step to obtain a good classification using PNN is to optimally estimate the misclassification costs and the prior probabilities. Unfortunately, there is no definitive science to obtain them and must be assigned as a specific part of the problem definition. In our practical experiment we have estimated them heuristically. Thus, as far as the costs parameters are concerned, we have considered them to be inversely proportional to the average distances D_i, that is $l_i = 1/D_i$. As concerns the prior probabilities, they measure the membership probability in each group and, thus, we have considered them equal to each group size, that is $h_i = m_i$.

To avoid overfitting, the data set was randomly partitioned into two sets: the training set and the testing set. A number of 254 persons (85%) of the initial group were withheld from the initial group for the training process. Once the optimal smoothing parameter σ was obtained using the training set, the trained PNN was applied to the testing set. The procedure was repeated 10 times and the algorithms were coded in Java for the ease of implementation, using JDBC (*Java Database Connectivity*) for data processing. In Table 1 we have displayed the results of our experiment.

Table 1. Experimental results

No. N of dividing knots	Accuracy (%)			
	Incremental search		Hybrid search	
	Training	Testing	Testing	Sigma
100	90	83	86	488.065
150	92	85	90	557.786
300	96	90	95	581.278

The gain in classification accuracy is obviously, obtained without a significant loss in speed.

5. Discussion

A hybrid searching model of the smoothing parameter for probabilistic neural networks was proposed. The proposed approach incorporates an incremental (deterministic) search for the optima of the cost function and a Monte Carlo (stochastic) search for the global optimum of the smoothing parameter. The effectiveness of this PNN-based hybrid model is assessed on a medical data set related to hepatic cancer diagnosis. We used raw data (the only data available for the experiment) and we obtained reliable results providing the PNN ability and flexibility to learn from raw examples. Implementation is relatively rapid and it is an alternative to standard statistical approaches.

To evaluate further the capabilities of this model, comparisons with other searching techniques have to be done. Further work will also include an alternative to this approach, using evolutionary algorithms for the second step, instead of the Monte Carlo simulation.

References

[1] Bolat B., Yldirim T., 2003, Performance increasing methods for probabilistic neural networks, Pakistan Journal of Information and Technology, 2(3), pp. 250-255.

[2] Georgiou V.L., Pavlidis N.G., Parsapoulos K.E., Alevizos P.D., Vrahatis M.N. 2004, Optimizing the performance of probabilistic neural networks in bioinformatics task, Proceedings of the EUNITE Conference, pp. 34-40.

[3] Gorunescu F., Gorunescu M., El-Darzi E., Ene M., Gorunescu S., 2005, Statistical Comparison of a Probabilistic Neural Network Approach in Hepatic Cancer Diagnosis, Proceedings IEEE International Conference on "Computer as a tool"- Eurocon2005, Belgrade, Serbia, pp. 237-240.

[4] Specht D.F., 1990, Probabilistic neural networks, Neural Networks, vol. 3, pp. 109-118.

[5] Streit R.L., Luginbuhl T.E., 1994, Maximum likelihood training of probabilistic neural networks, IEEE Trans. Neural Networks 5(5), pp. 764-783.

(1) Universitatea de Medicina si Farmacie din Craiova, str. Petru Rares, Nr. 2-4
E-mail address: fgorun@rdslink.ro

(2) Universitatea din Craiova, Str. A.I. Cuza, Nr. 13
E-mail address: mgorun@inf.ucv.ro

(3) University of Westminster, Harrow School of Computer Science, London, UK
E-mail address: revettk@westminster.ac.uk

(4) Universitatea de Medicina si Farmacie din Craiova, str. Petru Rares, Nr. 2-4
E-mail address: enem@umfcv.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 114–121

A COMPARISON OF QUALITY CRITERIA FOR UNSUPERVISED CLUSTERING OF DOCUMENTS BASED ON DIFFERENTIAL EVOLUTION

D. ZAHARIE[(1)], F. ZAMFIRACHE[(2)], V. NEGRU[(3)], D. POP[(4)], AND H. POPA[(5)]

Abstract. This paper presents an analysis of different quality criteria for unsupervised clustering of documents. The comparative analysis is based on a differential evolution algorithm which allows the estimation both of the number of clusters and of their representatives. The proposed approach is tested on a classical data set in document clustering. The results illustrate the particularities of different clustering criteria and the ability of the proposed approach to identify both the number of clusters and their representatives.

1. Introduction

Clustering is one of the first steps in organizing large sets of electronic documents and it plays an important role in topic extraction and in guiding the documents browsing. In a general sense, clustering means identifying natural groups in data such that data in a group, called cluster, are sufficiently similar while data belonging to different groups are sufficiently dissimilar. Clustering is, usually, an unsupervised process based only on the data to be analyzed. However, most partitional algorithms (e.g. k-Means) need the knowledge of the expected number of clusters. When even this number is unknown the process is called unsupervised clustering.

The main issues in designing a document clustering system are: (i) finding an adequate encoding of the documents; (ii) finding appropriate similarity measures between documents and appropriate quality measures of the clustering result; (iii) choosing a clustering technique.

The ideal encoding of documents is highly task-dependent since certain features should be taken into account when dealing with documents containing only text and other kind of features when processing web-documents [8]. In the case of text clustering a common representation is that based on the vector-space model

2000 *Mathematics Subject Classification.* 62H30, 68T20.
Key words and phrases. document clustering, clustering criteria, differential evolution.

©2007 Babeş-Bolyai University, Cluj-Napoca

using the term weighting based on the term frequency combined with the inverse document frequency.

Because of the unsupervised character, evaluating the quality of a clustering result is a difficult task. Unfortunately there does not exist one unique quality criterion but at least two of them should be combined. A thorough analysis of the influence of different quality criteria and of their combinations on document clustering is presented in [9]. However this analysis is based on the hypothesis that the number of clusters is known and the quality criteria are used only to compare partitions containing the same number of clusters. One of the aims of this work is to analyze the effectiveness of some combinations of quality criteria when they are used in fully unsupervised document clustering.

From the traditional clustering techniques the partitional ones are, by far, the most used in document clustering. This is due not only to the fact that the partitional techniques are less computational expensive than the hierarchical ones but also to the fact that, as reported in [10], they lead to comparable or even better clustering performance. Partitional approaches can be interpreted as optimization problems having as aim to maximize the quality criteria. The involved optimization problem is a complex one, making simple searching strategies to be easily trapped in local minima. In the last decade a lot of evolutionary and other nature-inspired approaches in clustering have been proposed [4, 5]. Recently some evolutionary related approaches have been applied also to document clustering. In [3] is presented a Particle Swarm Optimization approach while in [1] is presented a document clustering method based on the Differential Evolution (DE) algorithm. Both approaches proved to behave better than k-Means but they are based on the knowledge of the number of clusters. In this work we propose an extension of the approach in [1] characterized by the fact that both the number of clusters and their representatives are evolved.

2. Preprocessing and representation of documents

The set of documents to be clustered should be first preprocessed in order to find an appropriate representation of each document. If the documents are interpreted as plain text they could be considered bag of words. Since not all words appearing in a document are relevant, a first step would be to just eliminate the words which are very common in the language. This is usually done by using a so-called stop list specific to the document language. There currently exist stop lists for different languages. In our experiments we used such a classical stop list for English.

Another common processing is that of stemming, i.e. reducing derived words to their root form. One of the most used algorithms, which we also used in our work, is that of Porter [6]. Even if stemming is a frequently used procedure it is not always beneficial for the clustering process, as is illustrated in [7].

After these steps, each document will be a multi-set of terms (stemmed words) which can have a large number of elements. In order to reduce the dimensionality

corresponding to a document the terms having a low-frequency over the entire set of documents could be eliminated. In our approach we avoided to apply this in order to not eliminate terms which could play an important role in discriminating the clusters. In order to obtain a numerical finger-print of each document a score is assigned to each term belonging to the document. One of the most used approaches is that of using the term frequency in the document and the frequency of documents containing that term. Let us consider a set of N documents, D_1, D_2, ..., D_N and let t be a term. The weight of term t with respect to document D_i is defined as follows:

$$w(t, D_i) = f(t, D_i) \log(N/F(t)) \tag{1}$$

where $f(t, D_i)$ denotes the relative frequency of term t in the document D_i and $F(t)$ denotes the number of documents which contain the term t. It is easy to see that $w(t, D_i) \in [0, \log(N)]$. The minimal value corresponds to terms belonging to all documents and large values are obtained for terms which appear very frequently but only in one document. In order to limit the size of a document description, only the weights corresponding to terms in the document were stored. Each document can be thus described by the set of weights corresponding to the terms it contains: $\{w(t, D_i); t \in D_i\}$. Based on this representation the classical cosine similarity measure between two documents, D_i and D_j, can be defined as follows:

$$s(D_i, D_j) = \frac{\sum_{t \in D_i \wedge t \in D_j} w(t, D_i) w(t, D_j)}{\|D_i\| \|D_j\|} \tag{2}$$

where $\|D\| = \sqrt{\sum_{t \in D} w(t, D)^2}$ is in fact the Euclidean norm of the vector of weights corresponding to terms in D. Even if in the method implementation each document is described by the weights of terms belonging to the document in the following we shall formally consider that each document corresponds to a vector having the length equal with the number of terms in the entire set of documents. In this way all operations on vectors (summation and multiplication) are also valid on documents.

3. Choosing quality criteria for unsupervised clustering

The aim of partitional clustering is to find a partition $(C_1, \ldots, C_k)$ of the set $D = \{D_1, \ldots, D_N\}$ of documents such that $D = \cup_{i=1}^{k} C_i$ and $C_i \cap C_j = \emptyset$ for all $i \neq j$. A cluster C_r can be described by the set of all documents belonging to it or by a representative R_r which could be an element of D or another element from the vector space corresponding to the documents encoding ($[0, \log(N)]^\tau$, τ being the number of all terms in the set of documents). Based on their representatives, the clusters can be constructed by assigning each document to the nearest representative. A particular case of representatives is represented by the clusters centers, $R_r = (\sum_{D \in C_r} D)/n_r$, n_r being the number of elements of C_r.

The aim of the clustering process is to find that partition which maximizes the similarity between the elements of the same cluster and minimizes the similarity between elements belonging to different clusters. Thus partitional clustering can be formulated as an optimization problem involving one or multiple optimization criteria. Typical quality criteria of a partition are compactness, connectedness and separability.

Compactness is a measure of the concentration of data inside a cluster. It should be maximized and can be expressed as either the averaged similarity between all pairs of elements in the cluster or the total similarity between the elements in the cluster and its center. The most used is the second variant, characterized through

$$\mu_1(C_1,\ldots,C_k)=\sum_{r=1}^{k}\sum_{D\in C_r} s(D,R_r)=\sum_{r=1}^{k}\|S_r\| \tag{3}$$

where S_r is the sum of all documents in C_r and $R_r = S_r/n_r$ is the center of C_r. When the cluster representatives are not necessarily their centers then the last equality is no necessarily true.

Connectedness evaluates the degree to which similar documents (neighboring data) have been placed in the same cluster. The corresponding measure is [4]:

$$\mu_2(C_1,\ldots,C_k)=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{L}\sum_{j=1}^{L}\gamma(D_i,D_{\nu(i,j)}) \tag{4}$$

where $D_{\nu(i,j)}$ denotes the j-th nearest neighbor of document D_i,

$$\gamma(D_i,D_{\nu(i,j)})=\begin{cases} 1/j & \text{if } D_i \text{ and } D_{\nu(i,j)} \text{ are placed in the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and L is the number of considered nearest neighbors. While compactness favors spherical clusters, connectedness allows the generation of arbitrary shaped clusters.

Separability measures how the various clusters are different from each other. Unlike the previous measures this one should be minimized. A measure, corresponding to the case when the representatives are the clusters centers is:

$$\mu_3(C_1,\ldots,C_k)=\sum_{r=1}^{k} n_r s(R_r,R)=\sum_{r=1}^{k} n_r s(S_r,S) \tag{6}$$

where S is the sum of all documents in the set and $R = S/N$. In [1] is used a different separability measure, based on the maximal similarity between the clusters representatives (which usually are not the clusters centers but vectors generated by the evolutionary algorithm):

$$\mu_4(C_1,\ldots,C_k)=\sum_{r=1}^{k}\max_{q=\overline{1,k},q\neq r} s(R_q,R_r) \tag{7}$$

In order to obtain a quality clustering criteria, the above measures should be combined. Different combinations were proposed in the literature. For instance in [9], besides other measures which are not presented here, was analyzed μ_1/μ_3, which proved to be the best. In [1] was used $1/(\mu_4/\mu_1 + \epsilon)$ with $\epsilon > 0$ a small correction value.

When the number of clusters is not predefined and the quality criteria are used to compare partitions having different numbers of clusters we have to take into account the natural bias of different measures to favor a small or a large number of clusters. Let us suppose that all document vectors are normalized ($\|D_i\| = 1, i = \overline{1,N}$). If the representatives are the centers of the clusters then the compactness measure μ_1 satisfies

$$\mu_1(C_1, \ldots, C_k) = \sum_{r=1}^{k} \| \sum_{i=1}^{n_r} D_{l(i)} \| \leq \sum_{r=1}^{k} \sum_{i=1}^{n_r} \|D_{l(i)}\| = N \tag{8}$$

where $l(r,i)$ denotes the index of the ith document in the cluster C_r. Since the trivial clustering which correspond to the case when each document is in its own cluster is characterized by a value of μ_1 equal to N it follows that by using only the μ_1 criterion and letting k to vary, the maximum will be attained for the maximal value of k. A similar behavior was remarked in the case when the representatives are not necessarily the centers but in this case this fact cannot be theoretically proven so easy.

On the other hand, it is easy to see that the maximal value of connectedness is obtained if all documents are assigned to one cluster, thus when trying to maximize μ_2 a small number of clusters is favored. In the case of the separability measure μ_3 (which should be minimized) the following relations hold:

$$\mu_3(C_1, \ldots, C_k) = \sum_{r=1}^{k} n_r \frac{S_r^T \cdot S}{\|S_r\|\|S\|} \geq \frac{1}{\|S\|} \sum_{r=1}^{k} n_r \frac{S_r^T \cdot S}{n_r} = \frac{1}{\|S\|} \sum_{i=1}^{N} D_i^T \sum_{j=1}^{N} D_j \tag{9}$$

since $\|S_r\| \leq \sum_{i=1}^{n_r} \|D_{l(r,i)}\| = n_r$. The last term in eq. 9 is the value of μ_3 corresponding to the case of N clusters, thus it follows that by minimizing μ_3 one maximizes k. On the other hand, by minimizing the separability measure μ_4, the partitions having a small number of clusters are favored.

If the number of clusters should be estimated, the optimization criterion should involve two measures which are characterized through opposed dependence on the number of clusters (otherwise trivial partitions having either 1 cluster or N clusters are obtained). In Table 1 are summarized all possible combinations of the above four measures by marking which ones favor the increase of the number of clusters and which favor their decrease. Combinations where both measures favor the same modification on the number of clusters are not appropriate when this number should be estimated. Thus the criteria μ_1/μ_3 which proved to have a good behavior in the case of a fixed number of clusters is no more appropriate in the case

of a variable number, since by favoring high values of k it leads to an overestimation of the number of clusters. Combining μ_2 with μ_4 one obtains a criterion which favor the small number of clusters and could lead to an underestimation of k. On the other hand, the criteria $\mu_1\mu_2$ used in [4], $1/(\mu_4/\mu_1+\epsilon)$ used in [1] and those obtained by combining μ_2 with μ_3 (μ_2/μ_3) or μ_3 with μ_4 $(1/(\mu_3\mu_4))$ ensures the compromise between small and large values of k.

(μ_1,μ_2) [4]	(μ_1,μ_3) [9]	(μ_1,μ_4) [1]	(μ_2,μ_3)	(μ_2,μ_4)	(μ_3,μ_4)
$(\uparrow,\downarrow)$	$(\uparrow,\uparrow)$	$(\uparrow,\downarrow)$	$(\downarrow,\uparrow)$	$(\downarrow,\downarrow)$	$(\uparrow,\downarrow)$

TABLE 1. Influence of different combinations of criteria on the evolution of the number of clusters when the combined clustering criterion is maximized ($\uparrow$ - favor the increase, $\downarrow$ - favor the decrease)

4. Application of differential evolution for unsupervised document clustering

In [1] is illustrated the fact that Differential Evolution (DE) provides better results than classical Genetic Algorithms and Particle Swarm Optimization when applied to document clustering. This is why we chose to extend this approach in order to deal with the case of an unknown number of clusters. DE is a simple evolutionary approach based on a particular type of recombination which involves three randomly selected parents in order to obtain an offspring. The differences between our approach and that in [1] are related with the population elements encoding and with the recombination operator.

In the approach we developed (called extended DE - eDE) each element of a population corresponds to a partition and has the following components: $(k, R_1, \ldots, R_k)$ where $k \in \{k_{min}, \ldots, k_{max}\}$ is the number of clusters and R_r, $r = \overline{1,k}$ are representatives of the clusters (vectors of weights associated to terms in the documents set). At the start of the evolutionary process, the population elements are randomly initialized: k is randomly selected from its range, $\{k_{min}, \ldots, k_{max}\}$, and for each cluster the representative is randomly selected from the entire set of documents, D. At each iteration of the evolutionary process for each element e_i $(i = \overline{1,m})$ of the population the following operations are executed:

(i)Three other distinct elements e_{j_1}, e_{j_2} and e_{j_3} are randomly selected from the population.

(ii)A new trial element $e' = (k', R'_1, \ldots, R'_{k'})$ is constructed as follows: $k' = \lfloor |k^{(j_1)} + F \cdot (k^{(j_2)} - k^{(j_3)})| \rfloor$ with a probability p and remains $k^{(i)}$ with the probability $1-p$. For each r the representative R'_r is constructed as a linear combination of randomly selected representatives from those three elements: $R'_r = R_{r_1}^{(j_1)} + F \cdot (R_{r_2}^{(j_2)} - R_{r_2}^{(j_2)})$ with probability p and remains R_r with probability $1-p$.

(iii)The trial element is evaluated with respect to the chosen optimization criteria and if it is better than the original element, e_i, then it replaces e_i.

This iterative process continues until a given number of generations is reached. The parameters $p \in (0, 1]$ and $F \in (0, 2)$ influences the convergence properties of the DE algorithm but in this study we did not give a particular attention to these. They were fixed on $p = 0.5$ and $F = 0.75$ (the average of the values used in [1]). Based on the results reported in [1] we also hybridized the DE approach with k-Means: after a given number of DE generations (e.g. 50), k-Means is applied to all elements of the population.

Algorithm	Error ± stdev	Entropy ± stdev	No.clusters ± stdev	Success ratio
k-Means	0.25797± 0.00753	0.52077± 0.02606	3	10/10
eDE + $\mu_1\mu_2$	0.24129± 0.08577	0.32948± 0.15888	6.8± 1.469	0/10
eDE +μ_1/μ_3	0.29683± 0.01234	0.37365± 0.05982	10	0/10
eDE+μ_1/μ_4	0.29958± 0.072117	0.48675± 0.13626	5.8± 2.749	1/10
eDE+μ_2/μ_3	**0.18442± 0.04454**	**0.32838± 0.05580**	**4± 1.549**	**6/10**
eDE+μ_2/μ_4	0.37988± 0.10048	0.69751± 0.15007	2.6± 0.66332	4/10
eDE+$1/(\mu_3\mu_4)$	0.37569± 0.06833	0.74449± 0.13602	5.2± 2.0396	3/10

TABLE 2. Clustering results for a small set of documents (210 documents belonging to 3 classes and containing 14284 terms)

5. Results and further work

The experimental analysis is based on a classical dataset consisting of 3891 documents representing abstracts corresponding to three categories: CISI, CRANFIELD and MEDLINE (ftp://ftp.cs.cornell.edu/pub/smart). The total number of terms remained after preprocessing is 283720. In a direct vector space encoding this would mean to work with vectors having 283720 components. In order to analyze different clustering criteria we randomly selected a subset of the dataset consisting of 210 documents. In order to evaluate the quality of the obtained partition we used two measures based on the knowledge of the real assignment of data to classes: error ratio (ratio of documents pairs which either belong to the same class and have been assigned to different clusters or they belong to different classes and were assigned to the same cluster) and the classical entropy measure [9]. The results obtained for this set (for a population of 15 elements and 50 generations, for a number of clusters limited to $\{2, \ldots, 10\}$ and for 10 independent

runs) are presented in Table 2. These results confirm the remarks presented in the previous section and suggest that the best behavior is obtained by combining the connectedness and separability measures. In the case of the set of all 3891 documents the results obtained by the DE-based approach using the criterion μ_2/μ_3 are 0.01366±0.00109 (error ratio) and 0.05347±0.003511 (entropy) while k-Means led to 0.03753±0.00499 (error ratio) and 0.09374±0.01347 (entropy).

An extended experimental analysis based on other documents collections, including web documents will be further conducted. Another aspect to be analyzed is that of reducing the number of terms considered into the clustering process. The present analysis was intentionally based on the entire set of terms in order to have a reference result for future variants based on reduced feature vectors.

Acknowledgement. This work was supported by the project MindSoft (RO-CNCSIS 1385/05).

References

[1] A. Abraham, S. Das, A. Konar; Document Clustering Using Differential Evolution, Proceedings of CEC 2006 - IEEE Congress on Evolutionary Computation, pp. 1784- 1791, 2006.

[2] A. Casillas, M. T. Gonzalez de Lena, R. Martynez; Document Clustering into an unknown number of clusters using a Genetic Algorithm, Proc. of 6th International Conference on Text, Speech and Dialogue, LNCS 2807, pp.43-49, 2003.

[3] X. Cui, T. E. Potok, P. Palathingal, Document Clustering using Particle Swarm Optimization, Proc. of the 2005 IEEE Swarm Intelligence Symposium, June, 2005, Pasadena, California, USA, pp. 185-191, 2005.

[4] J. Handl, J. Knowles; Multiobjective clustering with automatic determination of the number of clusters. Technical Report TR-COMPSYSBIO-2004-02. UMIST, Manchester, 2004.

[5] T. Krink, S. Paterlini; Differential Evolution and Particle Swarm Optimization in Partitional Clustering, Computational Statistics and Data Analysis, Volume 50, Issue 5, pp. 1220-1247, 2006.

[6] M.F. Porter; An Algorithm for Suffix Stripping, Program, 14(3), pp. 130-137, 1980.

[7] M.P. Sinka, D.W.Corne; A Large Benchmark Dataset for Web Document Clustering, in Abraham, A., Ruiz-del-Solar, J., Koeppen, M. (eds.), Soft Computing Systems: Design, Management and Applications, Volume 87 of Frontiers in Artificial Intelligence and Applications, pp. 881-890, 2002.

[8] M.P. Sinka, D.W.Corne; Evolving Document Features for Web Document Clustering: A Feasability Study, Proceedings of CEC 2004 IEEE Congress of Evolutionary Computation, Portland, USA, 2004.

[9] Y. Zhao, G. Karypis; Criterion Functions for Document Clustering. Experiments and Analysis,Technical Report of Army HPC Research Center, Minneapolis, #01-40, 2002, (available online: http://citeseer.ist.psu.edu/zhao02criterion.html)

[10] Y. Zhao, G. Karypis; Hierarchical clustering algorithms for document datasets, Data Mining and Knowledge Discovery, Volume 10, Number 2, pp. 141-168, 2005.

(1,2,3,4,5) Department of Computer Science, West University of Timişoara, bv. V. Pârvan, no. 4, 300223, Timişoara

E-mail address: (1)dzaharie@info.uvt.ro, (2)zflavia@info.uvt.ro, (3)vnegru@info.uvt.ro, (4)danielpop@info.uvt.ro,(5)hpopa@info.uvt.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 122–129

SIMULATING MICROCAPILLARY NETWORKS USING RANDOM GRAPHS

SERBAN RARES POP [(1)], CIPRIAN-IONUT DUDUIALA [(2)], AND CAMELIA CHIRA [(3)]

Abstract. The blood contains plasma, a Newtonian fluid, and other suspended elements like red blood cells (erythrocytes), white blood cells (leukocytes) and platelets. These components, particularly red blood cells, strongly influence the blood properties and behaviour. A mathematical model is proposed to solve problems like the flow in a microcapillary network that includes the blood rheology and non-linear cell splitting at bifurcations. This model can be used to perform statistical studies on real microcapillary networks. The introduced model facilitates the characterization and prediction of events and behaviours unreachable with standard tools.

1. Introduction

Due to their different structures, microcapillaries and veins have different biological properties. In the case of a vein being destroyed, it can be replaced with an artificial one. However, this procedure can not be applied for a microcapillary. For this reason a representation of microcapillary networks that allows predictions and statistical studies is needed. A method based on random graphs for generating microcapillary networks is proposed. The resulting network is compared with a real microcapillary network. These results can be used to analyze microcapillary networks around the brain or around tumors. Furthermore, prediction of events and behaviours is enabled. For example, the proposed method can predict which network is vulnerable emphasizing that damaging the network or portions of it can be fatal.

2. Mathematical Model of a Microcapillary Network

Let us denote by $\mathcal{V}$, $\mathcal{I}$, $\mathcal{O}$, and $\mathcal{N}$ the sets of microvessels, inlet, outlet and interior nodes. In order to construct a microcapillary network, information about each vessel of uniform radius R_j and length L_j, with $j \in \mathcal{V}$ is needed. Also, the inlet and outlet pressures should be known since the flow is driven by the overall pressure drop (i.e. the difference between the inlet and outlet pressures). Using

Key words and phrases. microcapillary networks, random graph.

©2007 Babeş-Bolyai University, Cluj-Napoca

network parameters and the Network Solver Algorithm (presented in Section 2.4), the hematocrit in each link, pressures and flow rates distribution in the network can be determined. This actually refers to the way in which the blood moves into the network from the input nodes to the output nodes. An oriented weighted graph can be used to capture structural information about such a network.

2.1. **Hematocrit-dependent viscosity.** The blood viscosity reflects the property of the blood/vessel system for given flow conditions rather than solely the property of the blood itself. This viscosity is called the "apparent" or "effective" viscosity and depends strongly on the hematocrit and diameter of the vessel. This dependence is known as the Fåhræus-Lindqvist effect (see for instance [1]).The proposed model relies on the in vivo viscosity law, provided by Pries [1], which is derived from direct viscosity measurements:

$$\mu_j(H_j(x,t)) = \mu^p \cdot \mu_j^{rel},\ j \in \mathcal{V}, \tag{1}$$

where

$$\mu_j^{rel} = \left[1 + \left(\mu_j^* - 1\right) \frac{\left(1 - H_j(x,t)\right)^{C_j} - 1}{(1-0.45)^{C_j} - 1} \left(\frac{2\bar{R}_j}{2\bar{R}_j - 1.1}\right)^2\right] \left(\frac{2\bar{R}_j}{2\bar{R}_j - 1.1}\right)^2, \tag{2}$$

$$\mu_j^* = 6\exp\left(-0.17\bar{R}_j\right) + 3.2 - 2.44\exp\left(-0.06(2\bar{R}_j)^{0.645}\right), \tag{3}$$

and

$$C_j = \left(0.8 + \exp(-0.15\bar{R}_j)\right)\left(-1 + \frac{1}{1 + 10^{-11}(2\bar{R}_j)^{12}}\right) + \frac{1}{1 + 10^{-11}(2\bar{R}_j)^{12}}. \tag{4}$$

In the above equations, μ_j is the effective blood viscosity, μ^p is the plasma viscosity (4×10^{-3} Pa $\cdot$ s), and $C_j, \mu_j^*, \mu_j^{rel}$ are fitting coefficients which depend on the vessel radius and hematocrit - the proportion of blood occupied by red blood cells is referred to as the hematocrit.

2.2. **Poiseuille law.** The blood flow rate Q_j is given by the Poiseuille law (see [1]), under the assumption that the vessels are long and thin (lubrication theory). Hence, the blood flow rate is related at the pressure drop ΔP_j along the jth vessel by

$$Q_j = \sigma_j \Delta P_j, \tag{5}$$

where

$$\sigma_j = \frac{\pi R_j^4}{8 L_j \mu_j(H_j, R_j)}, \tag{6}$$

and L_j is the length of the jth vessel.

2.3. **Boundary conditions at bifurcations.** An important characteristic of the microcirculation is the non-uniform distribution of the blood components between the outgoing vessels at the splitting nodes. In particular, the higher the fraction of blood an outgoing vessel receives, the higher the hematocrit in that vessel. Hence, it is possible for one branch to receive a higher hematocrit than that of the parent vessel and the other to receive a lower (possibly zero) hematocrit. This phenomenon is called "plasma skimming" or "phase separation" [1].

In what follows, the parametric description of phase separation *in vivo*, proposed by Pries *et al* [1] is used. This model was obtained by fitting *in vivo* experimental data.

For any splitting node $k,\ k \in \mathcal{N}$, connecting the incoming vessel F_k and outgoing vessels α_k and β_k, with $F_k,\ \alpha_k,\ \beta_k \in \mathcal{V}$, mass conservation implies

$$Q_{F_k} = Q_{\alpha_k} + Q_{\beta_k}. \tag{7}$$

The fractional hematocrits of the outgoing vessels are

$$\frac{H_{\alpha_k}}{H_{F_k}} = \frac{1}{Q_k} \begin{cases} F_\alpha(Q_k), & X_{0_k} < Q_k < 1 + X_{0_k} \\ 0, & Q_k \leq X_{0_k} \\ 1, & Q_k \geq 1 + X_{0_k} \end{cases} \tag{8}$$

with $Q_k = Q_{\alpha_k}/Q_{F_k}$. The functions $F_\alpha(Q_k)$ and $F_\beta(Q_k)$ are given by

$$F_\alpha(Q_k) = \frac{1}{1 + \exp\left[-A_{\alpha_k} - B_k \log\left(G(Q_k)\right)\right]}, \tag{9}$$

$$F_\beta(Q_k) = \frac{1}{1 + \exp\left[-A_{\beta_k} - B_k \log\left(G(Q_k)\right)\right]}, \tag{10}$$

where

$$G(Q_k) = \frac{Q_k - X_{0_k}}{1 - Q_k - X_{0_k}}. \tag{11}$$

Likewise H_{β_k}/H_{F_k} satisfies (8) with Q_k replaced by $1 - Q_k$ and F_α by F_β.

The parameters A_{α_k}, A_{β_k}, B_k and X_{0_k} define the main aspects of the phase separation (i.e. asymmetry, sigmoidal shape and threshold):

$$A_{\alpha_k} = -\frac{6.96}{2\bar{R}_{F_k}} \ln\left(\frac{\bar{R}_{\alpha_k}}{\bar{R}_{\beta_k}}\right),\ A_{\beta_k} = -\frac{6.96}{2\bar{R}_{F_k}} \ln\left(\frac{\bar{R}_{\beta_k}}{\bar{R}_{\alpha_k}}\right), \tag{12}$$

$$B_k = 1 + 6.98\left(\frac{1 - H_{F_k}}{2\bar{R}_{F_k}}\right),\ X_{0_k} = \frac{0.2}{\bar{R}_{F_k}}. \tag{13}$$

The Pries-Secomb phase separation rule ensures conservation of hematocrit

$$H_{\alpha_k} Q_{\alpha_k} + H_{\beta_k} Q_{\beta_k} = H_{F_k} Q_{F_k}. \tag{14}$$

At conjunctions (two vessels α_k and β_k meet to form a single output vessel G_k) we need to apply only

$$Q_{\alpha_k} + Q_{\beta_k} = Q_{G_k},\ H_{\alpha_k} Q_{\alpha_k} + H_{\beta_k} Q_{\beta_k} = H_{G_k} Q_{G_k}. \tag{15}$$

2.4. **Network Solver Algorithm.** Proposed model facilitates computation of the hematocrit in each link as well as pressures and flow rates distribution in the network. The computational steps can be described by a procedure called Network Solver Algorithm (NSA). NSA is outlined below.

Network Solver Algorithm

Step 1: For given initial conditions (value of inlet/outlet pressure or flow rate), network geometry (length and radius for each vessel), and assuming an initial uniform hematocrit distribution for each vessel, $H_j,\ j \in \mathcal{V}$, compute viscosity for each link, pressures and flow rates distribution in the entire network.

Step 2: Apply the splitting rule to compute new values for the hematocrit in each link, H_j^{new} and repeat *Step 1* until the absolute error $\epsilon = \max(|H_j - H_j^{new}|),\ j \in \mathcal{V}$ is smaller than a given tolerance.

3. Random Graphs

A random graph [3] is a collection of vertices and edges randomly connecting pairs of nodes. Ussually, it is assumed that the presence or absence of an edge between two vertices is independent of the presence or absence of any other edge, so that each edge may be considered to be present with independent probability p. If the graph has N vertices each of them connected to an average of z edges, then it is trivial to show that $p = z/(N-1)$, which for large N is usually approximated by z/N. The number k of edges connected to any particular vertex is called the degree of that vertex. Ordinary random graphs are characterized by the Poisson distribution of vertex degree:

$$p_k = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{z^k e^{-z}}{k!}, \tag{16}$$

where p_k is the probability that a randomly chosen vertex on the graph has degree k.

Random graphs serve as models of real-world networks of different types, especially in epidemiology. For example, a disease passing through a community is strongly dependent on the contacts between those infected and those susceptible to disease. The network will have individuals represented by vertices and contacts by edges. Another widely studied network is the Internet. However, random graphs turn out not to be able to simulate with accuracy real-world phenomens.

3.1. **Generating functions.** Generating function [2] is a standard tool that can be used to create microcapillary networks. An example of such a function is $G_0(x)$

for the probability distribution of vertex degrees k. For a unipartite undirected graph of N vertices, with N large, $G_0(x)$ can be expressed as:

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k, \tag{17}$$

where p_k is the probability distribution of k. Usually the generating functions verify the condition $G_0(1) = 1$, since the distribution p_k is assumed correctly normalized.

Some properties of probability generating functions are listed below:

- probability p_k is given by the kth derivative of G_0: $p_k = \frac{1}{k!}\frac{d^k G_0}{dx^k}(0)$,
- the average degree z of a vertex is $z =< k >= \sum_k k p_k = G_0^{'}(1)$, which means that computing the mean of the probability distribution that the function generates, as well as higher moments of the distribution using the corresponding higher order derivatives of G_0.
- if a generating function is used for the distribution of a property k of an object, then the distribution of the total of k summed over m independent realizations of the object is generated by the m^{th} power of that generating function [2].

Depending on the distribution used, several other function can be defined: for Poisson-distributed graphs the probability $p = z/N$ of the existence of an edge between any two vertices is the same for all vertices and

$$G_0(x) = \sum_{k=0}^{\infty} \binom{N}{k} p^k (1-p)^(N-k) x^k = (1-p+px)^N, \tag{18}$$

while for exponentially distributed graphs we have $p_k = (1-e^{-1/K})e^{-k/K}$, with K constant, and

$$G_0(x) = (1-e^{-1/K}) \sum_{k=0}^{\infty} e^{-k/K} x^k = \frac{1-e^{-1/K}}{(1-xe^{-1/K})}. \tag{19}$$

Furthermore, the expression of the function that generates the distribution of outgoing edges can be determined

$$G_1(x) = \frac{G_0^{'}(x)}{G_0^{'}(1)} = \frac{1}{z} G_0^{'}(x), \tag{20}$$

with z the average vertex degree. Using the third property of the generating function described above, the generating function for the probability distribution concerning the number of second neighbors for a vertex is obtained:

$$\sum_{k=0}^{\infty} p_k G_1^k(x) = G_0(G_1(x)). \tag{21}$$

The distribution of third-nearest neighbors is generated by $G_0(G_1(G_1(x)))$, and so on. Since $G_1(1) = 1$, the average number of second neighbors is $z_2 = G_0^{'}(G_1(x))_{x=1} = ... = G_0^{'}(1)G_1^{'}(1) = G_0^{''}(1)$. Taking into acount that we also have $z = G_0^{'}(1)$, one should not think that $z_k = G_0^{(k)}(1)$ because in general this is not true.

All the properties defined above can be used for component sizes, mean component size, phase transition, giant component, numbers of neighbors and average path length analysis [2].

4. Microcapillary Networks Generation Using Random Graphs

An algorithm useful for simulating a microcapillary network using random graphs is presentes. The obtained oriented weighted graph contains all the information needed to analyze the network. It is not possible to use directed random graphs to generate the microcapillary network for two main reasons as follows: (i) the blood direction through a vessel is given by the inlet and outlet pressures, and (ii) the blood direction may change in time (this means that even if at each moment the microcapillary network can be seen as a directed graphs, in reality the graph is in reality the graph is undirected).

When random graphs with various distributions of vertex degree are generated, a set of N random numbers k_i to represent the degrees of the N vertices in the graph is needed. Next, pairs of vertices are randomly chosen and joining edges are placed on the graph . This way a graph is generated (with equal probability) from the set of all possible graphs with the given set of vertex degrees. The only condition that has to be checked is that the sum $\sum_i k_i$ of the degrees is even, since each edge added to the graph has two ends. If the condition is not satisfied, a new set k_i is generated and the procedure is repeated until a suitable set is obtained. Integers representing vertex degrees with any desired probability distribution can be generated using the transformation method or a rejection or hybrid method [2].

Using the above algorithm, the edges of the graph are constructed. In case of microcapillary networks, things are a lot less complicated. As mentioned in Section 2, the network will have a set of in-nodes $\mathcal{I}$, a set of out-nodes $\mathcal{O}$ and a set of interior nodes $\mathcal{N}$. The in-nodes and out-nodes will have degree 1, while the interior nodes, due to the biological properties of microcapillaries, will have degree 3. In other words the set k_i may contain only elements with values 1 or 3, thus the condition that should be satisfied is $|\mathcal{I}| + |\mathcal{O}| \equiv |\mathcal{N}|(mod\ 2)$. This condition means that the number of in-nodes and out-nodes must have the same parity as the number of interior nodes. The generating function for the probability distribution becomes

$$G_0(x) = \frac{|\mathcal{I}| + |\mathcal{O}|}{TOT}x + \frac{|\mathcal{N}|}{TOT}x^3, \tag{22}$$

where $TOT = |\mathcal{I}| + |\mathcal{O}| + |\mathcal{N}|$.

The key of the paper is the fact that the obtained graph contains several connected componets. In reality, we need a graph which is connected, which means that the giant component [2] contains all the nodes of the graph. This is possible only if $|\mathcal{I}| + |\mathcal{O}| \leq |\mathcal{N}|$. The condition is implied by the network structure: an in-node or out-node can be linked only with interior nodes. Hence, it is easier to construct the graph by determining first the edges connecting only interior nodes and only at the end the edges involving in-nodes and out-nodes.

There are two ways of constructing a connected graph, using the random graphs theory: (i) construct the graph as described above and then add as few edges as possible between the different connected components or (ii) construct $\mathcal{N}/2$ edges that form $\mathcal{N}/2$ different connected components (i.e. each interior node is used exactly once), add at each step only edges that connect nodes from different components to reduce their number and when only one component is obtained, pairs of vertices are randomly chosen and joining edges are also placed on the graph, taking into acount the last observation from the previous paragraph.

First method implies modifying the number of interior nodes. Adding an edge between two different connected components acctually means adding two new nodes in the graph and replacing one edges from each compenent with two new edges. Second methods keeps constant the number of nodes and generates - with equal probability - one of the connected graphs that respect the above properties.

On the other hand, a network containing several connected components can also be useful. In such a case, studies about how the flow changes in each component when the components are joined together by adding nodes and edges can be made.

The next step is to assign to each vessel a random radius R_j and a random length L_j, whose values are taken between a minimum and a maximum value obtained from experiments. Observations of microcapillary networks show that some parts of the network may contain larger vessels than others, thus it might be useful to use different minimum and maximum values for different components of the network.

Finally, the flow problem of the network should be solved. Since only one edge leaves the in-nodes only one edge enters the out-nodes enters the out-nodes, the blood direction for these vessels can be exactly determined. For the rest the Network Solver Algorithm (proposed in Section 2) is applied.

5. Conclusions and Future Work

Some properties of the microcapillary networks are presented. A mathematical model for a microcapillary network and an algorithm to solve the flow in such a network are proposed. Random graphs and generating functions represent useful tools for analyzing the networks simulating real-world phenomena. A way in which microcapillary networks can be simulated using undirected graphs (which are then transformed into weighted directed graphs by solving the flow in the network) is indicated.

Future work focuses on analyzing the differences between a simulated microcapillary network and a real one. Simulations can indicate the potential of the proposed model for microcapillary networks analysis. Genetic algorithms can be used to indicate the damaging rate that will not indluence the overall functionality of the network. Furthermore, the vital vessels of the network (i. e. destroying them could lead to a hemorrhage) can be identified using evolutionary techniques.

References

[1] A.R. Pries, T.W. Secomb, P. Gaehtgens - Biophysical aspects of blood flow in the microvasculature. *Cardiovascular Research* **32** (1996), 654-667.

[2] M. E. J. Newman, S. H. Strogatz, and D. J. Watts - Random graphs with arbitrary degree distributions and their applications, Physical Review E, Volume 64, 026118 (2001)

[3] P. Erdos and A. Renyi, On random graphs I, Publ. Math. (Debrecen) 9 (1959), 290-297.

(1) School of Mathematical Sciences, University of Nottingham, UK
E-mail address: popserban@yahoo.com

(2) Centre for Mathematical Medicine and Biology, School of Mathematical Sciences, University of Nottingham, UK
E-mail address: cipriduduiala@yahoo.com

(3) Mathematics and Computer Science Faculty, "Babes-Bolyai" University, Cluj-Napoca, Romania
E-mail address: cchira@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 130–137

MINING AN ANIMAL TOXIN DATABASE: CHARACTERIZING PROTEIN FOLDS

KENNETH REVETT[(1)], FLORIN GORUNESCU[(2)], AND MARINA GORUNESCU[(3)]

Abstract. The vast majority of animal toxins acts either on sodium or potassium channels. These channels regulate neuronal activity by either inhibiting or activating various neuronal systems. Toxins have been a useful tool for mapping the distribution of various channels types within a variety of organisms. The aim of this paper is to present an automated approach for detecting whether a toxin acts on voltage-sensitive sodium versus potassium channels. In addition, our consensus sequence is also able to reliably determine whether the toxin acts as a gate modifier or pore blocker ($> 93\%$ accuracy). Lastly, we present evidence regarding the existence of two new putative potassium channel gate binding motifs through the use of a learning vector quantization neural network.

1. Introduction

Evolution has endowed animal species with the ability to produce a bewildering array of toxic substances, for both protection and predation. The effects of toxins range from being mildly irritating to lethal on the inflicted victim. Toxins can be broadly classified as either venoms or poisons. Toxins interfere with the normal functioning of specific cell types within the prey organism, although the mechanism(s) of action of several toxins remain to be elucidated [6].

Considering the wide range of toxicities exhibited by animal toxins, it would be very helpful to be able to have a structure function relationship for a given toxin. This task has not yet been accomplished, as there appears to be a small number of structural features (i.e. protein folds) for the vast array of toxins that produce overlapping mechanisms of action (e.g. $\alpha\alpha$ or $\beta\alpha\beta\beta$, [7]). This lack of specificity has made the automated determination of site of action versus structure very difficult. What is still required is a clear consensus sequence that relates structure to site of action. We have attempted to address the issue of generating a consensus sequence, focusing solely on potassium and sodium channels [2]. The challenge has

2000 *Mathematics Subject Classification.* 68P05, 68T05.

Key words and phrases. animal toxins, consensus sequence, learning vector quantization, codebook vector.

©2007 Babeş-Bolyai University, Cluj-Napoca

been to find a direct association between the structure of the toxin and its specific site of action.

There are several internet-based databases that contain specific information about various toxins ([9], http://www.ncbi.nlm.nih.gov/, http://ca.expasy.org/, http://www.expasy.org/sprot/tox-prot etc.).

In this work, we sought to determine if we could discover one or more consensus sequences for sodium and potassium channels. In addition, we wished to determine if a consensus sequence could be obtained, providing a classification based on whether they acted on the pore or the gate. The results would allow one to determine directly from sequence databases whether the toxin acted on sodium/potassium channels and also whether they would inactivate the channel either by binding to the gate or by blocking the channel pore.

This paper is organized as follows: the next section presents a description of the basic methodology employed, followed by a presentation of the major results, and lastly by a brief conclusion section.

2. Methods

This study entailed the use of several internet based protein structure repositories. The basic outline of our consensus development strategy is as follows:

1. Obtaining the toxins targeting sodium and potassium channels.
2. Refining toxins by extracting the active forms.
3. Separate toxins based on their site of actions.
4. Using various consensus sequence extraction tools sat as PRATT.
5. Comparing the resultant consensus sequences thus obtained by doing a search through the PDB looking to see what the sensitivity and specificity of the resultant hit list was.
6. PRATT consensus builder was employed.
7. Database search using the generated consensus sequences.
8. Repeat from step 2 as necessary.

Below we describe the process in more detail.

2. 1. *Obtaining the toxins targeting Na and K Voltage gate ion channels*: The keywords 'potassium channel inhibitor' were entered into the SRS on the ExPASy server returning a list of peptide toxins targeting Kv channels, returning a list of 155 peptides. Sequences were in FASTA format and output saved into a text document named Master-K.txt. FASTA format was chosen as it is recognized by many types of bioinformatics analysis tools available online [7]. The process was repeated for the Sodium channel toxins using the keywords 'Sodium channel inhibitor', returning 283 toxins.

Refining toxins by extracting active forms: Toxins in ExPASy can contain entire active sequences but sometimes contain Signal peptides and/or Propeptide

sequences. These regions were removed in order to obtain active peptide sequences only, as programs may take those features to generate results. Under the region 'Features' is where details are presented if available, that describes details such as domains and disulphide bonds, e.g. Charybdotoxin b precursor from the scorpion *Leiurus quinquestriatus hebraeus*, SWISSPROT id P59943:

MKILSVLLLA LIICSIVGWS EAQFTDVSCT TSKECWSVCQ RLHNTSIGKC MNKKCRCYS

Key/From/To/Length/Description ⇒ [SIGNAL/1/22/22/"by similarity"]

MKILSVLLLA LIICSIVGWS EA

(leaving the active peptide):

QFTDVSCT TSKECWSVCQ RLHNTSIGKC MNKKCRCYS

These features in ExPASy are not all experimentally derived and have been assigned 3 types of comments [7]: (*a*) Potential, (*b*) Probable and (*c*) By similarity.

'Potential' -there is some logical evidence that given annotation could apply. This non-experimental qualifier is often used to present the results from protein sequence analysis tools if the results make sense in respect to a given protein [1], [2]. 'Probable' -is a stronger indicator than 'Potential' and is based on some experimental evidence, that the given information is expected to be found in the natural environment the protein [3], [4].'By similarity' -facts proven for the protein or part of it, and then transferred to other protein family members within a certain taxonomic range. Sites within conserved domains to each other include active sites and disulfide bonds [5]. The Master files contain active toxin peptides, were 'saved as' Primary files: Prim-Na.txt and Prim-K.txt files. Master files remained untouched since downloading.

Separating toxins by site of action: The toxins in the Primary files were separated by site of action, sites where determined through literature reading from associated links of ExPASy. The resulting data was stored on disc for further analysis (see below).

Prim-K.txt: K-pores.txt & K-gaters.txt

MasterNa.txt: Napores.txt & Na-gaters.txt

K-files: SWISSPROT ID's for each toxin in the Prim-K.txt file was entered into ExPASy. The resulting page has links to related (published) literature that was reviewed to determine the toxin as a pore blocker or a gate modifier. Once determined, the toxin entry was cut and pasted into the respective file.

Na files: The process for Kv toxins was repeated, however due to the structural differences between the channels, Nav was found to have 6 binding sites, 5 of which are associated with the gate (sites 2-6) and site 1 the pore.

Veratridine, Batrachotoxin and Grayanotoxin are toxins acting on site 2, but are not peptides. They were not present on the list of the ExPASy output for Sodium channel inhibitors. Brevetoxin and Ciguatoxin acting on site 5 again are

not peptide toxins that were returned on the output for Sodium channel inhibitors, therefore these toxins will be excluded.

PRATT Consensus builder: PRATT (http://www.ebi.ac.uk/pratt/) is part of EBI (http://www.ebi.ac.uk) and generates consensus sequence motifs from unaligned fasta files. ExPASy has many other databases and tools for analysis of proteins and PROSITE will be initially used to search against the generated consensus. PRATT search input parameters can be modified to the user preferences, i.e. consensus must match at least 50% of inputted sequences. The output of PRATT is in PROSITE format and feed directly into the PROSITE database search [9].

2.2. *Database search*: PROSITE is a database that can be searched when a Motif is entered as a parameter (http://www.ExPASy.org/prosite/). The website is able to understand patterns and motifs by using the accepted one letter abbreviations of the amino acids (e.g. G is Glycine), i.e. the usual format of a typical pattern/motif can be something like:

G-[ASP]-V-X(2)-GLA-SP,

where letters correspond to their aminoacids, '-' separates the next aminoacid, X calls for any aminoacid (the number in the brackets determines how many of the preceding letter), {∗} -aminoacids within curly braces are not to be included in the pattern/Motif search.

Example of interpretation: "Starting with a Glycine, followed by Alanine, Serine or Proline, followed by a Valine, then by any 2 aminoacids, followed the aminoacid order Glycine, Leucine and Alanine and finally ending on any aminoacid barring Serine or Proline".

3. Results

Potassium - GATE: The gate (voltage sensor) of the Kv channel currently has two known folds targeting it. These folds are: $\beta\beta\beta$, $3_{10}\beta\beta$ and 2 non-categorised folds $\beta\beta$ and $\beta\beta\beta\beta$.

The consensus that was applied to the structures was:

C-X(3)-[WMILF]-X(9,10)-C-X(0,3)-[REKH]-X(1,5)-C-X(3,10)-C

This consensus finds 19 out of the 20 toxins that have been classed to target the potassium gate. The only toxin that does not contain this consensus is Kappa-conotoxin BtX from the Cone snail.

3.1. **POTASSIUM-PORE**: Starting with the potassium channel pore consensus,

C-X(9,12)-C-X(2,5)-C,

Folds include: $\beta\beta\beta$, $\alpha\alpha$ (hairpin), $\alpha\alpha$ (helical cross), $3_{10}\alpha\alpha$, $\alpha\beta$, $\beta\alpha\beta\beta$ and $3_{10}\beta\beta\alpha$

$\beta\beta\beta$ PRATT ouput: CXK7A-CONPU∗: 8- 26: Cfqhlddccsrk-CnrfnkC.

3.2. **SODIUM-PORE**: Site 1 is the pore of the sodium channel. The folds that recognize the Nav Pore are: $\beta\beta$, $\beta\beta\beta$ and $\beta\alpha\beta\beta$. The Pore consensus will be applied to a structure that represents each fold.

C-x(3,6)-C-x(4,6)-C-[ACGN].

Matches 15 out of 16 known Nav pore blocking toxins.
Example displayed: $\beta\beta$-Conotoxin GS. CXGS-CONGE: 9-20: grgsr Cppqc–Cmglr–CGrgnpq

3.3. **SODIUM-GATE**: Site 3 best consensus for C-X(6,9)-C-X(0,6)-C
This consensus matches some toxin sequences six times, Neurotoxin Tx2-6 from the Brazilian armed spider. As a result alignment B will be chosen as it matches some toxin sequences only 3 times.

C-x(5,7)-C-x(10,13)-C

All folds except $\beta\beta\alpha\beta\beta\alpha$ target site 3 of the sodium channel. Each fold type will be shown $\beta\beta$ Robustoxin (funnel web spider)

TXDT1-ATRRO: 1-20: Cakkrnw-Cgknedcccpmk–C iyawy
TXDT1-ATRRO:14-31: gkned Cccpmk–Ciyawynqqgs—Cqttit

Finally, a consensus was determined for all potassium and sodium channel toxins. This produced an interesting output, in that the individual consensus sequences seem to be the reverse of each other.
Potassium toxins C-X(9,12)-C-X(2,6)-C,
Sodium toxins C-X(3,6)-C-X(5,9)-C.
The potassium toxin consensus has a long spacer region between the first and second Cystiene and the sodium has a longer spacer region between the second and third Cystiene. The predominant type of channel toxin for the potassium channel seems to be Pore specific and for the sodium channel the gate conversely. The relationships between these observations are not clear but may help to understand the differences between the two types of ion channel toxins.

In our previous work [2], we have noticed that there were 2 unreported potassium channel gate modifiers -with the $\beta\beta$ and $\beta\beta\beta\beta$ folds. It is well known that the $\beta\beta$ fold is found in conotoxins, but these act on sodium channels as pore blockers, not gate modifiers. Our initial study of these toxins suggested that these 2 motifs are also able to bind to potassium channels as well. Table 1 summarizes statistics of the overall consensus sequences for each of the categories.

Table 1. The summary statistic

Target	Number	Consensus Sequence	% Found
K Pore	127	C-X(9,12)-C-X(2,5)-C	93.7%
K Gate	20	C-X(3)-[WMILF]-X(9,10)-C-X(0,3)-[REKH]-X(1,5)-C-X(3,10)-C	95.0%
Na Pore	16	C-X(3,6)-C-X(4,6)-C-[ACGN]	93.75%
Na Gate	247	C-X(3,6)-C-X(9,11)-C	93.9%
All K	145	C-X(11,14)-C-X(2,5)-C	91.72%
All Na	262	C-X(3,6)-C-X(10,13)-C	93.13%

This result was based on motif matching from our selected database extracted from various toxin repositories. We sought to strengthen the case for these two new putative potassium gate modifiers in this study.

To provide additional support for this hypothesis, we generated a classifier based on a self-organized type of neural network -the *learning vector quantization* (LVQ2.1). LVQ is a supervised version of vector quantization, similar to Self-Organizing-Maps (SOM). LVQ can be understood as a special case of an artificial neural network, applying a winner-take-all Hebbian learning-based approach. It can be applied to pattern recognition, multi-class classification and data compression tasks. LVQ algorithms directly define class boundaries based on prototypes, a nearest-neighbour rule and a winner-takes-it-all paradigm. The main idea is to cover the input space of samples with "codebook vectors" (CV), each representing a region labelled with a class. A CV can be seen as a prototype of a class member, localized in the centre of a class or decision region ("Voronoi cell") in the input space. As a result, the space is partitioned by a "Voronoi net" of hyperplanes perpendicular to the linking line of two CVs (mid-planes of the lines forming the "Delaunay net"). A class can be represented by an arbitrarily number of CVs, but one CV represents one class only. In terms of neural networks a LVQ is a feedforward net with one hidden layer with Kohonen neurons, adjustable weights between input and hidden layer and a winner takes it all mechanism. The basic LVQ algorithm -LVQ1- tends to push CVs away from Bayes decision surfaces. In order to get better approximations of the Bayes rule by pairwise adjustments of two CVs belonging to adjacent classes, in LVQ2 (LVQ2.1 too) adaptation only occurs in regions with cases of misclassification in order to get finer and better class boundaries. To conclude, a main advantage of using LVQ is that it creates prototypes that are easy to interpret for experts in the field.

The code vectors were selected randomly from the set of sequences that were found during our motif search (see Table 1 for details). After seeding the code book vectors by randomly selecting members from of each of the classes (pore blockers - sodium/potassium and gate modifiers - sodium/potassium), the rest of the candidates from the consensus search were presented to the network. If the putative consensus were indeed from a different class then the others, they would tend to aggregate in the vicinity of their respective code-book vectors. A critical

feature in this type of neural network is a distance metric. We encoded the motifs according to aminoacid letter and number of residues. We therefore presented the sequences according to the motifs presented in Table 1. An X(3,6) for instance is matched with the letter 'X' and the numbers 3-6 in the form of a set of strings XXX, XXXX, XXXXX, XXXXX. Then we used the ordinal position within the alphabet as the distance metric to calculate which code book vector a particular consensus sequence was closest too. We performed a 10-fold cross validation, partitioning the dataset into disjoint sets of 10 randomly selected elements until all of the consensus sequences were used as code book vectors. The results indicate that the two new sequences did map to the potassium gate modifier, indicating that they were closest to this code book vector than to any other code book vector.

4. Conclusions

In this study we developed a set of consensus sequences that could differentiate potassium from sodium channel acting toxins. Our methodology was able to generate consensus sequences for potassium gate and pore blockers (2 and 3 respectively) as well as sodium channel gate and pore blockers (4 and 5 respectively). In addition, we were able to come up with a consensus sequence that was able to generically differentiate potassium from sodium channel toxins (6 and 7 respectively). No literature report has provided a consensus sequence that could distinguish pore from gate blockers for either sodium or potassium channels and this is the novel result from this study. When the consensus sequences we have generated are entered into a standard protein sequence database such as PDB, we find that the accuracy, in terms of number of hits per known number of toxins acting on that particular site, is quite high -on the order of 93% or higher in this study. It should be noted however that consensus generation has been optimal when sequence numbers are low or toxins are from a related family. The toxins that act on the channels in this project are from a wide range of organisms.

This approach provides additional evidence that there are two more gate modifier motifs: $\beta\beta$ and $\beta\beta\beta\beta$. The $\beta\beta$ motif is similar to a subset of toxins that are sodium channel pore blockers, yet there is no evidence reported in the literature that indicates they bind the gate (voltage sensor) of potassium channels to our knowledge. This is still a preliminary result and further work must be performed to strengthen the case for this argument. If it holds true, then there will be a total of 10 potassium channel motifs, although one may cross-react with one or more sodium channel pore(s) and potassium gate(s).

References

[1] Boeckmann B., Bairoch A., Apweiler R., Blatter M.C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S. and Schneider

M., 2003. The Swiss-Prot protein knowledgebase and its supplement TrEMBL, Nucleic Acids Res. 31, pp. 365-370.

[2] Bhogal S., Revett, K., 2005, Animal Toxins: What Features Differentiate Pore Blockers From Gate Modifiers, CIMA 2005 Conference, Istanbul Turkey.

[3] Bucher P., Bairoch A., 1994, A generalized profile syntax for biomolecular sequences motifs and its function in automatic sequence interpretation, Proceedings 2nd International Conference on Intelligent Systems for Molecular Biology.

[4] Bucher P., 2002, PROSITE: a documented database using patterns and profiles as motif descriptors, Brief Bioinform, 3, pp. 265-274.

[5] Elsworth J., Deadly by Nature, Web Project, http://www.chm.bris.ac.uk/webprojects 2002/elsworth/deadly-by-nature.htm

[6] Jungo F., Bairoch A., 2994, Tox-Prot, the toxin protein annotation program of the Swiss-Prot protein Knowledgebase, Toxicon., 45, pp. 293-301.

[7] Miller C., Naranjo D., 1996, A strongly interacting pair of residues on the contact surface Charybdotoxin and a shaker channel, Neuron, 16, pp. 123-130.

[8] Possani L.D., Merino E., Corona M., Bolivar F. and Becerril B., 2000, Peptides and genes coding for scorpion toxins that effect ion-channels, Biochemie., 82, pp. 861-868.

[9] Schonbach C., Kowalski-Saunders and Brusic V., 2000, Data warehousing in molecular biology, Brief Bioinformatics., 1, pp. 190-198.

(3) University of Westminster, Harrow School of Computer Science, London, UK
E-mail address: revettk@westminster.ac.uk

(1) Universitatea de Medicina si Farmacie din Craiova, str. Petru Rares, Nr. 2-4
E-mail address: fgorun@rdslink.ro

(2) Universitatea din Craiova, Str. A.I. Cuza, Nr. 13
E-mail address: mgorun@inf.ucv.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 138–144

COLLABORATIVE SELECTION FOR EVOLUTIONARY ALGORITHMS

ANCA GOG [1] AND D. DUMITRESCU [2]

Abstract. A new selection scheme for evolutionary algorithms is proposed. The introduced selection operator is based on the collaboration between individuals that exchange information in order to accelerate the search process. The NP-hard Travelling Salesman Problem is considered for testing the proposed approach. Numerical experiments prove the efficiency of the proposed technique, compared with the most popular selection operators used within evolutionary algorithms.

1. Introduction

A new selection operator is proposed in order to improve the search process of the evolutionary algorithms. Unlike the standard evolutionary algorithm, in the proposed approach each individual has information about its best related individual. The new collaborative selection operator is based on this extra information that each individual has and tends to favor the fittest individuals within each group of individuals having a common best ancestor.

The proposed Collaborative Selection (CS) operator is compared to the most popular selection operators. Several instances of the NP-hard Travelling Salesman Problem (TSP) are considered in order to prove the efficiency of the proposed operator.

The paper is organized as follows: Section 2 presents some of the existing selection operators; Section 3 describes the new proposed Collaborative Selection operator; Section 4 contains experimental results and there are conclusions in Section 5.

2. Selection Operators

The purpose of the selection procedure is to choose from the current population the set of individuals (parents) that will be used to create the next generation. The

2000 *Mathematics Subject Classification.* 68T20, 68T99.

Key words and phrases. Collaborative Selection, Travelling Salesman Problem, Combinatorial Optimization.

©2007 Babeş-Bolyai University, Cluj-Napoca

choice of which individuals are allowed for reproducing determines which regions of the search space will be visited next. This choice is often the result of a trade-off between exploration and exploitation of the search space. Some of the most popular selection operators [1] are briefly described in what follows.

Roulette Selection is used to generate a uniform probability distribution. This mechanism ensures the selection of the individual x^i with probability:

$$p_i = \frac{f(x^i)}{F}, i = 1, 2, \ldots, n,$$

where F is the total fitness of the population, defined as the sum of all individuals' fitness and $f(x^i)$ is the fitness of the individual x^i.

In *Linear Ranking Selection* at each generation the individuals are sorted according to their fitness and a rank is assigned to each individual in the sorted population. The selection probabilities of the individuals are given by their rank in the population.

In *Tournament Selection N* individuals are chosen from the population in order to produce a tournament subset of chromosomes. The best chromosome in this subset is then selected.

Best Selection operator selects the best chromosome determined by fitness. If there are two or more chromosomes with the same best fitness, one of them is chosen randomly.

3. Collaborative Selection Operator

In evolutionary algorithms, one individual (or chromosome) encodes a potential solution of the problem and is composed by a set of elements called genes. Each gene can take multiple values called alleles. In the proposed collaborative approach an individual has extra information regarding its best related individual, the so-called *LineOpt* (related individuals refer to all individuals that have existed in one of the previous generations and have contributed to the creation of the current individual: its parents, the parents of its parents, and so on).

Selection operator should provide a good equilibrium between the exploration and the exploitation of the search space. On one hand, selection has to provide high reproductive chances to the fittest individuals; on the other hand, selection must preserve population diversity in order to explore all promising search space regions.

Selection operator chooses which individuals should enter the mating pool in order to be subject of recombination. On this purpose, the proposed selection operator called *Collaborative Selection (CS)* uses the information about the *LineOpt* of each individual. This selection is essentially rank-based.

Let us assume that the current population *P(t)* has n individuals:

$$P(t) = \{x^1, x^1, \ldots, x^n\}.$$

Let f be the fitness function. Within Monte Carlo selection mechanism [1] the selection probability of the individual x_i is the number p_i defined as:

$$p_i = \frac{f(x^i)}{F}, i = 1, 2, \ldots, n,$$

where F is the *total fitness* of the population.

In order to prevent an individual or a group of high-fit individuals from dominating the next generation the introduced CS operator uses rank information.

All individuals within the current population are grouped by their *LineOpt*. Considering the current population *P(t)*, clusters $A_1, \ldots, A_k, k \leq n$, are formed according to the rules:
(i) the clusters $A_1, \ldots, A_k, k \leq n$ represent a partition of *P(t)*:

$$(a) \quad A_i \neq \phi, 1 \leq i \leq k,$$

$$(b) \quad \bigcup_{i=1}^{k} A_i = P(t),$$

$$(c) \quad A_i \bigcap A_j = \phi, 1 \leq i \leq k, 1 \leq j \leq k, i \neq j.$$

(ii) all the individuals that belong to the cluster $A_i (1 \leq i \leq k)$, have the same *LineOpt*:

$$LineOpt(x^i) = LineOpt(x^j),$$
$$\forall x^i, x^j \in A_l, 1 \leq l \leq k.$$

(iii) every two different clusters $A_i, A_j (1 \leq i \leq k, 1 \leq j \leq k, i \neq j)$ have a different *LineOpt*:

$$LineOpt(x^i) \neq LineOpt(x^j),$$
$$\forall x^i \in A_i, x^j \in A_j, 1 \leq i \leq k, 1 \leq j \leq k, i \neq j.$$

Let us suppose that there are two individuals in two different clusters having the same fitness value:

$$x^i \in A_i, x^j \in A_j, 1 \leq i \leq k, 1 \leq j \leq k, i \neq j,$$
$$f(x^i) = f(x^j).$$

Furthermore, let us suppose that x^i is the fittest individual in the cluster A_i and that x^j is the worst individual in the cluster A_j. The Monte Carlo technique assigns the same probability of being selected to individuals having the same fitness. The CS operator favors the individual x^i by assigning it a higher probability of being selected than the probability of the individual x^j, even if both of them have the same fitness. The reason is the fact that x^i is the fittest individual of the cluster A_i, while x^j is the worst individual of the cluster A_j. The goal of the

proposed strategy is to favor the selection of the fittest individuals within each cluster, i.e. within each group of individuals having the same *LineOpt*.

This goal is achieved by modifying the selection probability p_i. Within CS individuals in each cluster are ranked according to their relative fitness. A pure rank-based selection scheme (like tournament) may be used. A different approach modifies the selection probability according to the rank. In this case the selection probability p_i of an individual is modified according to the rank of the individual in its cluster but in such a way that the sum of renormalized probabilities p_i remains 1.

Renormalized probability is computed for each individual according to its membership to a cluster and its rank in that cluster. Let us suppose that the cluster $A_i (1 \leq i \leq k)$, contains the individuals $x^j, (1 \leq j \leq |A_i|)$. For each individual x^j the relative rank in the cluster A_i is the number pc_j computed as:

$$pc_j = \frac{rank(x^j)}{\sum_{l=1}^{|A_i|} l}, j = 1, 2, \ldots, |A_i|.$$

The selection probability of the individuals x^j belonging to the cluster A_i is modified according to their relative ranks. The new probability new_p_j of the individual x^j is computed as follows:

$$new\_p_j = pc_j * S,$$

where S is the sum of the probabilities p_j for all individuals x^j from the cluster A_i. This way, the sum of the new probabilities for all individuals from a cluster satisfies:

$$\sum_{x^j \in A_i} pc_j = 1.$$

Indeed we may successively write,

$$\sum_{x^j \in A_i} new\_p_j = \sum_{x^j \in A_i} pc_j S = S \sum_{x^j \in A_i} pc_j = S = \sum_{x^j \in A_i} p_j.$$

The sum *1* for the new computed probabilities for all the individuals within the population is conserved.

The new probabilities favor the fittest individuals as well as the fittest individuals within each cluster ensuring that good genetic material already obtained is preserved. This promotes the exploitation of all promising regions discovered during the search process therefore preventing a group of high-fit individuals from dominating the next generation.

4. Experimental results

Evolutionary Computation provides good approximate methods for solving Combinatorial Optimization Problems, especially NP-complete and NP-hard problems [2], [5]. One representative combinatorial optimization problem, namely Travelling Salesman Problem (TSP) [3] is investigated to prove the efficiency of the proposed selection operator. A set of k points in a plane is given, corresponding to the location of k cities. The Travelling Salesman Problem requires finding the shortest closed path that visits each city exactly once. The problem can be formalized as follows:

A set of k cities

$$C = \{c_1, c_2, \ldots, c_k\}$$

is given. For each pair

$$(c_i, c_j), i \neq j,$$

let

$$d(c_i, c_j)$$

be the distance between the city c_i and the city c_j. One has to find a permutation π' of the cities

$$(c_{\pi'(1)}, \ldots, c_{\pi'(k)}),$$

such that

$$\sum_{i=1}^{k} d(c_{\pi'(i)}, c_{\pi'(i+1)}) \leq \sum_{i=1}^{k} d(c_{\pi(i)}, c_{\pi(i+1)}),$$

$$\forall \pi \neq \pi', (k+1 \equiv 1).$$

This problem can be defined as the search for a minimal Hamiltonian cycle in a complete graph.

The simplest evolutionary approach of this NP-hard problem is outlined in what follows. A potential solution for the problem (a chromosome) is a string of length k that contains a permutation π of the set

$$\{1, \ldots, k\},$$

and represents the order of visiting the k cities. A chromosome is evaluated by means of a fitness function f that needs to be minimized:

$$f : S \to \Re^+, f(\pi) = \sum_{i=1}^{k} d(c_{\pi(i)}, c_{\pi(i+1)}),$$

$$(k+1 \equiv 1),$$

where S represents the search space of the problem, i.e. the set of all permutations π of the set

$$\{1, \ldots, k\}.$$

Thus, the fitness of a chromosome is the length of the closed path that visits the cities in the order specified by the permutation π.

A Standard Genetic Algorithm (SGA) with OX recombination and inverse mutation [1] is considered for numerical experiments. Several TSP instances taken from TSPLIB are investigated [4]. For the considered problems, SGA is applied with Collaborative Selection and with all selection operators described in Section 2: Roulette Selection, Linear Rank Selection, Tournament Selection and Best Selection. Table 1 and Table 2 contain the results obtained after 10 runs of the SGA with all considered selection operators. Results regard the average solution obtained after 100 and after 500 generations. The best values obtained are bolded in both tables.

TABLE 1. Average results obtained after 10 runs of SGA (after 100 generations) with all considered selection operators.

TSP instance	**Roulette Selection**	**Linear Rank Selection**	**Tournament Selection**	**Best Selection**	**Collaborative Selection**
EIL51	878	745	**753**	**753**	755
ST70	2063	1707	1726	1713	**1672**
PR76	328475	290645	288150	285725	**278838**
EIL76	1439	1281	1294	1276	**1271**
KROA100	95431	82792	**81908**	84672	82225
KROB100	93425	79438	81398	86447	**78462**
KROC100	95164	83840	81577	83297	**80989**
KROD100	92065	81609	**79202**	81549	81604
KROE100	97615	84903	82678	87209	**79437**
EIL101	2053	1890	1862	1838	**1807**

The test results indicate the acceleration of the search process when using CS, especially in the first generations of the algorithm, compared with all the other selection operators. In the latest stages of the algorithm, the only selection operator that outperforms CS is the tournament selection, but CS outperforms the other selection operators in most of the cases.

5. Conclusions and Further Work

A new collaborative selection operator (CS) for evolutionary algorithms has been proposed. CS is using extra information regarding the best ancestor of each individual obtained so far by the search process. Numerical experiments, for which several instances of TSP have been used, have proved that the proposed selection outperforms most of the existing selection operators by accelerating the search process.

TABLE 2. Average results obtained after 10 runs of SGA (after 500 generations) with all considered selection operators.

TSP instance	**Roulette Selection**	**Linear Rank Selection**	**Tournament Selection**	**Best Selection**	**Collaborative Selection**
EIL51	550	514	**498**	499	**498**
ST70	1102	981	**942**	980	962
PR76	187717	158380	**151715**	159378	160502
EIL76	841	763	761	768	**738**
KROA100	51281	41869	**41389**	42315	42765
KROB100	48812	42164	42361	42769	**41996**
KROC100	49882	41344	**40934**	42450	41531
KROD100	48318	40023	**39998**	42179	41579
KROE100	51897	**39871**	41006	42770	40780
EIL101	1205	1039	**1028**	1047	1046

REFERENCES

[1] Dumitrescu, D., Lazzerini, B., Jain, L.C. and Dumitrescu, A., Evolutionary Computation, CRC Press, Boca Raton, FL., 2000.

[2] Blum, C., Roli, A., Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, ACM Computing Surveys, Vol. 35, 268-308, 2003.

[3] Gutin, G., Punnen, A.P., Traveling Salesman Problem and Its Variations, Kluwer Academic Publishers, 2002.

[4] Reinelt, G., TSPLIB - A Traveling Salesman Problem Library, ORSA Journal of Computing, 376-384, 1991.

[5] Alba Torres, E., Khuri, S., Applying Evolutionary Algorithms to Combinatorial Optimization Problems, ICCS 2001, LNCS 2074, Springer Verlag Berlin Heidelberg, 689-698, 2004.

[1] BABES-BOLYAI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
KOGALNICEANU 1,
R0 - 400084 CLUJ-NAPOCA
E-mail address: anca@cs.ubbcluj.ro

[2] BABES-BOLYAI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
KOGALNICEANU 1,
R0 - 400084 CLUJ-NAPOCA
E-mail address: ddumitr@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 145–152

GENETIC CHROMODYNAMICS. DATA MINING AND TRAINING APPLICATIONS

D. DUMITRESCU[(1)], KÁROLY SIMON[(2)], AND ELEONÓRA VÍG[(3)]

Abstract. To avoid some usual difficulties of standard evolutionary algorithms recently a new multimodal optimization metaheuristics - called Genetic Chromodynamics (GC) - has been proposed. Based on the GC metaheuristics a new dynamic evolutionary clustering technique (GCDC) has been developed. Some applications of GCDC are presented. GCDC is used for gene expression analysis. A GCDC-based method for designing optimal neural network topologies is also presented.

Evolutionary algorithms represent ideal tools for solving difficult optimization problems [3]. Several optimization problems for which classical methods do not work very well or are simply inapplicable can be solved with evolutionary techniques. Evolutionary algorithms can be used for constrained, dynamic, multiobjective and multimodal optimization.

Standard evolutionary algorithms find only one solution, even if the search space is a highly multimodal domain. In order to identify several optimum points special evolutionary models have been proposed. In some cases classical evolutionary multimodal optimization methods, like niching techniques, cannot focus the search on each optimum and find the optimal solutions efficiently.

To avoid some usual difficulties of these standard algorithms recently a new multimodal optimization metaheuristics - called Genetic Chromodynamics (GC) - has been proposed [4]. The model may be used to solve real- world optimization problems including static and dynamic multimodal and multiobjective optimization problems. GC-based techniques can be applied in various scientific, engineering or business fields. Clustering, learning from data, data compression and other data mining problems are very suitable for a GC treatment.

Based on the GC metaheuristics a new clustering technique - called GC-based Dynamic Clustering (GCDC) - has been proposed [7]. Dynamic clustering is a typical multi-modal optimization problem. The problem of cluster optimization is

2000 *Mathematics Subject Classification.* 68Q05, 62H30, 68T05.

Key words and phrases. evolutionary multimodal optimization, dynamic clustering, RBF neural networks, gene expression analysis.

©2007 Babeş-Bolyai University, Cluj-Napoca

twofold: optimization of cluster centers and determination of number of clusters. The latter aspect has often been neglected in standard approaches (static clustering methods) (see [12, 13]), as these typically fix the number of clusters *a priori*. In case of practical problems the number of existing clusters is generally unknown. Dynamic clustering does not require *a priori* specification of the number of clusters. GC-based clustering can be particularly useful to detect the optimal number of clusters in a data set and the corresponding set of useful prototypes [7].

Clustering is a useful exploratory tool in gene expression data, however there are only a few works that deal with the problem of automatically estimating the number of clusters in bioinformatics datasets. GCDC is capable of automatically discovering the optimal number of clusters and its corresponding optimal partition in gene expression datasets.

Solving a problem with a neural network a primordial task is the determination of the network topology. Generally the determination of the neural network topology is a complex problem and cannot be easily solved. When the number of trainable layers and processor units (neurons) is too low, the network is not able to learn the proposed problem. If the number of layers and neurons is too high then the learning process becomes too slow. The main aim is designing optimal topology. In some cases complexity of networks can be reduced by clustering the training data.

In Section 1 the GC metaheuristics and the GC-based dynamic clustering technique are presented. In Section 2 GCDC is used for gene expression analysis. A method for designing optimal RBF neural network topologies using GCDC is presented in Section 3. Some numerical experiments are also described.

1. GC-based dynamic clustering

Genetic Chromodynamics (GC) [4] is a new kind of evolutionary search and optimization metaheuristics. GC is a metaheuristics for maintaining population diversity and for detecting multiple optima. The main idea of the strategy is to force the formation and maintenance of stable sub-populations.

GC-based methods use a variable-sized population, a stepping-stone search mechanism, a local interaction principle and a new operator for merging very close individuals.

Corresponding to the stepping-stone technique each individual in the population has the possibility to contribute to the next generation and thus to the search progress. Corresponding to the local interaction principle the recombination mate of a given individual is selected within a determined mating region. Only short range interactions between solutions are allowed. Local mate selection is done according to the values of the fitness function. An adaptation mechanism can be used to control the interaction range, so as to support sub-population stabilization. Within this adaptation mechanism the interaction radius of each individual could be different.

To enhance GC, micropopulation models can be used. Corresponding to these models, for each individual a local interaction domain is considered. Individuals within this domain represent a micropopulation. All solutions from a micropopulation are recombined using local tournament selection. When the local domain of an individual is empty the individual is mutated.

Within GC sub-populations co-evolve and eventually converge towards several optima. The number of individuals in the current population usually changes with the generation. A merging operator is used for merging very close individuals. At convergence, the number of sub-populations equals the number of optima. Each final sub-population hopefully contains a single individual representing an optimum, a solution of the problem.

GC allows any data structure suitable for the problem together with any set of meaningful variation/search operators. For instance solutions may be represented as real-component vectors. Moreover the proposed approach is independent of the solution representation.

Based on the GC metaheuristics a new dynamic clustering algorithm - called GCDC - has been developed. This technique is described below.

1.1. **Solution representation.** Corresponding to the proposed GCDC method each cluster is represented by a prototype (cluster center). Each prototype is encoded into a chromosome. The initial population is randomly generated and it contains a large number of individuals.

1.2. **Interaction domain.** For realizing the local interaction principle, an interaction domain (mating region) is considered for each individual in the population (a chromosome representing a prototype). To support subpopulation stabilization an adaptation mechanism is used for controlling interaction domains [9]. For realizing the stepping-stone search principle a micropopulation model is used and it is combined with direct survival competition.

1.3. **Search operators.** The crossover operation can be a convex combination of the parent genes. A randomly generated number for each gene can be considered as combination coefficient. An additive perturbation of genes with a randomly chosen value from a normal distribution N(0,σ), where σ is a control parameter called *mutation step size* can be considered as mutation operator.

1.4. **Fitness evaluation.** Fitness values of individuals are evaluated using suitable fitness functions. For instance Gaussian functions could be used (see [7, 9]).

The set of input samples $X = \{x_1, ..., x_n\}$ is considered. Cluster structure corresponding to this input data set is given by a set of prototypes $L = \{L_1, ..., L_m\}$, represented by chromosomes. Fitness of a chromosome L_j is calculated using the

following Gaussian fitness function:

$$g(L_j) = \sum_{i=1}^{n} e^{-\frac{\|x_i - L_j\|^2}{\gamma_j^2}}.$$

Parameters of corresponding normal distribution are L_j and γ_j. An adaptation mechanism is used for controlling parameter $\gamma_j, j = 1, ..., m$. In this way a dynamical adaptation of the fitness function is realized.

1.5. **Improving GCDC.** To achieve a better performance final merging and a post-processing methods can be performed [10]. A Link-Cell method can be applied for improving GCDC by promoting local search and deriving new parameter adaptation techniques [11].

2. GCDC for gene expression analysis

Gene expression analysis is of great importance in molecular biology for inferring the functions and structures of a cell since changes in the physiology of an organism are accompanied by changes in the pattern of gene expression.

Gene expression is the process by which a gene's coded information is converted into the structures and functions of a cell. Expressed genes include those that are transcribed into mRNA. The amount of protein that a gene expresses depends on the tissue, the developmental stage of the organism and the metabolic or physiologic stage of the cell. By capturing the cell expression level, biologists can build up a picture of what levels of gene expression may be normal, or abnormal, and what the relative expression levels are between different genes within the same cell [1, 2].

DNA microarray, a recently developed technology, allows thousands of gene expression levels to be measured simultaneously. Data collected by this method is called gene expression data, which after preprocessing (reduction of the noise-level and normalization), forms the data source of our clustering algorithms.

The main characteristics of gene expression data is the very high number of genes (up to 10^6), and the generally small number of samples (< 100). Thus gene expression data is usually represented by a real-valued matrix whose rows correspond to genes and whose columns correspond to conditions, experiments or time points. An element of the matrix represents the expression level of a specific gene under a specific condition.

2.1. **Clustering gene expression data.** Clustering is a fundamental and widely used technique in data analysis and pattern discovery aimed at a better understanding of gene structure, function and regulation. During clustering genes are systematically grouped together according to their similarity in expression patterns. Performing cluster analysis on gene expression data can help detecting gene groups with similar expression patterns, determining the function of new genes,

finding correlation between different groups, understanding gene regulation and cellular processes, observing gene expression differentiation in various diseases or drug treatments, thus digging out biologically meaningful information from genetic data. Multi-gene expression patterns could characterize diseases and lead to new precise diagnostic tools capable of discriminating different kinds of cancers [2].

The desired features of data analysis techniques dealing with gene expression data are robustness, understandability, fastness and automatic detection of the optimal cluster-number. The key challenges regarding gene clustering are the development of methods that can extract order across experiments in typical datasets of size 30000 x 1000, methods which can deal with highly connected, intersecting or even embedded clusters. Boundaries between clusters can be very noisy. There is a need for algorithms that handle effectively these problems.

Several classical clustering and classification algorithms have been applied to gene-expression data from k-means to hierarchical clustering, principal component analysis, factor analysis, independent component analysis, self-organizing maps, decision trees, neural networks, support vector machines, graph-theoretic approaches, and Bayesian networks to name a few. Each method has different advantages depending on the specific task and specific properties of the data set being analyzed. Typically, simpler methods are more robust, while the advanced approaches provide more accurate results [5, 2].

In most clustering methods setting the number of clusters beforehand is necessary; however, the choice of the number K of clusters is a delicate issue, and only a few works deal with the automatic estimation of the number of clusters in bioinformatics datasets.

2.2. **Numerical experiments.** We have chosen a data set for which a biologically meaningful partition into classes is known in the literature. We refer to that partition as the true solution.

RCNS data set [14] contains the expression levels of 112 genes during rat central nervous system development over 9 time points. According to Wen et al. the true partition of this data set contains 6 classes, four of which are composed of functionally related genes. In order to capture the temporal nature of the data, the difference between the values of two consecutive data points is added as an extra data point. Therefore, the final data set consists of a 112 x 17 data matrix. This transformation enhances the similarity between genes.

The GCDC technique provides an optimal clusters-center set containing usually from 5 to 8 solutions; however, the most frequently appearing cluster number is 6.

The slight differences in result sets after multiple runs of the algorithms are due to the stochastic nature of the method. The use of random numbers to pick crossover and mutation locations embed stochastic processes into the algorithm.

Figure 1 shows the average normalized expression pattern over the 9 time points for all the genes in each cluster. These plots are very similar for multiple runs of the algorithm; however the starting dominant ancestor might be different.

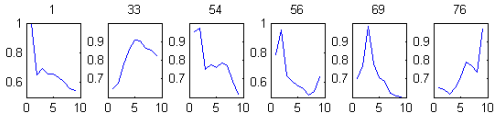


FIGURE 1. Average normalized expression pattern over the 9 time points for all the genes in each cluster. The numbers above each graph represent the indices of the starting dominant ancestors of the different clusters.

The partition retrieved by the algorithm does not correspond entirely to the original partition found by Wen et al., however it also makes sense. It finds a refinement of some initial classes, while others are grouped together. In all cases genes clustered in the same class share similar expression patterns, and a biological interpretation of the classification is also possible.

3. GCDC for designing RBF neural networks

Radial Basis Function (RBF) networks are relatively simple neural networks, especially used for solving interpolation problems [6]. RBF is a feed-forward neural network with an input layer (made up of source nodes: sensory units), a single hidden layer and an output layer. Within RBF networks there is a dependence between the number of training samples and the number of hidden neurons.

Complexity of RBF networks depends on the number of hidden neurons. This complexity can be reduced by clustering the training data. The number of hidden neurons supplies the number of radial basis functions with different centers. It should be favorable the use of training samples as RBF centers, but in some cases this is impossible. If the number of training samples is high, then not all of them might be used (the number of hidden processor units must be reduced). The solution is to consider a single neuron for a group of similar training points. Groups of similar training points can be identified by using clustering methods.

GCDC does not require *a priori* specification of the number of clusters. Therefore GCDC can be used for designing optimal RBF neural network topologies [8].

The number of neurons in the hidden layer of the network is the number of clusters determined by the GCDC method. Cluster centers identified by the GCDC algorithm are used as center parameters for the activation functions. RBF function parameters can be determined according to the cluster diameters. In this way optimal RBF neural network topology can be obtained.

For investigating the performance of the GCDC method a numerical experiment is performed. RBF neural network is used for approximating the function:

$$F_2(x) = 2 \cdot \sin\left(\ln(x) \cdot e^{\cos\left(\frac{x}{2}\right)}\right),$$

where $0 \leq x \leq 9.5$.

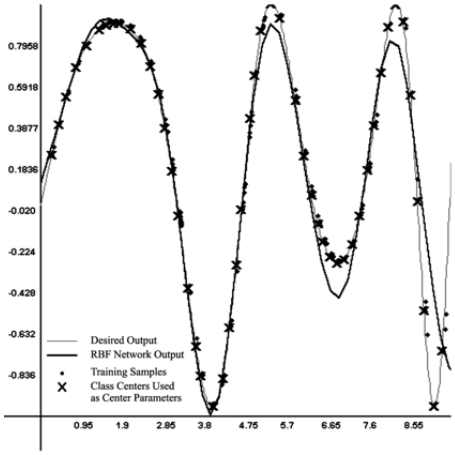


FIGURE 2. 200 training samples organized in 50 clusters, centers determined by the GCDC technique, output of the RBF network after 10000 training epochs.

The network is trained using 10 data sets. Each training set consists of 200 points from the interval [0,9.5]. In each set the points are organized in 50 well-separated clusters. For each set the GCDC method is performed and RBF neural network topologies are created based on the returned results. In 5 cases the number of centers determined by GCDC is 50. In other 5 cases there is a little difference (maximum +4). The *generalization error* is calculated using $M = 400$ inputs (that do not belong to the training set) from the interval [0,9.5]. After training the obtained RBF networks, the mean generalization error is 0.539953496. Satisfactory approximation results are obtained (Figure 2).

The GCDC method has been compared with standard (static) clustering methods. Better results were obtained by using GCDC.

4. Conclusions

Genetic Chromodynamics (GC) a new evolutionary optimization metaheuristics aimed for addressing static and dynamic multimodal and multiobjective optimization problems is described.

A GC-based clustering technique - called GCDC - is proposed. GCDC is used for gene expression analysis and for designing RBF network topology. Numerical experiments indicate the potential of the proposed approach.

References

[1] Attwood T. K., Parry-Smith D. J., Introduction to Bioinformatics, Pretince Hall, 1999Attwood T. K., Parry-Smith D. J., Introduction to Bioinformatics, Pretince Hall, 1999

[2] Baldi P., Hatfield G. W., DNA Microarrays and Gene Expression - From experiments to data analysis and modeling, Cambridge University Press, Cambridge, 2002.

[3] Dumitrescu D., Lazzerini B., Jain L. C., Dumitrescu A., *Evolutionary Computation*, CRC Press, Boca Raton, 2000.

[4] Dumitrescu D., *Genetic Chromodynamics*, Studia Univ. Babes-Bolyai, Ser. Informatica, 35 (2000), pp. 39-50.

[5] Kohane I. S., Kho A. T., Butte A. J., Microarrays for an integrative genomics, MIT Press, 2003.

[6] Powell M. J. D., *Radial basis functions for multivariable interpolation: A review*, in Algorithms for Approximation, J. C. Mason and M. G. Cox, ed., Clarendon Press, Oxford, (1987), pp. 143-167.

[7] Dumitrescu D., Simon K., *Evolutionary prototype selection*, Proceedings of ICTAMI, (2003), pp. 183-191.

[8] Dumitrescu D., Simon K., Genetic Chromodynamics for designing RBF neural networks, Proceedings of SYNASC, (2003) pp. 91-101.

[9] Dumitrescu D., Simon K., *Fitness functions and interaction domain adaptation mechanisms for dynamic evolutionary clustering*, Proceedings of ICCC, (2004), pp. 132-138.

[10] Dumitrescu D., Simon K., *Post-processing techniques for evolutionary clustering*, Proceedings of the Symposium "Zilele Academice Clujene", (2004), pp. 75-83.

[11] Dumitrescu D., Ferenc Jrai-Szab, Simon K., *Link-Cell method for evolutionary multi-modal optimization. Application in dynamic evolutionary clustering*, Carpathian Journal of Mathematics, 20 (2004), pp. 177-186

[12] Schreiber T., *A Voronoi diagram based adaptive k-means type clustering algorithm for multidimensional weighted data*, Universitat Kaiserslautern, Technical Report, (1989).

[13] Selim S. Z., Ismail M. A., *k-means type algorithms: a generalized convergence theorem and characterization of local optimality*, IEEE Tran. Pattern Anal. Mach. Intelligence, PAMI-6, 1 (1986), pp. 81-87.

[14] Wen X, Fuhrman S, Michaels GS, Carr GS, Smith DB, Barker JL, Somogyi R. Large scale temporal gene expression mapping of central nervous system development. Proc of The National Academy of Science USA., 95 (1998), 334339.

[1] Babes-Bolyai University, Faculty of Mathematics and Computer Science, Computer Science Department

E-mail address: ddumitr@cs.ubbcluj.ro

[2] Babes-Bolyai University, Faculty of Mathematics and Computer Science, Computer Science Department

E-mail address: ksimon@cs.ubbcluj.ro

[3] Babes-Bolyai University, Faculty of Mathematics and Computer Science, Computer Science Department

E-mail address: vig_nora@yahoo.com

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 153–160

GENETIC CHROMODYNAMICS FOR THE JOB SHOP SCHEDULING PROBLEM

D. DUMITRESCU[(1)], CATALIN STOEAN [(2)], AND RUXANDRA STOEAN [(2)]

Abstract. A novel evolutionary computing approach to the job shop scheduling problem is proposed. The new technique is based on a recent metaheuristic which, through its nature and mechanisms, manages to maintain the diversity in the population, fact that conducts to the discovery of several local optima and, thus, avoids the blockage of the entire population into a local optima region, which represents a major concern when designing an algorithm for scheduling. Furthermore, the aim of present paper is to find multiple optimal schedules within one problem and thus bring several options to its managers. The newly evolutionary approach to the job shop scheduling problem is validated on an instance, which is sufficient to be considered NP-hard. Results demonstrate the success of this first attempt and prove the promise of the new approach.

1. Introduction

Scheduling is a well-known problem that deals with the efficient allocation of resources with respect to time in order to perform a collection of tasks. Scheduling problems appear in many real life situations. In manufacturing, tasks correspond to parts that need to be processed on a set of machines. In hospitals, tasks are patients and resources are doctors, nurses, hospital beds or medical equipment. In education, tasks are classes and resources can be teachers, classrooms, and students. Finally, in transportation, problems include material transportation, airport terminal scheduling, train scheduling [7].

The generalized version of scheduling is the job shop scheduling problem (JSSP) and is a widely studied NP-hard problem. Various solutions have been brought to JSSP, among which evolutionary algorithms (EA) make a significant part.

In present paper, a new method based on a recently developed evolutionary metaheuristic, called genetic chromodynamics (GC) [2], is proposed as a solver of JSSP. The novel technique is very suitable for the given task due to its multimodal

2000 *Mathematics Subject Classification.* 68T20, 68T05.

Key words and phrases. job shop scheduling, evolutionary computation, multimodality, genetic chromodynamics.

©2007 Babeş-Bolyai University, Cluj-Napoca

nature that is both able to direct search to multiple optimal regions of the solutions space and to escape local optima.

Experiments are conducted on an 3×3 instance of JSSP and results validate the assumptions. However, this is only a first attempt with the new technique. Many of its components and corresponding values still remain to be improved. Also, validation on other examples has to be continued.

The paper is structured as follows. Section 2 gives the definition of JSSP, while section 3 briefly describes some evolutionary approaches to the problem. Section 4 presents the support evolutionary metaheuristic, whereas section 5 outlines the new approach to JSSP. Section 6 presents the experimental results and the paper ends with conclusions and ideas for future work.

2. Job Shop Scheduling Problem

The JSSP can be enunciated in the following manner. Suppose there are m manufacturing machines, M_i, $i = 1, 2, ..., m$, and n jobs to be performed, J_k, $k = 1, 2, ..., n$. Each job has an ordering of m tasks of specified duration, T_{ki}, $k = 1, 2, ..., n$, $i = 1, 2, ..., m$ and each task must be performed using the corresponding machine. Each job visits each machine exactly once. The tasks of each job have to be performed in the specified order only. A machine can perform only one task at a time and cannot be interrupted. The task is to make such a *schedule* that minimizes the time that is required to process all jobs. An example of a JSSP is outlined in the beginning of section 6.

3. Evolutionary Computing Approaches to Job Shop Scheduling Problem

The crucial issue in any technique that addresses JSSP is represented by the blockage into local optima. Consequently, some mechanism to handle this problem has to be implemented in any such approach. In addition, the ability of a heuristic to find a means to discover multiple optimal schedules would constitute an advantage.

Apart from deterministic techniques ranging from the classical backtracking to simulated annealing or tabu search, several heuristics and metaheuristics from the field of EAs (which are half deterministic, half probabilistic algorithms) have been proposed to solve JSSP.

An EA encodes the space of candidate solutions to the problem to be solved into a population of *individuals*. The values for the *genes* of the individuals in the initial population are randomly chosen. Subsequently, through the means of a fitness function that measures the quality of individuals (and thus the quality of solutions) and through the use of selection for reproduction and variation operators (crossover and mutation), candidate solutions evolve and, in the end, reach the

optimum. The optimum can be considered as the best individual from the last generation or the best individual from all generations [4], [5].

We will enumerate only a number of the evolutionary attempts to JSSP. One approach was from the travelling salesman problem point of view [10]. In a different view, a simple representation for individuals is chosen and it is interpreted by a schedule builder and specialized variation operators are employed [10]. Other attempts prefer to incorporate problem specific knowledge both into operators and representation [1], [6]. In other methods, a disjunctive graph model is used where each arc is binary labelled to define ordering and thus a schedule can be represented by a binary string and evolved by an EA with corresponding representation. A recent algorithm uses a representation which is much simpler from an EA point of view [11]. As it is very often the case that a machine has to choose among many jobs that are in queue, an individual encodes the machines and, for each of them, a vote which points for the job in the line that wins the right to use the machine. The JSSP is then solved by evolving genetic algorithms by a particular genetic programming method.

4. Genetic Chromodynamics Metaheuristic

Generally speaking, if, for some problem, multiple optimal solutions are possible then a multimodal EA may be employed in order to get the whole picture of possibilities before making a decision. Moreover, multimodal evolutionary heuristics can be used even for unimodal tasks as they have the ability to escape local optima.

The novel GC [2] has demonstrated its suitability with respect to problems exhibiting issues discussed above through the numerous results obtained by its application to different types of problems like function optimization, clustering or classification [2], [3], [8], [9].

GC represents a multimodal evolutionary metaheuristics based on radii. Only individuals that are similar under a given radius can recombine. Moreover, individuals that resemble each other under another specified radius are merged.

Algorithm 1 Merging procedure within GC

repeat
  A chromosome c is considered to be the current one;
  Select all individuals in the merging region of c, including itself;
  Remove all but the best chromosome from the selection;
until the merging region of each chromosome remains empty

Depending on the encoding for the individuals, some distance between them has to be defined (Hamming for binary encoding, Euclidean or Manhattan or any other distance for real encoding). Within one generation, distance is computed

twice - once when the selection for mating is done and once when the merging process takes place.

Evolution in GC takes place as in Algorithm 2. The initial population is randomly generated. Each individual is then taken into account for forming the new generation. For each such stepping stone, all individuals in the mating region are found and one of them is selected for recombination by means of proportional selection. Recombination takes place and competition for survival of the fittest is held between the offspring and the first parent only. If there are no individuals in the mating region of the current individual then the current individual suffers mutation. Obtained mutated offspring is inserted in the population and takes the place of its parent only if it is better than the current individual. Before proceeding to the next generation, the merging procedure is applied, so that unfit individuals are removed from the population (Algorithm 1).

Algorithm 2 GC Algorithm

```
Initialize population;
while termination condition is not satisfied do
  Evaluate each chromosome;
  for all chromosomes c in the population do
    if mating region of c is empty then
      Apply mutation to c;
      if obtained chromosome is fitter than c then
        Replace c;
      end if
    else
      Select one chromosome from the mating region of c for crossover;
      Obtain and evaluate one offspring;
      if offspring is fitter than c then
        Replace c;
      end if
    end if
  end for
  Merging
end while
```

As a consequence of the GC mechanisms, subpopulations appear and, with each iteration, they become more and more separated. Depending on the choice of values for the two GC radii, sooner or later each subpopulation contains only one individual that suffers only mutation, leading thus to refinements in the final solutions which are, in the end of the algorithm, each connected to an optimum point in the search space.

5. Genetic Chromodynamics for Job Shop Scheduling Problem (GCJSSP)

The new evolutionary technique for the JSSP acts in the following manner. The multiple optimal possible schedules are encoded into EA individuals in an adapted manner of [11] and evolved through GC.

In what follows, the choice for the evolutionary components with respect to JSSP is outlined.

5.1. **Representation of Individuals.** Each individual in the population contains a number of $m \times n$ genes. The i-th group made of n genes corresponds to machine M_i , $i = 1, 2, ..., m$, and the genes of a group encode votes for jobs that are in queue for M_i. This means that, in case there are multiple jobs - say for instance J_p and J_q - that both start with the task that has to be performed by M_i only, the choice between them is taken by selecting the job corresponding to the maximum value between c_{ip} and c_{iq}.

An individual thus has the form:

$$c = (c_{11}c_{12}...c_{1n}|c_{21}c_{22}...c_{2n}|...|c_{m1}c_{m2}...c_{mn}) \tag{1}$$

Consequently, an individual encodes a schedule, with the succession of jobs given by votes, as will be illustrated by an example in the following subsection. Initially, values for genes are randomly generated using a uniform distribution in the interval [0, 1].

5.2. **Fitness Function.** The quality of an individual c is associated with the total time it takes for all jobs J_k, $k = 1, 2, ..., n$, to be performed using the schedule provided by c and the given duration of each task. As the fitness of individuals corresponds to the time necessary for all jobs to be completed, the task for GCJSSP is to minimize evaluations; we deal therefore with a minimization problem.

In order to outline the way in which quality is computed, we consider a JSSP instance (Figure 1 (a)) and we carry out the evaluation of a certain randomly generated individual $c = (0.5, 0.7, 0.9|0.2, 0.8, 0.3|0.4, 0.9, 0.6)$. There are three jobs which each consist of three tasks to be performed by three corresponding machines. The duration of each task may be found enclosed in parentheses.

Initially, only machines M_1 and M_2 can start working as there is no task for M_3 in the beginning of any job. On the other hand, M_1 may handle either J_1 or J_2; the values corresponding to M_1 in c are 0.5, 0.7 and 0.9 respectively. As M_1 has to choose between J_1 and J_2, J_2 is the job selected first as the second value, 0.7, is greater than 0.5. At the same time, M_2 starts the only job it has in line, which is J_3(Figure 1 (a), (b)).

Machine M_1 finishes the task first (after 2 units of time, e.g. minutes) and is next free to start its task in another job, while machine M_2 still has one minute to

go until its first task for job J_3 is done. Until now, the total time spent measures 2 minutes.

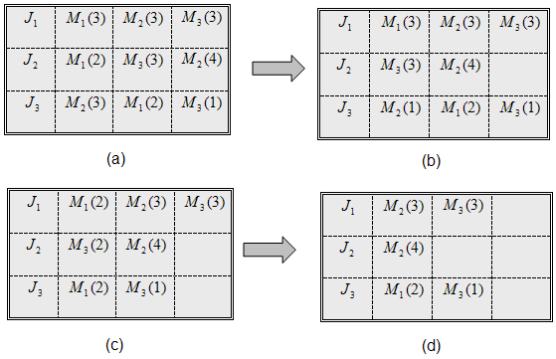


FIGURE 1. An illustration of fitness computation within GCJSSP

At the current moment, M_1 and M_3 are ready to begin another job. Naturally, M_1 selects job J_1 and M_3 starts the task of J_2. After 1 minute, M_2 finishes its task and has to wait as the current tasks of the other jobs J_1 and J_2 are still undertaken by the other two machines for the next 2 minutes.

Figure 1 (c) shows M_1 and M_3 executing their current tasks, while M_2 waits. At the step illustrated in Figure 1 (d), all three machines are available. M_1 naturally starts job J_3, while M_2 has to pick between J_1 and J_2. The latter is selected as its corresponding value in the individual's representation, 0.8, is higher than 0.2. Total time reached 5 minutes.

After two more minutes, M_1 finishes and M_3 starts the last task of J_3. Machine M_3 finishes the final task of J_3 before it ends J_2's last task. After one more minute, M_2 finally selects job J_1 and only after this task is completed, M_3 begins its last task. All jobs are eventually completed. The total time reaches 15 minutes and represents the quality of individual c.

5.3. **Variation operators.** Intermediate crossover and mutation with normal perturbation [4], [5] were experimentally considered for reproduction.

5.4. **Stop condition.** The evolutionary process stops after a predefined number of generations. The last population gives the multiple optimal individuals that are decoded into the corresponding schedules.

6. Experimental Results

An experimental environment was set up and results of GCJSSP were collected. The choices of a specific problem, values for parameters of GC and obtained results are further on presented.

TABLE 1. Manual choice of the values for GCJSSP parameters

Parameters	Values
Initial population size	50
Number of generations	100
Mating radius	0.1
Merging radius	0.15
Mutation probability	0.4
Mutation strength	0.15

6.1. **JSSP instance.** The new approach was validated on the JSSP 3×3 problem presented above [12]. The representation encodes the routing of each job, J_k, $k = 1, 2, ..., n$ ($n = 3$ here), through each machine, M_i, $i = 1, 2, ..., m$ ($m = 3$ in this case), and the processing time for each task, T_{ki}, $k = 1, 2, ..., n$, $i = 1, 2, ..., m$, which is written in parentheses. 3×3 JSSP instances have been demonstrated to be already NP-hard.

6.2. **Experimental Setup.** The values for the GCJSSP parameters are depicted in Table 1. Note that, in order to find these values, manual tuning was performed.

6.3. **Results.** GCJSSP was applied to the problem instance for 30 runs. In each run, results gave multiple possible configurations for schedules of total time equal to 12.0 (which is in fact the optimal time for the considered problem stated in [12]). One obtained individual (schedule) is given in Figure 2.

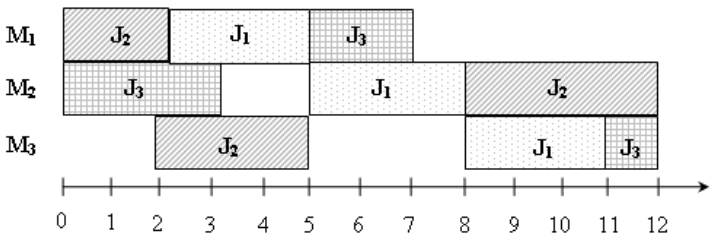


FIGURE 2. Representation of a solution to the chosen 3×3 problem reached by GCJSSP

7. Conclusions and future work

Present paper addresses a novel evolutionary approach to the job shop scheduling problem. The technique is based on a recent powerful metaheuristic that is

able to cope with the local optima issues and, at the same time, to find multiple optimal configurations of schedules.

Experiments are conducted on an instance of JSSP, whose dimensionality suffices to say the problem is NP-hard. Results demonstrate the suitability of the new approach. Nevertheless, proposed technique is here at its first attempt, thus other choices for parameters and their values together with higher dimensional instances of JSSP have to be investigated in the near future.

References

[1] Bagchi, S., Uckun, S., Miyabe, Y. and Kawamura, K., "Exploring Problem-Specific Recombination Operators for Job Shop Scheduling", Proceedings of the 4th International Conference on Genetic Algorithms, 1991, pp. 10-17.

[2] Dumitrescu, D., "Genetic Chromodynamics", Studia Universitatis Babes-Bolyai, Informatica, 2000, pp. 39 - 50.

[3] Dumitrescu, D., Gorunescu, R., (2004), "Evolutionary clustering using adaptive prototypes", Studia Univ. Babes - Bolyai, Informatica, Volume XL IX, Number 1, 2004, pp. 15 - 20.

[4] Dumitrescu, D., Lazzerini, B., Jain, L., C., Dumitrescu, A., *Evolutionary Computation*, CRC Press, Boca Raton, Florida, 2000

[5] Eiben, A. E., Smith J. E., *Introduction to Evolutionary Computing*, Springer-Verlag Berlin Heidelberg, 2003

[6] Husbands, P., Mill, F., Warrington, S., "Genetic Algorithms, Production Plan Optimization and Scheduling", Proceedings of Parallel Problem Solving from Nature I, 1991, pp. 80-84.

[7] Sadeh, N., *Look-ahead techniques for micro-opportunistic job shop scheduling*, PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1991.

[8] Stoean, C., Preuss, M., Gorunescu, R., Dumitrescu, D., "Elitist Generational Genetic Chromodynamics - a New Radii-Based Evolutionary Algorithm for Multimodal Optimization", Proceedings of the 2005 IEEE Congress on Evolutionary Computation - CEC 2005, Edinburgh, UK, September 2-5, 2005, pp. 1839 - 1846.

[9] Stoean, C., Gorunescu, R., Preuss, M., Dumitrescu, D., "An Evolutionary Learning Classifier System Applied to Text Categorization", Annals of University of Timisoara, Mathematics and Computer Science Series, vol. XLII, special issue 1, 2004, pp. 265-278.

[10] Syswerda, G., "Schedule Optimization Using Genetic Algorithms", Handbook of Genetic Algorithms, Davis, L. (Ed), 1991, pp. 332-349.

[11] Werner, J., *Evolving Genetic Algorithm for Job Shop Scheduling Problems*, Technical Report, School of Computing, Information Systems and Mathematics, South Bank University, 2000

[12] Yamada T., Nakano R., "Genetic Algorithms for Job-Shop Scheduling Problems", Proceedings of Modern Heuristic for Decision Support, Unicom seminar, London, 1997, pp. 67-81.

[1] Babes - Bolyai University, Faculty of Mathematics and Computer Science, Department of Computer Science, Str. M. Kogalniceanu, No. 1, Cluj-Napoca, 400084, Romania

E-mail address: ddumitr@cs.ubbcluj.ro

[2] University of Craiova, Faculty of Mathematics and Computer Science, Department of Computer Science, Str. Al. I. Cuza, No. 13, Craiova, 200585, Romania

E-mail address: {catalin.stoean, ruxandra.stoean}@inf.ucv.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 161–168

EXACT MODEL BUILDING IN HIERARCHICAL COMPLEX SYSTEMS

DAVID ICLĂNZAN[1] AND DAN DUMITRESCU[2]

Abstract. The paper proposes a novel trajectory based method which optimizes problems via explicit decomposition. The method is able to learn and deliver the problem structure in a comprehensible form to human researchers.

1. Introduction

Complex systems are characterized by multiple interactions between many different components where local and global phenomena interact in complicated, often nonlinear ways [6]. Researchers from different areas confront with them on a daily basis. To successfully address large-scale problems in complex systems, a proper problem decomposition must be employed.

A special type of complex systems are the hierarchical ones, where the system is composed from subsystems, each of which is hierarchical by itself [7]. Many systems around as are hierarchical. The successive levels used in physics are familiar to everyone: materials are composed of molecules, molecules are composed of atoms, atoms are composed electrons, protons, neutrons and so forth.

Hierarchical problems derive from hierarchical complex systems. Their efficient solving requires proper problem decomposition and assembly of solution from subsolution with strong non-linear interdependencies.

Pelikan and Goldberg [5] proposed the Hierarchical Bayesian Optimization Algorithm (hBOA), one of the few methods which is able to optimize problems with random linkages. hBOA can optimize problems which are not fully decomposable in one single level by hierarchical decomposition. Nevertheless, the decomposition information is implicitly stored in a Bayesian network which do not reveal the problem structure in a comprehensible form.

In a very recent development Yu and Goldberg [12] proposed an explicit hierarchical decomposition scheme for GAs. The method uses dependency structure matrix clustering techniques for linkage detection and it is able to approximately

2000 *Mathematics Subject Classification.* 68T20, 49M27, 91E40.
Key words and phrases. Problem Decomposition, Linkage Learning, Model Building.

©2007 Babeş-Bolyai University, Cluj-Napoca

capture the underlying problem structure. The inaccuracies are mainly caused by the intrinsic probabilistic nature of the algorithm.

In this paper we propose the Building Block Wise Greedy Search Algorithm (BBWGSA), a more systematic approach which can exactly capture the problems anatomy. The proposed method operates on a single point -instead of populations- and uses an explicit hierarchical decomposition scheme based on a greedy search operator which performs local search among competing building blocks (BBs) neighborhood. The search experience accumulated is used to reveal linkages and to update the BB structure.

The rest of the paper is organized as follows. Section two revisits and formally presents hierarchical problems. The third section describes the proposed method. Experiments and results are presented in section four. The paper is concluded with discussion and some outlines for future work in chapter five.

2. Hierarchical Difficulty and Hierarchical Problems

Although having a gross-scale BB structure, hierarchical problems are hard to solve without proper problem decomposition as the blocks from these functions are not separable.

The fundamental of hierarchically decomposable problems is that there is always more than one way to solve a (sub-) problem [9] leading to the separation of BBs "fitness" i.e. contribution to the objective function, from their meaning. This conceptual separation induces the non-linear dependencies between BBs: providing the same objective function contribution, a BB might be completely suited for one context whilst completely wrong for another one.

Hierarchical problems are very hard for mutation based hill-climbers as they exhibit a fractal like structure in the Hamming space with many local optima [11]. This bit-wise landscape is fully deceptive; the better is a local optimum the further away is from the global ones. At the same time the problem can be solved quite easily in the BB or "crossover space", where the block-wise landscape is fully non deceptive [9]. The forming of higher order BBs from lower level ones reduces the problem dimensionality.

If a proper niching is applied and the promising sub-solutions are kept until the method advances to upper levels where a correct decision can be made, the hierarchical difficulty can be overcome.

2.1. **Design of hierarchical problems.**

In this paper two hierarchical test functions are considered: the hierarchical IFF [9] and the hierarchical XOR [10]. These problems are defined on binary strings of the form $\{0,1\}^{k^p}$ where k is the number of sub-blocks in a block, and p is the number of hierarchical levels. The meaning of sub-blocks is separated from their fitness by the means of a boolean function h which determines if the sub-block is valid in the current context or not. In

the shuffled version of these problems the tight linkage is disrupted by randomly reordering the bits. The functions are detailed as follows.

2.1.1. *Hierarchical if and only if (hIFF).* The hIFF has $k = 2$ and it is provided by the if and only if relation, or equality. Let $L = x_1, x_2, \ldots, x_{2^{p-1}}$ be the first half of the binary string x and $R = x_{2^{p-1}+1}, x_{2^{p-1}+2}, \ldots, x_{2^p}$ the second one. Then h is defined as:

$$h_{iff}(x) = \begin{cases} 1 & \text{, if } p = 0; \\ 1 & \text{, if } h_{iff}(L) = h_{iff}(R) \text{ and } L = R; \\ 0 & \text{, otherwise.} \end{cases} \tag{1}$$

Based on h_{iff} the hierarchical iff is defined recursively:

$$H_{iff}(x) = H_{iff}(L) + H_{iff}(R) + \begin{cases} length(x) & \text{, if } h_{iff}(x) = 1; \\ 0 & \text{, otherwise.} \end{cases} \tag{2}$$

At each level $p > 0$ the $H_{iff}(x)$ function rewards a block x if and only if the interpretation of the two composing sub-blocks are both either 0 or 1. Otherwise the contribution is zero.

hIFF has two global optima: strings formed only by 0's or only by 1's. At the lowest level the problem has $2^{l/2}$ local optima where l is the problem size.

2.1.2. *Hierarchical exclusive or (hXOR).* The global optima of hIFF are formed by all 1's or all 0's which may ease the task of some methods biased to a particular allele value. To prevent the exploitation of this particular problem property the hXOR was designed.

The definition of hXOR is analogous with the hIFF, having only a modification in the validation function h, where instead of equality we do a complement check.

$$h_{xor}(x) = \begin{cases} 1 & \text{, if } p = 0; \\ 1 & \text{, if } h_{xor}(L) = h_{xor}(R) \text{ and } L = \bar{R}; \\ 0 & \text{, otherwise.} \end{cases} \tag{3}$$

The $\bar{R}$ stands for the bitwise negation of R.

3. Systematic Exploitation of the Building Block Structure

As already indicated in Section 2, hierarchical problems are fully deceptive in Hamming space and fully non deceptive in the BB space. The problem representation together with the neighborhood structure defines the search landscape. With an appropriate neighborhood structure, which operates on BBs, the search problem can be transferred from Hamming space to a very nice, fully non deceptive search landscape which should be easy to systematically exploit (ex. hill-climb).

Usually hill-climbers described in the literature use bit-flipping for replacing the current state [1, 2, 3]. This implies a neighborhood structure which contains strings that are relatively close in Hamming distance to the original state, making those

methods unsuited for solving hierarchical problems, where local optima and global optima are distant in Hamming space. But the neighborhood can be defined as an arbitrary function which assign to a valid state s a set of valid states $N(s)$. The main idea of the paper is to build a trajectory method which takes into account the BB structure of the problems and defines its neighborhood structure accordingly.

3.1. **Adaptation of the neighborhood structure.** Theoretical studies denote that a GA that uses crossover which does not disrupts the building block structure holds many advantages over simple GA [8].

Similarly, in order to be able to efficiently exploit the BB landscape, the proposed method must learn the problem structure and evolve the solution representation to reflect the current BB knowledge. The changing of representation implies the adaptation of the neighborhood structure which is the key to conquer hierarchical problems: by exploring the neighborhood of the current BB configuration the next level of BB can be detected.

In order to be able to identify linkages we enhance our method with a memory where hill-climbing results are stored. Evolutionary Algorithms with linkage learning mechanism extracts the BB information from the population. Similar techniques can be applied to devise BB structures from the experience stored in the memory. However, the solutions stored by the proposed method offer an important advantage over populations: they are noise free. While individuals from populations may have parts where good schemata's have not yet been expressed or have BBs slightly altered by mutation, the solution stored in memory are always exact local optima. In the case of fully non deceptive BB landscapes the systematic exploration of BB configurations guarantees that in the close neighborhood of these states there are no better solutions.

3.2. **Building Block Wise Greedy Search Algorithm.** The proposed method involves three main steps: (i) hill-climbing the search space according to a BB neighborhood structure; (ii) local optima obtained in (i) are used to detect linkages and extract BB information; (iii) the BB configuration and implicitly the neighborhood structure are updated.

3.2.1. *The Greedy Search Algorithm.* BB hill-climbing is rather straightforward: instead of flipping bits, the search focuses on the best local BB configuration. Each BB is processed systematically by testing its configurations and selecting the one which provides the highest (or lowest in the case of minimization) objective function value. While the best configuration of a particular BB is searched, the configurations of the other BBs are hold still.

The individual is represented as a sequence of BBs: $s = (b_1, b_2, \ldots, b_n)$ where n is the number of BBs. Each BB b_i can represent multiple configurations: $V_i = \{v | v \in \{0,1\}^l\}$ where l is the length of b_i. This allow the sustenance and parallel processing of competing schemata.

1. Choose randomly a building block b_i from s which has not yet been clustered;
2. Let L be the set of building blocks whose configuration from the memory are mapped bijectively to b_i;
3. If L is empty update the possible configurations V_i to the configurations encountered in the memory;
4. If L is not empty form a new building block $new\_b = b_i \bigcup L$ by setting the loci it's define to the union of loci from b_i and the building blocks from L. Also set the possible values V_{new_b} to all distinct configuration encountered, on the position defined by the new_b , operating on the binary representation of states from the memory;
5. Set $b_i \bigcup L$ as clustered;
6. If there exists building blocks which have not been clustered *goto 1*;

FIGURE 1. The linkage detection and new building block forming method.

3.2.2. *Linkage Detection.* Several techniques for detecting gene dependency from a population have been already presented in the literature [4, 12] which could be also employed by the proposed method. But due to the fact that the hierarchical problems under study are fully non deceptive in the BB space, a very simple method for linkage detection is considered. This process is facilitated by the advantage of having noise free states stored in memory.

The clustering of loci in new BBs is done by searching for bijective mappings. For a given block b_i, all BBs b_j are linked if distinct configurations of b_i map to distinct configurations of b_j. The configurations of b_i that can be found in the memory represent the domain while the configurations of b_j from the memory are the codomain.

Due to the transitivity property of bijective mappings (functions) all relevant BBs are discovered simultaneously. The linkage detection algorithm is presented in Figure 1. Harder problems (exhibiting overlapping BB structure for example), may require a more sophisticated linkage learning method.

All BBs linked together by a bijective mapping will form a new BB which replaces the linked loci in the BB structure. The possible configurations of the new BBs are extracted from the binary representation of states from the memory. All distinct configurations from the positions defined by the composing BBs are taken into account. If a BB can not be linked with any other BB it keeps its original place and only its possible configurations are updated in the same manner as the new BBs.

Proposed model can be summarized by the algorithm presented in Figure 2.

```
1.  Generate a random state s from the current BB structure;
2.  BB cill-climb from s and store the result in memory;
3.  If the resulted state is better then the best states seen so
far, keep the new state;
4.  If the memory is not filled up goto 1;
5.  Learn linkage from memory and update the BB configuration
according to the detected linkages;
6.  Empty memory;
7.  If termination condition not met goto 1;
```

FIGURE 2. Outline of the hill-climbing enhanced with memory and linkage learning. In steps 1-4 we accumulate the search experience (phase 1) which is exploited in steps 5-7 (phase 2).

4. RESULTS

Enhanced with linkage learning mechanism and variable neighborhood structure the BBWGSA should be able to efficiently solve relevant problems by hierarchical decomposition. Also due to the more systematic approach of the BBWGSA we expect it to deliver uncorrupted problem structure.

We tested these hypothesis on the 128-bit, 256-bit shuffled hIFF and hXOR problems. The memory size was set to 30 on the case of the 128-bit version respectively to 40 on the 256-bit one. 25 independent runs were performed on both test suits. The BBWGSA was able to find one of the global optima in all cases. More important it detected the perfect problem structure (complete binary tree) in all runs! In Figure 3 we depict the performance of the DSGMA++ on a small test suit as reported in [12]. Albeit the problem structure is quite well approximated it contains some inaccuracies which on bigger problem instances may be problematic.

Similarly to other methods like the DSMGA++, the BBWGSA uses explicit chunking mechanism enabling the method to deliver the problem structure. While DSGMA++ and other stochastic methods have to fight the sampling errors which sometimes induce imperfections, the BBWGSA was able to detect the perfect problem structure in all runs, due to its more systematic and deterministic approach. The enhanced capability of BBWGSA to capture the problem structure is also revealed by the fact that hIFF and hXOR are solved approximately in the same number of steps as their underlying BB structures (complete binary tree) coincide. However for the DSGMA++ the time needed to optimize the two problems differs significantly, being $O(l^{1.84}\log(l)$ for the hIFF and $O(l^{1.96}\log(l)$ on hXOR.

On hIFF and hXOR the multiple runs of the BBWGSA showed an unbiased behavior, finding in almost half-half proportion both global optima.

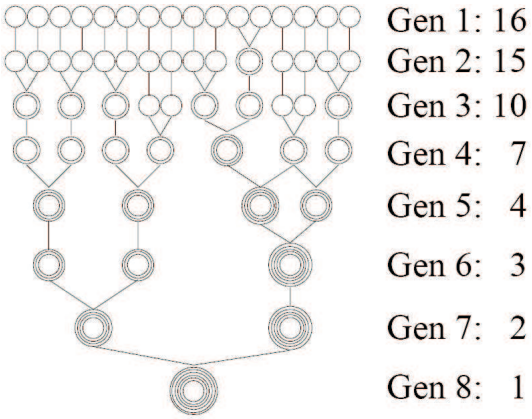


FIGURE 3. Example of problem structure obtained by the DSMGA++ for the hXOR function with 4 levels (chromosome length= $2^4 = 16$) as reported in [12]. The nodes represents genes of chromosomes. The number of circles represents the compression level. The descriptions on the right show the generation number and the chromosome length. The ideal case would be a complete binary tree with 4 layers.

A final remark concerns the stability of BBWGSA: the highest standard deviation encountered is 196 while other methods deal with standard deviations of much higher magnitude on the same test suites.

5. Conclusions

The Building Block Wise Greedy Search Algorithm (BBWGSA), a generic method for solving problems via hierarchical decomposition is proposed. The BBWGSA operates in the BB space where it combines BBs in a systematic and exhaustive manner. Exploration experience is used to learn the underlying BB structure of the search space expressed by linkages.

A very important aspect of the proposed method is that similar to DSMGA++, BBWGSA delivers the problem structure in a form comprehensible to humans. Gaining knowledge about the hidden, complex problem structure can be very useful in many real-world applications. Nevertheless, by adopting a more systematic approach, the proposed method was able to detect the perfect problem structure in all runs.

In the future we would like to enhance our method with more powerful linkage learning techniques in order to be able to tackle more difficult problem structures.

6. Acknowledgments

This work was supported by the CNCSIS, AT-70 / 2006 grant and the Sapientia Institute for Research Programs (KPI).

References

[1] S. Forrest and M. Mitchell. What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *MACHLEARN: Machine Learning*, 13, 1993.

[2] M. Mitchell, S. Forrest, and J. H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In F. J. Varela and P. Bourgine, editors, *Proc. of the First European Conference on Artificial Life*, pages 245–254, Cambridge, MA, 1992. MIT Press.

[3] H. Mühlenbein. How genetic algorithms really work: I. mutation and hillclimbing. In R. Männer and B. Manderick, editors, *Proceedings of the Second Conference on Parallel Problem Solving from Nature (PPSN II)*, pages 15–25, Amsterdam, 1992. North-Holland.

[4] M. Pelikan. *Bayesian optimization algorithm: from single level to hierarchy*. PhD thesis, 2002. Adviser-David E. Goldberg.

[5] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 511–518, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.

[6] D. Rind. Complexity and climate. *Science*, 284(5411):105–107, April 1999.

[7] H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, Massachusetts, first edition, 1969.

[8] D. Thierens and D. E. Goldberg. Mixing in genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 38–47, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[9] R. A. Watson, G. S. Hornby, and J. B. Pollack. Modeling building-block interdependency. *Lecture Notes in Computer Science*, 1498:97–108, 1998.

[10] R. A. Watson and J. B. Pollack. Hierarchically consistent test problems for genetic algorithms: Summary and additional results. In S. Brave and A. S. Wu, editors, *Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference*, pages 292–297, Orlando, Florida, USA, 13 July 1999.

[11] R. A. Watson and J. B. Pollack. Symbiotic composition and evolvability. In J. Kelemen and P. Sosik, editors, *Advances in Artificial Life, 6th European Conf., (ECAL 2001)*, pages 480–490, Berlin, 2001. Springer.

[12] T.-L. Yu and D. E. Goldberg. Conquering hierarchical difficulty by explicit chunking: substructural chromosome compression. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1385–1392, New York, NY, USA, 2006. ACM Press.

[1] Babeș-Bolyai University, Faculty of Mathematics and Computer Science, Str. Mihail Kogălniceanu nr. 1, 400084, Cluj-Napoca, România
E-mail address: david.iclanzan@gmail.com

[2] Babeș-Bolyai University, Faculty of Mathematics and Computer Science, Str. Mihail Kogălniceanu nr. 1, 400084, Cluj-Napoca, România
E-mail address: ddumitr@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 169–176

GENETIC PROGRAMMING WITH HISTOGRAMS FOR HANDWRITTEN RECOGNITION

OANA MUNTEAN[(1)]

Abstract. Handwritten recognition is a popular problem which requires special Artificial Intelligent techniques for solving it. In this paper we use Genetic Programming (GP) for addressing the off-line variant of the handwritten digit recognition. We propose a new type of input representation for GP: histograms. This kind of representation is very simple and can be adapted very easily for the GP requirements. Several numerical experiments with GP are performed by using several large datasets taken from the well-known MNIST benchmarking set. Numerical experiments show that GP performs very well for the considered test problems.

1. Introduction

Handwriting recognition concerns the conversion of the analog signal of handwriting into a digital symbolic representation. The analog signal can be in the form of either a two-dimensional scanned image of paper or a temporal one-dimensional signal captured from a device such as a tablet or PDA.

It is widely accepted that a perfect recognition of digits requires intelligence. This is why Artificial Intelligence technique have been intensively used for solving this problem. Among them, Artificial Neural Networks are the most popular.

In the recent years several techniques inspired from nature have emerged as good alternatives to the standard mathematical methods. One of them is Genetic Programming (GP) [6] whose aim is to generate computer programs based on a high level description of the problem to be solved.

In this paper we use GP for handwritten digit recognition. The novelty consists in the representation of the input. Because the matrices, encoding the images involved in the numerical experiments, are too big to be used as direct input for GP we have to extract and use only some partial information. This is why we have constructed the horizontal and vertical histograms [3] and this information was used as input for GP.

2000 *Mathematics Subject Classification.* 68T10, 68T20.
Key words and phrases. Genetic Programming, Handwritten recognition, Classification.

©2007 Babeş-Bolyai University, Cluj-Napoca

We have performed several numerical experiments by using a well-known benchmarking data set: MNIST [9]. Results have shown a more than 90% classification accuracy for most of the cases. For some tests the accuracy was more than 99%.

The paper is organized as follows: GP technique is briefly surveyed in Section 2. The handwritten digit recognition problem is defined in Section 3. Related work in the field of handwritten recognition is briefly reviewed in Section 4. The proposed approach is deeply presented in Section 5. Test problems are given in Section 6. The results of numerical experiments are given in Section 7. Strengths and limitations of the proposed technique are discussed in Section 8. Conclusions and future work directions are given in Section 9.

2. Genetic Programming

Genetic Programming (GP) technique provides a framework for automatically creating a working computer program from a high-level problem statement of the problem [6, 7]. Genetic programming achieves this goal by genetically breeding a population of computer programs using the principles of Darwinian natural selection and biologically inspired operations.

Five major preparatory steps [6] must be specified in order to apply a GP technique to a particular problem:

(1) the set T of terminals (e.g., the independent variables of the problem, zero-argument functions and random constants)
(2) the set F of primitive functions,
(3) the fitness measure (for explicitly or implicitly measuring the quality of individuals in the population),
(4) certain parameters for controlling the run
(5) the termination criterion and the method for designating the result of the run.

These preparatory steps are problem dependent so they must be specified, by a human user, for each particular problem.

3. Handwritten recognition problem

The recognition of handwritten characters by a machine is a very tough task. It takes years to come to a commercially valuable result in this area - years of research, years of development and years of turning theory into software that would do at least a small piece of an evident thing: turning human handwritten digital curves into letters and words. This work involves identifying a correspondence between the pixels of the image representing the sample to be analyzed (recognized) and the abstract definition of that character.

However, a lot of work has been done in this field lately and this happened due to the fact that it has an important role in many applications such as bank checks and tax forms reading, postal addresses interpretation, etc.

Another important thing to be considered about handwritten recognition methodologies is the manner in which the data is collected. There are two methods for accomplish this: on-line [11, 13] or off-line [10]. On-line handwritten data is collected using a digitizer or an instrumented pen to capture the pen-tip position $(x_t,\ y_t)$ as a function of time. Contrary to on-line, off-line handwritten data is collected using a scanner resulting in the generation of the signal as an image.

This paper presents an off-line implementation of the problem and focus on digits handwritten recognition; however it is possible to expand to full-character recognition.

4. Related work

A lot of work has been done in the field of handwritten recognition. The main drawback of this problem is the extremely variability of handwritten that produces a large number of features for different writing style. That is why there is no single method for solving handwritten recognition.

Artificial Neural Networks [5] are very popular techniques for solving this problem [1, 2, 11]. Although, GP [6] was suggested [12, 14] as an alternative method for attacking this problem.

In this section we consider the current state of research, the developments in the handwriting recognition field and present some related research work using GP algorithms. The basic idea of these algorithms is the classification of data in different classes, starting from different inputs and different way of representation.

One of the handwritten recognition approaches using GP uses Decision Trees to solve the problem [12]. Here two types of algorithms might be evolved: starting from bottom-up or top-down. The first one starts by dividing all digits into two different classes and applying the same pattern to both subtrees, while the second starts by classifying pairs of digits and continuing by a larger classification, i.e. from a binary classification, a classification with four classes can be performed.

Another GP approach for the problem focuses on the particular features of the characters using recursive cognition [14]. In this way the feature of an image containing a character is hierarchically divided into small sub-images and the process is applied recursively until an acceptance criteria is met (the character is recognized).

Similar to this, the fuzzy regional representation [10] is based on the features of the character, but here the image is divided into a grid from the beginning of the process. Each region of the grid is further included into a fuzzy vector, that will provide the particularities of the analyzed character.

5. Proposed approach

The proposed approach uses standard GP [6] as the main search mechanism.

What is different in our paper is the way in which the input is represented. GP ability to solve problems is highly related to the number of terminals [6]. If we have too many terminals it is more difficult for GP to select the good ones.

The original data sets consist in images represented as 20x20 matrices. If all pixels of this matrix are sent as input to GP we would have 400 terminals. This is a huge number for GP.

This is why we have tried to reduce the number of inputs. We can do that by extracting some information from the original 20x20 matrices. In this paper we have built the horizontal and vertical histograms and this information was actually sent as input to GP. This information is further processed by the GP classifier.

Histograms representation is very simple [3]. For each row and each column of the image we count the pixels containing ink (see Figure 1). The obtained number can give us some rough information about what digit we have there. Histograms have been used in the past for handwritten recognition [3]. However, this is the first time when they are used in conjunction with Genetic Programming.

FIGURE 1. Histograms for digit 2. For each row and column we have counted the ink pixels.

6. Test data

The MNIST is a database [9] of handwritten digits having a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST [4]. The digits have been size-normalized (to a 20x20 matrix) and centered in a 28x28 fixed-size image. In our case we have extracted only the information from the central 20x20 matrix. By building the histograms we have obtained a dataset with 60000 examples, each of them having 40 attributes.

7. Numerical experiments

First of all we wanted to find out if the method has enough power to distinguish between pairs of digits. For instance we are interested to find a mathematical formula that can be applied in order to distinguish between 0 and 1.

In this case we deal with a classification problem with 2 classes. The fitness is computed as the number of incorrectly classified examples over the total number of cases. Thus, the fitness has to be minimized.

For this purpose we have extracted from MNIST all 45 distinct pairs of digits. Secondly we have built the histograms and then we have applied GP for all of them.

Parameters of the GP method used in all numerical experiments are given in Table 1.

TABLE 1. General parameters of the GP algorithm

Parameter	**Value**
Population size	200
Number of generations	51
Mutation probability	0.1
Crossover probability	0.9
Selection	Binary Tournament
Terminal set	Problem inputs (40 as many)
Function set	$F = \{+, -, *, \%\}$
Maximum GP tree height	10

7.1. **Results.** The classification accuracy obtained by running GP method for all test problems is given in Table 2. Because the method works with pseudo-random numbers, we performed more runs (30 runs in fact) and we have averaged the results. In all runs we have obtained a very good classification error.

Taking into account the average values presented in Table 2 we can observe that the best error is obtained for classifying 6-7 digit pair. This means that is very easy to make distinction between digits 6 and 7. If we take into account the best values we can also see that the best classified pair of digits is again 6-7. The worst results are obtained for 3-5 pair and, then, for 5-8 pair. This means that is very difficult to make distinction between digits 3 and 5.

Note that the results can be further improved by using a larger population or running the search process for more generations.

7.2. **Comparison with other GP techniques.** In [14] GP with a special representation was used for digit classification. The reported error was between 2% and 5%. In [10] the error was between 3% and 9%.

In Table 3 we have compared our approach with another GP-based technique proposed in [8] on the same data set. The results show that our method outperformed the compared one in all cases.

Note that a perfect comparison cannot be made because the representations and the parameter settings are too different.

TABLE 2. The results (in percent) obtained by applying GP to the considered test problems. *Best/Worst* stands for the fitness of the best individual in the best/worst run. *Avg* and *StdDev* are the mean and the standard deviation of the quality for the best individual (in each run) over 30 runs. ds_{xy} means the dataset which contains only the images with digits x and y.

Data	*Best*	*Worst*	*Avg*	*StdDev*	*Data*	*Best*	*Worst*	*Avg*	*StdDev*
ds_{01}	3.32	5.29	4.36	0.61	ds_{29}	0.98	2.16	1.26	0.33
ds_{02}	2.89	6.16	4.04	0.88	ds_{34}	1.24	2.48	1.85	0.43
ds_{03}	3.73	5.21	4.44	0.50	ds_{35}	13.77	20.58	17.78	2.35
ds_{04}	0.84	1.43	1.12	0.19	ds_{36}	1.79	4.43	2.65	0.98
ds_{05}	5.89	14.28	8.33	2.60	ds_{37}	4.51	7.05	5.53	0.85
ds_{06}	1.63	3.02	2.08	0.43	ds_{38}	8.95	15.74	11.94	2.07
ds_{07}	1.18	2.57	1.83	0.43	ds_{39}	3.74	5.78	4.49	0.65
ds_{08}	5.23	6.85	5.91	0.50	ds_{45}	2.01	4.38	3.43	0.81
ds_{09}	1.06	2.15	1.61	0.38	ds_{46}	1.66	2.63	1.98	0.31
ds_{12}	4.52	9.42	7.42	2.17	ds_{47}	2.10	3.97	3.16	0.61
ds_{13}	7.13	13.09	9.77	2.07	ds_{48}	1.93	3.89	2.57	0.61
ds_{14}	1.03	6.32	2.12	1.68	ds_{49}	9.39	12.40	10.48	0.92
ds_{15}	5.91	10.95	8.63	1.76	ds_{56}	2.05	2.65	2.33	0.19
ds_{16}	2.18	4.90	2.71	0.79	ds_{57}	4.21	8.23	6.16	1.62
ds_{17}	1.30	6.30	2.09	1.50	ds_{58}	9.35	17.33	12.83	2.57
ds_{18}	11.10	14.80	13.06	0.97	ds_{59}	3.21	6.18	5.01	1.07
ds_{19}	1.85	3.21	2.52	0.51	ds_{67}	0.14	0.69	0.32	0.20
ds_{23}	6.86	11.20	8.23	1.23	ds_{68}	2.38	3.41	2.91	0.36
ds_{24}	1.48	2.37	1.77	0.23	ds_{69}	0.26	0.98	0.41	0.22
ds_{25}	9.00	11.58	10.21	0.91	ds_{78}	5.10	7.05	6.08	0.79
ds_{26}	6.27	9.88	7.82	1.20	ds_{79}	6.18	9.76	8.16	1.16
ds_{27}	1.36	3.58	1.97	0.64	ds_{89}	4.00	7.48	5.49	1.03
ds_{28}	3.02	5.81	4.29	0.87					

8. STRENGHTS AND WEAKNESSES

Genetic Programming strengths and weaknesses are already known to the research comunity. This is why we focus our attention to the advantages and disadvantages introduced by the representation with histograms.

The greatest benefit is the fact that we have been able to successfully apply GP for solving this problem. This is due to the reduced number of inputs which are sent to the GP individuals. Note that this was not possible if we send the entire

TABLE 3. Comparison between our approach and the one proposed in [8]. For each pair of digits we gave the errors obtained by both methods.

Pair	**Our approach**	**The approach proposed in** [8]
(1,7)	2.09	4.30
(2,7)	1.97	7.10
(3,8)	11.94	13.60
(4,9)	10.48	12.10
(8,9)	5.49	8.80

matrix to GP because the method cannot successfully handle such large amount of inputs.

The limitations of the proposed approach are mainly related to the limitations introduced by the histogram representation. There are several pairs of digits which are difficult to be distinguished when using this kind of representation. For instance the pairs 1-7, 3-8, 9-6 have very similar histograms and thus it is quite difficult to find a very good classification.

Luckily, our test data contains digits written by hand. In this case there is a more clear distinction between the digits from the previously enumerated pairs since the hand written characters have a huge number of possible representations. This fact has been shown by the results from Table 2 where the classification errors are quite good for pairs 1-7, 3-8 and 9-6.

Another weakness is that by using histograms we have reduced the amount of information which was sent to GP classifier. This could lead to some poor results in some cases.

9. Conclusions and further work

In this paper a new way of representing the input for solving handwritten recognition problems using GP has been suggested.

The proposed representation was tested on a well-known benchmarking data set. The results of the numerical experiments have shown very good classification accuracy.

Further efforts will be focused on the following directions:

- Testing the method for other difficult datasets (including characters).
- Reducing the size of the images. This will reduce the number of inputs for Genetic Programming.
- Using an extended set of functions for GP. This set may include the operators sin, exp, lg.

- Discovering classifiers which are able to make distinction between a particular digit and all other digits. This will increase the generalization ability of the method.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] Y. L. Cun and (et al). Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, pages 396–404. Morgan Kaufmann, 1990.

[3] H.-C. Fu, H. Y. Chang, Y. Y. Xu, and H. T. Pao. User adaptive handwriting recognition by self-growing probabilistic decision-based neural networks. *IEEE-NN*, 11(6):1373–1384, 2000.

[4] M. D. Garris and (et al). NIST form-based handprint recognition system. In *National Institute of Standards and Technology (NIST) Intelligent Systems Division*, 1994.

[5] M. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, 1995.

[6] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[7] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, 1994.

[8] K. Krawiec. Genetic programming using partial order of solutions for pattern recognition tasks. In E. Puchala, editor, *Proceedings of the second National Conference on Computer Recognition Systems KOSYR-2001*, pages 427–433, Strona palacu w Milkowie, Karpacza, Poland, 28-31 May 2001.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[10] A. Lemieux, C. Gagne, and M. Parizeau. Genetical engineering of handwriting representations. In *Frontiers in Handwriting Recognition*, pages 145–150, 2002.

[11] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell*, 22(1):63–84, 2000.

[12] T. Tanigawa and Q. Zhao. A study on efficient generation of decision trees using genetic programming. In D. W. (et al), editor, *Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 1047–1052, Las Vegas, Nevada, USA, 2000. Morgan Kaufmann.

[13] A. Teredesai and (et al). On-line digit recognition using off-line features. In S. C. (et al), editor, *ICVGIP 2002, Proceedings of the Third Indian Conference on Computer Vision, Graphics & Image Processing, 2002*. Allied Publishers Private Limited, 2002.

[14] A. Teredesai, J. Park, and V. Govindaraju. Active handwritten character recognition using genetic programming. In J. F. M. (et al), editor, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 371–379. Springer-Verlag, 2001.

[1] Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Kogălniceanu 1, Cluj-Napoca, 400084, Romania

E-mail address: oana_muntean85@yahoo.com

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 177–184

STIGMERGIC AGENT SYSTEMS FOR SOLVING NP-HARD PROBLEMS

C. CHIRA[1], D. DUMITRESCU[2], AND R. GĂCEANU[3]

Abstract. Problems classified in complexity theory as NP-hard are inherently harder than those that can be solved non-deterministically in polynomial time. Many solutions adopt nature-inspired metaheuristics to solve NP-difficult problems. A hybrid metaheuristic - called Stigmergic Agent System (SAS) - combining the strengths of Ant Colony Systems and Multi-Agent Systems concepts is proposed. The aim of the SAS model is to address NP-hard problems by exploring the solution space using cooperative proactive agents guided by direct and stigmergic communication. Numerical experiments use the Traveling Salesman Problem to evaluate the introduced algorithm. Testing results indicate the great potential of the proposed SAS metaheuristic for complex problems.

1. INTRODUCTION

Metaheuristic techniques refer to strategic frameworks for solving a wide variety of problems as opposed to individual heuristic algorithms designed to solve a specific problem. High-quality near optimal solutions for real-world complex problems can be efficiently identified. Metaheuristics inspired from nature represent a powerful and robust approach to solve NP-hard problems [12].

The aim of this paper is to combine the Ant Colony Optimization [5, 6] approach to solve NP-hard problems with elements of Multi-Agent Systems [4, 7]. A new hybrid metaheuristic called Stigmergic Agent System (SAS) able to better address NP-hard problems is described and investigated. The SAS model involves several cooperating agents able to communicate both directly and in a stigmergic manner to solve problems.

Evaluation results using Traveling Salesman Problems are presented indicating the robustness of the SAS technique.

2. ANT COLONY SYSTEMS

The ACO (Ant Colony Optimization) metaheuristic is composed of different algorithms in which several cooperative agent populations try to simulate real ants behavior [6, 9].

©2007 Babeş-Bolyai University, Cluj-Napoca

Initially ants wander randomly in order to find food, but they leave pheromone trails in their way. If another ant finds the trail it will likely follow it rather than continue its random path, thus reinforcing the trail.

Over time however, pheromone trails tend to evaporate and thus, in time, longer paths will tend to have a lower pheromone level.

When an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone has(have) a greater probability of being chosen by ants. As a result, ants eventually converge to a short path which hopefully represents the optimum or a near-optimum solution for the target problem.

3. MULTI-AGENT SYSTEMS

The modern approach to Artificial Intelligence (AI) is becoming increasingly centered around the concept of agent.

An agent is anything that can perceive its environment through sensors and act upon that environment through actuators [3]. An agent that always tries to optimize an appropriate performance measure is called a rational agent. Such a definition of a rational agent is fairly general and can include human agents (having eyes as sensors, hands as actuators), robotic agents (having cameras as sensors, wheels as actuators), or software agents (having a graphical user interface as sensor and as actuator).

However, agents are seldom stand-alone systems. In many situations they co-exist and interact with other agents in several different ways. Such a system that consists of a group of agents that can potentially interact with each other is called a multi-agent system (MAS).

The agents of a MAS are considered to be autonomous entities (such as software programs or robots). Their interactions can be either cooperative or selfish [4, 7]. MAS can manifest self-organization and complex behaviors even when the individual strategies of agents are simple.

To share knowledge, agents in a MAS can use an Agent Communication Language such as KQML (Knowledge Query Manipulation Language) [11] or FIPA ACL [10].

4. STIGMERGIC AGENTS

A metaheuristic algorithm called Stigmergic Agent System (SAS) that uses a set of autonomous reactive agents has been proposed [1]. The search space is explored by agents based on direct communication and stigmergic behavior.

4.1. **STIGMERGY.**

Stigmergy occurs as a result of individuals interacting with and changing a environment [9]. Stigmergy was originally discovered and named in 1959 by Grasse, a French biologist studying ants and termites. Grasse was intrigued by the idea that these simple creatures were able to build such complex structures. The ants are not directly communicating with each other and have

no plans, organization or control built into their brains or genes. Nevertheless, ants lay pheromones during pursuits for food, thus changing the environment. Even though ants are not able to directly communicate with each other, they do communicate however - indirectly - through pheromones.

Stigmergy provides a general mechanism that relates individual and colony level behaviors: individual behavior modifies the environment, which in turn modifies the behavior of other individuals.

4.2. **SAS ALGORITHM.** SAS mechanism employs several agents able to interoperate on the following two levels in order to solve problems [1]:

- Direct communication: agents are able to exchange different types of messages in order to share knowledge and support direct interoperation; the knowledge exchanged refers both local and global information.

- Indirect (stigmergic) communication: agents have the ability to produce pheromone trails that influence future decisions of other agents within the system.

The initial population of active agents has no knowledge of the environment characteristics. Each path followed by an agent is associated with a possible solution for a given problem. Each agent leaves pheromone trails along the followed path and is able to communicate to the other agents of the system the knowledge it has about the environment after a complete path is created [1].

The pseudo-code of the SAS algorithm [1] is outlined below:

Algorithm 4.2.1. *Stigmergic Agent System*

Begin
  Set parameters
  Initialize pheromone trails
  Initialize knowledge base
  Loop
    Activate a set of agents
    Each agent is positioned in the search space
    Loop
      Each agent applies a state transition rule to incrementally build a solution
      Next move is pro-actively determined based on stigmergic strategy or direct communication
      A local pheromone updating rule is applied
      Propagate learned knowledge to the other agents
    Until all agents have built a complete solution
    A global pheromone updating rule is applied
    Update knowledge base (using learned knowledge)
  Until endCondition
End.

One of the major properties of an agent is autonomy and this allows agents to take the initiative and choose a certain path regardless of communicated or stigmergic information. Agents can lead the way to the shortest path in a proactive way ensuring that the entire solution space is explored. Agents can demonstrate reactivity and respond to changes that occur in the environment by choosing the path to follow based on both pheromone trails and directly communicated information [1].

Using a purely stigmergic approach the solution of a problem could fall into a local optimum, but due to direct communication ability of the agents they can proactively break out of the local optima and continue to explore the search space in order to find a better solution.

5. SAS EVALUATION

SAS model is implemented and tested for solving the Traveling Salesman Problem.

5.1. **PROBLEM STATEMENT.** Given a number of cities and the costs of traveling from any city to any other city, the Traveling Saleman Problem (TSP) refers to finding the cheapest round-trip route that visits each city exactly once and then returns to the starting city.

An equivalent formulation in terms of graph theory is: Given a complete weighted graph (where the vertices would represent the cities, the edges would represent the roads, and the weights would be the cost or length of that road), find a Hamiltonian cycle with the least weight [9].

5.2. **THE SAS ALGORITHM FOR SOLVING THE TRAVELING SALESMAN PROBLEM.** SAS algorithm for solving TSP is given below:

Algorithm 5.2.1. *Algorithm SAS for TSP*

```
Begin
    *initialize noOfAgents, stigmergyLevel, startingCity, knowledge base
    While(true) execute
        LaunchAgents(noOfAgents, stigmergyLevel, startingCity);
            * wait until all agents finish execution
            * handle best solution found - a global update rule is applied, that is, update
            the pheromone level and store the best solution found so far
    endWhile
End.
```

Algorithm 5.2.2. *LaunchAgents(noOfAgents, stigmergyLevel, startingCity)*

```
Begin
    For i = 0,noOfAgents do
        Agent agent = createAgent(stigmergyLevel, startingCity)
```

** execute the agent's behavior*
endFor
End.

Algorithm 5.2.3. *Subalgorithm AgentBehavior*
Begin
While (a solution is not found) execute
** proactively determine if the next city should be chosen stigmergically or using direct communication*
** if the agent decides to behave stigmergically then the next city to be visited is chosen using standard ACS ; otherwise the next city to be visited is chosen using direct communication with the other agents*
** handle best solution so far - a local update rule is applied, that is, update the pheromone level*
endWhile
End.

The procedure starts by setting algorithm parameters such as the number of agents, the stigmergy level of the agents, the starting city - the knowledge base of the system in general.

The process runs until certain conditions are met. At the first step agents with the given parameters are launched. Once their task is completed, the best found solution is compared with the best already known solution (if any) and a global update is performed.

For updating the pheromone level the following local update rule (see [2]) is used:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\frac{1}{n * L^{+}}, \tag{1}$$

where τ_{ij} represents the stigmergy level of the edge (i, j) at moment t, ρ is the evaporation level and L^{+} is the cost of the best tour.

The global update rule is similar:

$$\tau_{ij} = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t), \tag{2}$$

where $\Delta\tau_{ij}(t)$ is the inverse cost of the best tour.

Agents are autonomous entities meaning that they can choose to ignore the path communicated by the system and proactively choose another city to explore. This is crucial for the SAS success since only using a purely stigmergic approach the solution of a problem could be trapped into a local optimum.

The algorithm allows stigmergic selection of the next city based on the probability (see [2]):

$$p_{ij}^k = \frac{\tau_{iu}(t)[\eta_{iu}(t)]^\beta}{\sum_{o \epsilon J_i^k} \tau_{io}(t)[\eta_{io}(t)]^\beta}, \tag{3}$$

where J_i^k represents the unvisited neighbors of node i by agent k, $\eta_{io}(t)$ is *visibility* and denotes the inverse of the distance from node i to node o and β shows what is more important between the cost of the edge and the pheromone level.

Using direct communication agents can proactively choose another city to explore. So at a certain point in time if an agent decides that it should use direct communication it can ask the other agents if they have already visited a certain city. This way an unexplored city can be identified and the agent can autonomously choose it as its next move (regardless of pheromone trail intensities).

Cooperating proactive agents capable of both direct and stigmergic communication provide a robust way to find a solution greatly reducing the risk of being trapped into local minima.

5.3. **SAS - A CASE STUDY.** The SAS algorithm for solving TSP was tested for different data sets (see Figure 1).

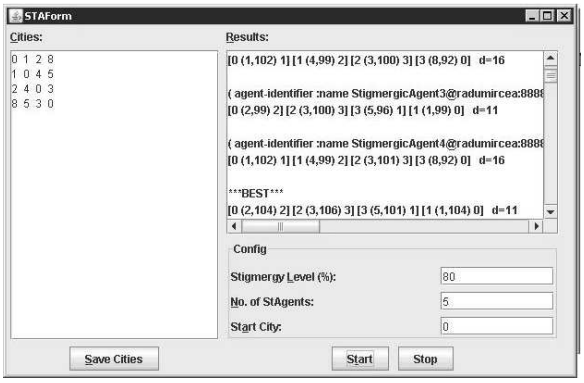


FIGURE 1. SAS application for solving TSP.

In the left pane the cities to be visited are given. The stigmergy level of agents, the number of agents and the starting city can be dynamically configured.

The agents stigmergically find paths of length 16 or 19 at the first iteration. On the given example the agents would be blocked in a local minimum given by nodes [0, 1, 2, 3] of length 16 if they would only act stigmergically. However, at certain moments in time, agents proactively decide to go on another path then the one indicated by the pheromone level, thus avoiding the local optimum.

In the above example, at the second iteration an agent proactively found a better solution (actually the best one) [0, 2, 3, 1] of length 11 and a global update rule was applied for making this new information available to all the other agents.

As the program is running one would observe that more and more agents tend to follow the better path, indicating that the best solution has been probably identified.

Running an ACS algorithm on the same testing data would cause the ants to be trapped in a local minimum of length 16 (when the best tour is of length 11).

5.4. **NUMERICAL EXPERIMENTS.** SAS algorithm for solving TSP is compared to standard Ant Colony System (ACS) model. In the ACS algorithm the values of the parameters were chosen as follows: $\beta = 5$, $\rho = 0.5$.

Table 1 presents comparative results of the proposed SAS algorithm and the ACS model for solving some instances of TSP taken from [5].

Problem	Number of Agents	Number of Generations	Best Known Solution	ACS Result	SAS Result
swiss42	3	30	1273	1589	**1546**
swiss42	5	30	1273	1539	**1517**
swiss42	10	30	1273	1491	**1489**
swiss42	15	30	1273	1472	**1470**
swiss42	20	30	1273	1472	**1439**
bays29	3	30	2020	2312	**2312**
bays29	5	30	2020	2288	**2225**
bays29	10	30	2020	2288	**2209**
bays29	15	30	2020	2288	**2202**
bays29	20	30	2020	2288	**2177**
gr120	3	30	6942	12271	**11999**
gr120	5	30	6942	12220	**10339**
gr120	10	30	6942	9571	**9548**
gr120	15	30	6942	9488	**9488**
gr120	20	30	6942	9488	**8668**

TABLE 1. Comparative testing results

Numerical experiments suggest a beneficial use of direct and stigmergic communication in cooperative multi-agent systems for addressing combinatorial optimization problems.

6. CONCLUSIONS AND FUTURE WORK

The proposed SAS approach is a powerful optimization technique that combines the advantages of two models: Ant Colony Systems and Multi-Agent Systems.

Interoperation between agents is based on both indirect communication - given by pheromone levels - and direct knowledge sharing, greatly reducing the risk of falling into the trap of local minima.

Ongoing research focuses on numerical experiments to demonstrate the robustness of the proposed model. The SAS method has to be further refined in terms of types of messages that agents can directly exchange. Furthermore, other metaheuristics are investigated with the aim of identifying additional potentially benefic hybrid models.

References

[1] C. Chira., C. M. Pintea, D. Dumitrescu, *Stigmergic Agent Optimization*, Romanian Journal of Information Sciences and Technology, pp. 175 - 183, Vol. 9, No. 3, 2006.

[2] C. M. Pintea, P. Pop, C. Chira, *Reinforcing Ant Colony System for the Generalized Traveling Salesman Problem*, Volume of Evolutionary Computing, International Conference Bio-Inspired Computing - Theory and Applications (BIC-TA), pp 245 - 252, Wuhan, China, September 18- 22, 2006.

[3] N.R. Jennings, K.P. Sycara, M. Wooldridge, *A Roadmap of Agent Research and Development*, Journal of Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, 1998, pp. 7-36.

[4] M. Wooldridge, *Intelligent Agents, An Introduction to Multiagent Systems*, ed. G. Weiss, 1999.

[5] The TSPLIB Symmetric Traveling Salesman Problem Instances - http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html

[6] V. Maniezzo, L. M. Gambardella, F. de Luigi - *Ant Colony Optimisation*, New Optimization Techniques in Engineering, by Onwubolu, G. C., and B. V. Babu, Springer-Verlag Berlin Heidelberg, pp. 101-117, 2004.

[7] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2002.

[8] Fabio Bellifemine, Giovanni Caire, Tiziana Trucco, Giovanni Rimassa, *JADE Programmer's GUIDE*, http://jade.tilab.com/, 2006

[9] M. Dorigo, G.D. Caro, *The Ant Colony Optimization Meta-Heuristic*, New Ideas in Optimization, D. Corne, M. Dorigo, F. Glover, 1999.

[10] *http://www.fipa.org/*

[11] T. Finin, Y. Labrou, J. Mayfield, *Kqml as an Agent Communication Language*, Software Agents, B.M. Jeffrey, MIT Press, 1997.

[12] C. Blum and A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM Computing Surveys 35(3) 268-308.

[1] Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania

E-mail address: cchira@cs.ubbcluj.ro

[2] Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania

E-mail address: ddumitr@cs.ubbcluj.ro

[3] Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania

E-mail address: gr90900@scs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 185–192

SENSITIVE ANT SYSTEMS IN COMBINATORIAL OPTIMIZATION

CAMELIA CHIRA[(1)], CAMELIA-M. PINTEA[(1)], AND D. DUMITRESCU[(1)]

Abstract. A new model based on the robust *Ant Colony System (ACS)* is introduced. The proposed *Sensitive ACS (SACS)* model extends ACS using the sensitive reaction of ants to pheromone trails. Each ant is endowed with a pheromone sensitivity level allowing different types of responses to pheromone trails. The SACS model facilitates a good balance between search exploitation and search exploration. Both ACS and SACS models are implemented for solving the *NP*-hard *Generalized Traveling Salesman Problem*. Comparative tests illustrate the potential and efficiency of the proposed metaheuristic.

1. Introduction

Metaheuristics are good strategies in terms of efficiency and solution quality for problems of realistic size and complexity. Regarded as strategic problem solving frameworks, metaheuristics are widely recognized as one of the most powerful approaches for combinatorial optimization problems. The most representative metaheuristics include genetic algorithms, simulated annealing, tabu search and ant colony [6].

The aim of this paper is to design a new metaheuristic based on *Ant Colony System (ACS)* [3] for solving combinatorial optimization problems. The introduced model is called *Sensitive ACS (SACS)* and uses different reactions of sensitive ants to pheromone trails. This technique promotes both search exploitation and search exploration for complex problems. The ACS and SACS models are implemented for solving the *Generalized Traveling Salesman Problem (GTSP)* [7, 8]. Numerical experiments indicate the potential of the introduced SACS model.

2. Ant Colony Systems

An ant algorithm is a system based on agents which simulate the natural behavior of ants including mechanisms of cooperation and adaptation. In [2] the use of this kind of system as a new metaheuristic was proposed in order to solve

2000 *Mathematics Subject Classification.* 90C59, 68T10.
Key words and phrases. metaheuristics, pattern recognition, stigmergy.

©2007 Babeş-Bolyai University, Cluj-Napoca

combinatorial optimization problems. This new metaheuristic has been shown to be both robust and versatile in the sense that it has been successfully applied to a range of different combinatorial optimization problems.

Ant algorithms are based on the following main ideas:

- Each path followed by an ant is associated with a candidate solution for a given problem.
- When an ant follows a path, the amount of pheromone deposited on that path is proportional to the quality of the corresponding candidate solution for the target problem.
- When an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone has(have) a greater probability of being chosen by ants. As a result, ants eventually converge to a short path which hopefully represents the optimum or a near-optimum solution for the target problem.

Well known and robust algorithms include *Ant Colony System (ACS)* [3] and $\mathcal{MAX} - \mathcal{MIN}$ Ant System [10]. *Ant Colony System (ACS)* metaheuristics is a particular class of ant algorithms. The insects behavior is replicated to search the space. While walking between their ant nest and the food source, ants deposit a substance called *pheromone*. In the future every ant can direct its search according to the amount of this hormone on the ground.

3. The Generalized Traveling Salesman Problem

Let $G = (V, E)$ be an n-node undirected graph whose edges are associated with non-negative costs. Let $V_1, ..., V_p$ be a partition of V into p subsets called *clusters*. The cost of an edge $(i, j) \in E$ is $c(i, j)$.

The *generalized traveling salesman problem* (*GTSP*) refers to finding a minimum-cost tour H spanning a subset of nodes such that H contains exactly one node from each cluster V_i, $i \in \{1, ..., p\}$. The problem involves two related decisions: choosing a node subset $S \subseteq V$, such that $|S \cap V_k| = 1$, for all $k = 1, ..., p$ and finding a minimum cost Hamiltonian in S (the subgraph of G induced by S).

Such a cycle is called a *Hamiltonian tour*. The *GTSP* is called *symmetric* if and only if the equality $c(i, j) = c(j, i)$ holds for every $i, j \in V$, where c is the cost function associated to the edges of G.

The *GTSP* has several applications to location and telecommunication problems [4, 5, 7].

4. Ant Colony System for solving GTSP

An *Ant Colony System* for solving the *GTSP* is introduced. Let $V_k(y)$ denote the node y from the cluster V_k. The *ACS* algorithm for solving the *GTSP* works as follows.

Initially the ants are placed in the nodes of the graph, choosing randomly the *clusters* and also a random node from the chosen cluster

At iteration $t+1$ every ant moves to a new node from an unvisited *cluster* and the parameters controlling the algorithm are updated.

Each edge is labeled by a trail intensity. Let $\tau_{ij}(t)$ is the trail intensity of the edge (i,j) at time t. An ant decides which node is the next move with a probability that is based on the distance to that node (i.e. cost of the edge) and the amount of trail intensity on the connecting edge. The inverse of distance from a node to the next node is known as the *visibility*, $\eta_{ij} = \frac{1}{c_{ij}}$.

Each time unit evaporation takes place. This is to stop the intensity trails increasing unbounded. The rate evaporation is denoted by ρ, and its value is between 0 and 1.

To favor the selection of an edge that has a high pheromone value, τ, and high visibility value, η a probability function ${p^k}_{iu}$ is considered. ${J^k}_i$ are the unvisited neighbors of node i by ant k and $u \in {J^k}_i, u = V_k(y)$, being the node y from the unvisited cluster V_k. This probability function is defined as follows:

$$(1) \qquad {p^k}_{iu}(t) = \frac{[\tau_{iu}(t)][\eta_{iu}(t)]^\beta}{\Sigma_{o \in {J^k}_i}[\tau_{io}(t)][\eta_{io}(t)]^\beta},$$

where β is a parameter used for tuning the relative importance of edge cost in selecting the next node. ${p^k}_{iu}$ is the probability of choosing $j = u$, where $u = V_k(y)$ is the next node, if $q > q_0$ (the current node is i). q is a random variable uniformly distributed over $[0,1]$ and $0 \leq q_0 \leq 1$. If $q \leq q_0$ the next node j is chosen as follows:

$$(2) \qquad j = argmax_{u \in J_i^k}\{\tau_{iu}(t)[\eta_{iu}(t)]^\beta\},$$

After each transition the trail intensity is updated using the local correction rule:

$$(3) \qquad \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_0.$$

Only the ant that generate the best tour is allowed to *globally* update the pheromone. The global update rule is applied to the edges belonging to the *best tour*. The correction rule is

$$(4) \qquad \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau(t),$$

where $\Delta\tau(t)$ is the inverse cost of the best tour.

The *ACS* for *GTSP* algorithm shown in the following, computes for a given time $time_{max}$ a (sub-)optimal solution for the given problem.

5. Proposed Sensitive Ant Colony System Model

The proposed *Sensitive Ant Colony System (SACS)* emphasizes a more robust and flexible system obtained by considering that not all ants react in the same way to pheromone trails. Within the proposed model, each ant is endowed with a pheromone sensitivity level denoted by PSL which is expressed by a real number in the unit interval [0, 1]. Extreme situations are:

- If $PSL = 0$ the ant completely ignores stigmergic information (the ant is 'pheromone blind');
- If $PSL = 1$ the ant has maximum pheromone sensitivity.

Small PSL values indicate that the ant will normally choose very high pheromone levels moves (as the ant has reduced pheromone sensitivity). These ants are more independent and can be considered environment explorers. They have the potential to autonomously discover new promising regions of the solution space. Therefore, search diversification can be sustained.

Ants with high PSL values will normally choose any pheromone marked move. Ants of this category are able to intensively exploit the promising search regions already identified. In this case the ant's behavior emphasizes search intensification.

During their lifetime the ants may improve their performance by learning. This process translates to modifications of the pheromone sensitivity. The PSL value can increase or decrease according to the search space topology encoded in the ant's experience.

6. Sensitive Ant Colony System for solving GTSP

The proposed SACS model for solving *GTSP* is described. Two ant colonies are involved. Each ant is endowed with a pheromone sensitivity level (PSL). Ants of the first colony have small PSL values indicating that they normally choose very high pheromone level moves. These sensitive-explorer ants are called *small PSL-ants (sPSL)*. They autonomously discover new promising regions of the solution space to sustain search diversification. Ants of the second colony have high PSL values. These sensitive-exploiter ants called *high PSL-ants (hPSL)* normally choose any pheromone marked move. They intensively exploit the promising search regions already identified by the first ant colony.

SACS for solving *GTSP* works as follows:

Step 1. Initially the ants are placed randomly in the nodes of the graph.

Step 2. At iteration $t + 1$ every *sPSL-ant* moves to a new node and the parameters controlling the algorithm are updated. When an ant decides which node is the next move it does so with a probability that is based on the distance to that node and the amount of trail intensity on the connecting edge. At each time unit evaporation takes place. This is to stop the intensity trails increasing unbounded. In order to stop ants visiting the same node in the same tour a tabu list is maintained. This prevents ants visiting nodes they have previously visited.

To favor the selection of an edge that has a high pheromone value, τ, and high visibility value, η a function ${p^k}_{iu}$ is considered. ${J^k}_i$ are the unvisited neighbors of node i by ant k and $u \in {J^k}_i$. ${p^k}_{iu}$ is the probability of choosing $j = u$ as the next node if $q > q_0$ (the current node is i). If $q \leq q_0$ the next node j is chosen as in Equation 2.

The sensitivity level is denoted by s and its value is randomly generated in $(0, 1)$. For *sPSL* ants s values are in $(0, s_0)$, where $0 \leq s_0 \leq 1$.

Step 3. The trail intensity is updated using the local rule as following.

$$\tau_{ij}(t+1) = s^2 \cdot \tau_{ij}(t) + (1-s)^2 \cdot \Delta\tau(t)\frac{1}{n}. \tag{5}$$

where n is the total number of the nodes.

Step 4. Step 2 and Step 3 are reconsidered by the *hPSL-ant* using the information of the *sPSL* ants. For *hPSL* ants s values are randomly chosen in $(s_0, 1)$.

Step 5. Only the ant that generates the best tour is allowed to *globally* update the pheromone. The global update rule is applied to the edges belonging to the *best tour*. The correction rule is Equation 4.

A run of the algorithm returns the shortest tour found. In the *SACS* algorithm for *GTSP* the implementation of the pheromone trail τ, in order to obtain more qualitative results comparing to the *ACS* for *GTSP* is improved.

The description of the *SACS* algorithm for *GTSP* is shown in Algorithm 1.

Algorithm 1. Sensitive Ant Colony System for GTSP

```
begin
  Set parameters, initialize pheromone trails
  Loop
    Place ant k on a randomly chosen node
    from a randomly chosen cluster
    Loop
      Each sPSL-ant incrementally build a solution (1)(2)
      A local pheromone updating rule (5)
      Each hPSL-ant incrementally build a solution (1)(2)
      A local pheromone updating rule (5)
    Until all ants have built a complete solution
    A global pheromone updating rule is applied (4)
  Until end_condition
end.
```

7. Numerical Experiments

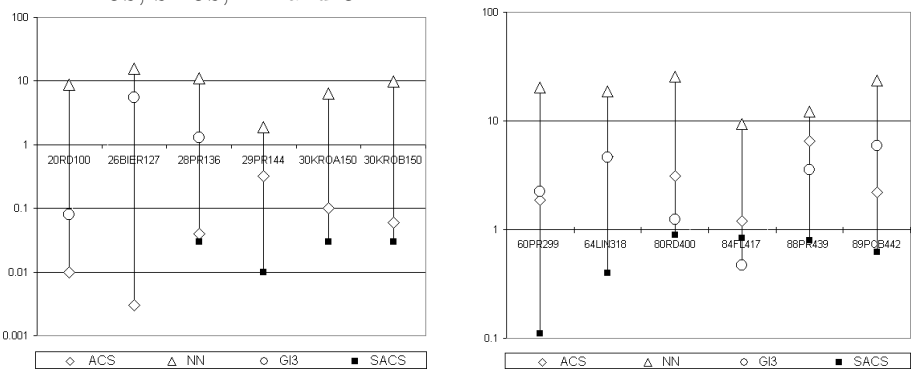
To evaluate the performance of the proposed model, the *SACS* algorithm for solving *GTSP* has been compared to the *ACS* algorithm, the *Nearest Neighbor (NN)* technique and the composite heuristic GI^3 [9]. Problems from *TSP* library [1] have been considered. *TSPLIB* provides optimal objective values for each of the problems. Several problems with Euclidean distances have been considered. Comparative results are shown in Table 2. To divide the set of nodes into subsets the procedure proposed in [4] has been used. This procedure sets the number of clusters $[n/5]$, identifies the m farthest nodes from each other, called centers, and assigns each remaining node to its nearest center.

The parameters used for both ant-based algorithms have been chosen as follows: τ_0=0.1, $\beta = 5$, $m = 10$, $\rho = 0.05$, $q_0 = 0.5$.

Besides the settings inherited from *ACS*, the *SACS* algorithm for *GTSP* uses an sensitivity parameter $s_0 = 0.5$. The sensitivity level of *hPSL* ants is considered to be distributed in the interval $(s_0, 1)$ while *sPSL* ants have the sensitivity level in the interval $(0, s_0)$.

All the solutions of *ACS* and *SACS* for *GTSP* are the average of five successively runs of the algorithm for each problem. Termination criteria is given by the $time_{max}$ the maximal computing time set by the user; in this case ten minutes. Figure 1 shows comparative computational results for solving the *GTSP* using the *ACS*, *SACS*, *NN* and GI^3.

Figure 1. Comparative Standard Deviation Average values for *ACS*, *SACS*, *NN* and GI^3



The *ACS* algorithm for *GTSP* performes well finding good solution in many cases. The test results clearly show that the newly introduced *SACS* algorithm outperforms the basic *ACS* model and obtains better results for most problems than those of the *NN* and GI^3.

TABLE 1. SACS algorithm for solving GTSP versus other algorithms

Problem	Opt.val.	NN	GI^3	ACS	SACS
11EIL51	174	181	**174**	**174**	**174**
14ST70	316	326	**316**	**316**	**316**
16EIL76	209	234	**209**	**209**	**209**
16PR76	64925	76554	**64925**	**64925**	**64925**
20RAT99	497	551	**497**	**497**	**497**
20KROA100	9711	10760	**9711**	**9711**	**9711**
20KROB100	10328	**10328**	**10328**	**10328**	**10328**
20KROC100	9554	11025	**9554**	**9554**	**9554**
20KROD100	9450	10040	**9450**	**9450**	**9450**
20KROE100	9523	9763	**9523**	**9523**	**9523**
20RD100	3650	3966	3653	3650.4	**3650**
21EIL101	249	260	250	**249**	**249**
21LIN105	8213	8225	**8213**	8215.4	**8213**
22PR107	27898	28017	**27898**	27904.4	27899.2
22PR124	36605	38432	36762	36635.4	**36619.2**
26BIER127	72418	83841	76439	72420.2	**72418**
28PR136	42570	47216	43117	42593.4	**42582.2**
29PR144	45886	46746	**45886**	46033	45890
30KROA150	11018	11712	**11018**	11029	11021.2
30KROB150	12196	13387	**12196**	12203.6	12199.6
31PR152	51576	53369	51820	51683.2	**51628.6**
32U159	22664	26869	23254	22729.2	**22693**
39RAT195	854	1048	**854**	856.4	**854**
40D198	10557	12038	10620	10575.2	**10562.2**
40KROA200	13406	16415	**13406**	13466.8	13416.8
40KROB200	13111	17945	**13111**	13157.8	13127.4
45TS225	68345	72691	68756	69547.2	**68473.6**
46PR226	64007	68045	**64007**	64289.4	64131
53GIL262	1013	1152	1064	1015.8	**1015.4**
53PR264	29549	33552	29655	29825	**29603.2**
60PR299	22615	27229	23119	23039.6	**22640.6**
64LIN318	20765	24626	21719	21738.8	**20846.8**
80RD400	6361	7996	6439	6559.4	**6417.4**
84FL417	9651	10553	**9697**	9766.2	9731.8
88PR439	60099	67428	62215	64017.6	**60571.6**
89PCB442	21657	26756	22936	22137.8	**21790.6**

8. Conclusions

A new *ACS*- based model called *Sensitive Ant Colony System* is introduced. Within *SACS* ants are endowed with a pheromone sensitivity level. The proposed model emphasizes a more aggressive exploration of the search space facilitating the detection of promising search areas. The *SACS* algorithm is applied for solving GTSP.

The computational results concerning the *SACS* algorithm are good and competitive - in both solution quality and computational time - with the existing heuristics from the literature [9]. Compared with the basic *ACS* model, the *SACS* algorithm produces better results for many of the test cases used. The results can be potentially improved by considering different parameters settings.

References

[1] Bixby, B., Reinelt, G.: http://nhse.cs.rice.edu/softlib/catalog/tsplib.html (1995)
[2] Dorigo, M.: Optimization, Learning and Natural Algorithms. Ph.D thesis, Dipart.di Elettronica, Politecnico di Milano, Italy (1992)
[3] Dorigo, M., Gambardella, L.M.:Ant Colony System:A cooperative learning approach to the Traveling Salesman Problem.IEEE Trans.Evol.Comp.,1,53-66 (1997)
[4] Fischetti, M., Gonzales, J.J.S, Toth, P.: A Branch-and-Cut Algorithm for the Symmetric Generalized Travelling Salesman Problem. Oper.Res. 45, 3, 378-394 (1997)
[5] Fischetti, M., Gonzales, J.J.S, Toth, P.: The Generalized Traveling Salesman and Orienteering Problem. Kluwer (2002).
[6] Glover, F.W., Kochenberger, G.A.: Handbook of Metaheuristics. Kluwer (2002).
[7] Laporte, G., Nobert, Y.: Generalized traveling salesman problem through n sets of nodes: An integer programming approach. INFOR 21, 1, 61–75 (1983)
[8] Noon, C.E., Bean, J.C.: A Lagrangian based approach for the asymmetric generalized traveling salesman problem. Oper.Res. 39, 623–632 (1991)
[9] Renaud, J., Boctor, F.F.: An efficient composite heuristic for the Symmetric Generalized Traveling Salesman Problem. Euro. J. Oper.Res., 108, 3, 571–584 (1998)
[10] Stützle, T., Hoos, H.H.: The $\mathcal{MAX} - \mathcal{MIN}$ Ant System and local search for the traveling salesman problem. Proc. Int. Conf. on Evol. Comp., IEEE Press, Piscataway, NJ, 309–314 (1997)

[1] Faculty of Mathematics and Computer Science, Babes-Bolyai University, Kogalniceanu 1, RO-400084 Cluj-Napoca, Romania
E-mail address: {cchira,cmpintea,ddumitr}@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 193–200

AN EVOLUTIONARY APPROACH FOR THE 3D PACKING PROBLEM

OANA MUNTEAN[(1)]

Abstract. The three-dimensional packing is a real-world problem that arises in different industrial applications such as container ship loading, pallet loading, cargo management, warehouse management, etc. This work requires a system that efficiently place the boxes (objects of rectangular shape) so as to maximize the utilized space. This paper introduces a simple genetic algorithm, called the Genetic Packing Algorithm (GPA). The basic idea is to encode in each chromosome a permutation of the objects to be used. The algorithm is deeply described and several numerical experiments are presented in order to prove its effectiveness.

1. Introduction

The three-dimensional packing is a real-world nowadays problem that arises in different fields of industries dealing with container ship loading, pallet loading, cargo management, warehouse management. The problem may be simply stated as follows: Given a set of three-dimensional rectangular objects (parallelepipeds) it is required to arrange these in a three-dimensional package (container) in an efficient way: so as to maximize the filled space and without overlapping the objects.

The problem of packing boxes is a well-known NP-Complete [5] problem. Many algorithms have been developed on this field. However, no polynomial-time algorithm is known so far for it. Moreover, algorithmic research has provided strong evidence that it is unlikely to find a polynomial algorithm for this kind of problems. Such an algorithm is guaranteed to find an optimal solution in time that, even in the worst case, can be bounded by a polynomial in the size of the input. If no such bound can be guaranteed, the necessary time for solving instances tends to grow very fast as the instance size increases.

Evolutionary algorithms [6] are powerful optimization techniques inspired by Darwinian Theory of evolution and natural selection [4]. They have successfully been used for solving various difficult real-world problems.

2000 *Mathematics Subject Classification.* 68T20, 68W20.
Key words and phrases. 3D packing, Genetic Algorithm, Permutation encoding.

©2007 Babeş-Bolyai University, Cluj-Napoca

A new evolutionary approach (called Genetic Packing Algorithm (GPA)) for the three-dimensional packing problem is suggested in this paper. The individuals are encoded as permutations and are modified by using special genetic operators.

Several numerical experiments have been performed by using some randomly generated data. The relationship between the algorithm performance and various parameter settings has been analyzed. Results show that GPA is able to perform very well even when small populations are used.

The paper is organized as follows. Related work in the field of 3D packing is briefly reviewed in Section 2. Section 3 gives a short description of the 3D packing problem and the required parameters. The proposed algorithm is described in Section 4. Numerical experiments are provided in Section 5. Strengths and weaknesses of this approach are discussed in Section 6. Conclusions and future work directions are suggested in Section 7.

2. Related work

Several heuristic methods for solving this problem have been proposed. Among the most popular algorithms are wall building [8], guillotine cutting [2], cuboid arrangements [1], Tabu search [3], branch-and bound, etc. The wall building approach fills the container in a number of layers across the depth of the container. The stack building packs the boxes into suitable stacks which then are arranged at the floor of the container by solving a two-dimensional packing problem. The guillotine approach is based on a slicing tree representation of the packing. Each slicing tree corresponds to a guillotine partitioning of the container into smaller parts, where the leaf nodes correspond to boxes. The cuboid arrangement approach recursively fills the container with cuboid arrangements (arrangement of similar boxes). Cuboid arrangements will always provide a sufficient support of the boxes.

Only a few evolutionary algorithms were developed in this field. In [7] I. Ikonnen proposed a genetic algorithm for solving three-dimensional packing with convex objects having holes and cavities. This approach is strongly connected to the particular features of the objects to be placed, i.e. the dimensions and the orientation of the objects are very important for representation of the solution.

3. Problem statement

In the three-dimensional packing problem we have a set of n rectangular 3D objects (boxes) $i = 1, 2, \ldots, n$. Each box i has the dimensions denoted by w_i (the width), d_i (the depth) and h_i (the height).

Let us also denote by W, D and H the width, the depth and the height of the rectangular container that has to be filled with these boxes.

The task is to pack the boxes in such a way that the utilized volume (of the container) is maximized. The ideal case is when the whole volume is utilized.

There are some constrains:

- The boxes cannot overlap each other,
- The boxes are packed with each edge parallel to the corresponding container edge,
- The items may or not be rotated. For speed purposed, in this paper we have not allowed rotation. However, the algorithm can be easily adapted to include rotation.

4. Proposed algorithm

Genetic Packing Algorithm uses a specific representation and specific search operators. All elements of the algorithm are deeply discussed in this section.

4.1. **Individual Representation.** Each individual is a permutation specifying the order in which the objects are placed inside the box.

Example

Let us consider a container having the dimensions 30x20x30 units. It is requested to arrange the following boxes inside it:

(1) 9x10x16
(2) 5x9x12
(3) 14x11x4
(4) 22x8x16
(5) 17x7x7
(6) 3x8x9
(7) 9x5x5
(8) 7x7x8
(9) 14x8x9
(10) 4x5x17

An example of GPA chromosome is the following:

$$1, 5, 4, 8, 7, 10, 9, 3, 6, 2$$

The result of packing the boxes into the container as given by the previously described chromosome is given in Figure 1. There are 7 boxes that can be fitted there. The rest of the boxes are not taken into account in this case.

4.2. **Fitness Assignment Process.** The boxes are placed one by one in the container (according to the order given by the permutation encoded into the current chromosome). For each box we find the lowest position where it can be placed. If there are multiple possibilities we choose one of them randomly.

If a box cannot be placed it will skipped and the next box is considered.

The quality of an individual is equal to the uncovered volume (the waste) of the container. Since it is required to fill the entire space of the container we are

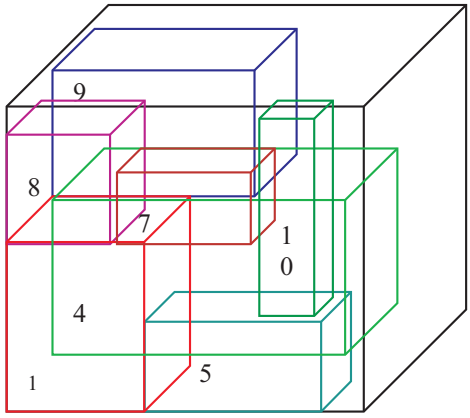


FIGURE 1. An example of packing 10 boxes into a container

dealing with a minimization type problem, which means that the best solution is given by the smallest fitness (we have to minimize it).

4.3. **Genetic Operators.** The main variation operators are recombination (crossover) and mutation.

The crossover operator proposed in this article uses one cutting point value and works as follows: a value p between 1 and n is randomly chosen and then the permutations are traversed from left to right and the values greater than p are exchanged between parents. It is necessary to mention that the interval for choosing this value is not $(1, n)$ as usual, but $(1, n/2)$. The reason for this choice is the fact that there might be some cases when not all the boxes can be used to fill the container. In this case, if the value of p is greater than the number of boxes that fitted into the container the resulting offspring from this recombination will provide the same fitness as their parents.

Mutation of a permutation was performed by swapping two randomly chosen positions.

4.4. **The Algorithm.** Genetic Packing Algorithm (GPA) uses steady state [9] as its underlying mechanism.

GPA starts by creating a random population of individuals. The following steps are repeated until a given number of generations is reached: Two parents are selected using a standard selection procedure. The parents are recombined

in order to obtain two offspring. Each offspring is considered for mutation. The best offspring O replaces the worst individual W in the current population if O is better than W.

Genetic Packing Algorithm is outlined below:

Algorithm 1 Genetic Packing Algorithm

```
1: Randomly create the initial population P(0)
2: for t=1 to NumberOfGenerations do
3:   for k=1 to PopulationSize do
4:     p1 = Select(P(t));{randomly select one individual from the current population}
5:     p2 = Select(P(t)); {select the second individual}
6:     Crossover (p1, p2, o1, o2); {crossover the parents p1 and p2 obtaining the offspring o1 and o2}
7:     Mutation(o1); {mutate the offspring o1}
8:     Mutation(o2); {mutate the offspring o2}
9:     if Fitness(o1) < Fitness(o2) then
10:      if Fitness(o1) < the fitness of the worst individual in the current population then
11:        Replace the worst individual with o1;
12:      else
13:        if Fitness(o2) < the fitness of the worst individual in the current population then
14:          Replace the worst individual with o2;
15:        end if
16:      end if
17:    end if
18:  end for
19: end for
```

5. Numerical experiments

In this section we perform several numerical experiments in order to assess the performance of the Genetic Packing Algorithm. Two statistics are of high interest:

- The relationships between the filled volume and the population size.
- The relationships between the filled volume and the number of generations.

The general parameters of the GPA are given in Table 1.

Test data are taken from a real-world warehouse. There is a high diversity in the size of boxes. The container has 30x30x30. The width, depth and height of the boxes have the values between 5 and 15. Thirty boxes are involved.

TABLE 1. General parameters of the GP algorithm

Parameter	**Value**
Mutations	1 mutation / chromosome
Crossover probability	0.9
Selection	Binary Selection

5.1. **Experiment 1.** In this experiment the relationship between the filled volume and the population size is analyzed. The number of generations is set to 30. The other parameters are given in Table 1. Results are depicted in Figure 2.

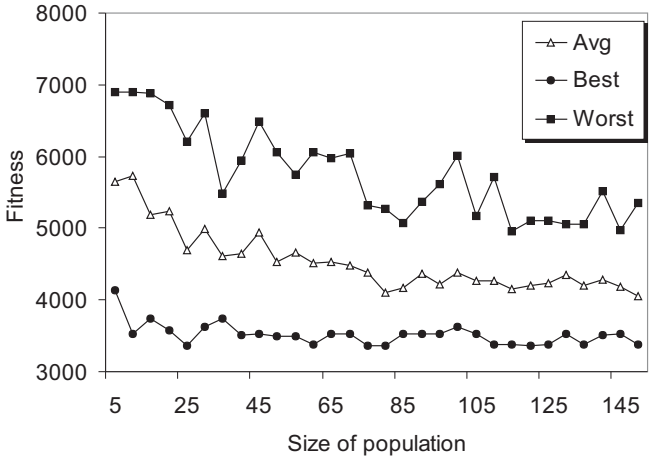


FIGURE 2. The relationship between the quality of the best chromosome and the population size. Results are averaged over 30 runs.

Figure 2 shows that the algorithm performs very well even if the population size is small (5 individuals). This means that we can obtain good results in a very short time.

5.2. **Experiment 2.** In this experiment the relationship between the filled volume and the number of generations is analyzed. The population size is set to 20 individuals. The other parameters are given in Table 1. Results are depicted in Figure 3.

Figure 3 shows that the performance of the GPA improves as the number of generations is increased.

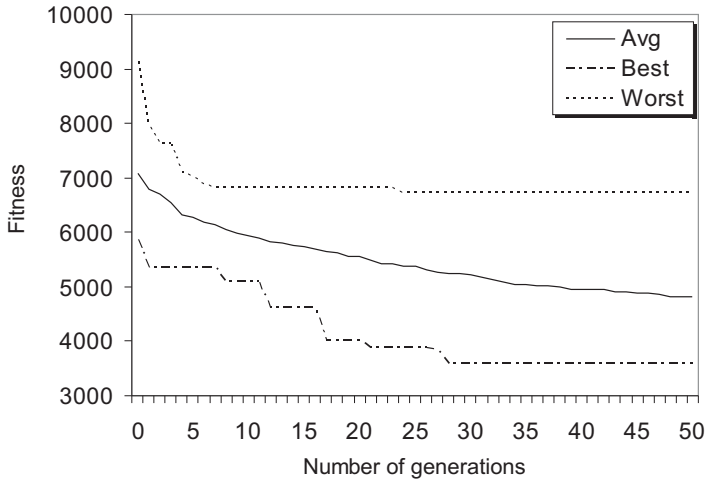


FIGURE 3. The relationship between the quality of the best chromosome and the number of generations. Results are averaged over 30 runs.

6. Strengths and weaknesses

The main reasons for using GA for 3D Packing problem are:

- Simple representation for the solution - each individual is represented as a permutation,
- Short running time - for a near optimal arrangement of 50 boxes, the solution is obtained in several seconds,
- Easy to compute fitness - each object is placed in the lowest position possible. This position is easy to be computed in GA algorithm,
- Good performance of operators - crossover and mutation used are well known and intensively studied, so the search space was efficiently explored.

However, this approach has some limitations:

- Usage of permutations for chromosome's representation can lead sometimes to some major changes in the solution. For example if a small perturbation takes place (e.g. mutation) the placement of the boxes following the changed box will radically change.
- It is needed parameters' tunning for obtaining good results. For instance, if the population is not large enough it is possible that the problem does not converge. More experiments have to be performed in order to adjust the value of GA's parameters.

7. Conclusions and further work

A new genetic approach for 3D packing problem has been proposed in this paper. Chromosomes are encoded as permutations which are varied by using special operators.

Numerical experiments have shown the ability of the algorithm to solve medium size instances of the problem by using very small populations is individuals. This means that we can obtain good results in a very short time.

Further efforts will be focused on analyzing the relationships between other parameters of the Genetic Packing Algorithm (such as mutation probability, crossover probability) and the GPA ability to find a very good solution of the problem.

Different heuristics for placement the boxes inside the container will be investigated in the near future. Their aim is to improve the quality of solutions and to speed up the algorithm.

References

[1] A. Bortfeldt, H. Gehring, and D. Mack. A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing*, 29(5):641–662, 2003.

[2] L. Brunetta and P. Grégoire. A general purpose algorithm for three-dimensional packing. *INFORMS Journal on Computing*, 17(3):328–338, 2005.

[3] T. G. Crainic, G. Perboli, and R. Tadei. An interval graph-based tabu search framework for multi-dimensional packing, Nov. 04 2003.

[4] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray 1^{st} Edition, London, 1859.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, New York, 1979.

[6] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, 1989.

[7] I. Ikonen, W. E. Biles, A. Kumar, J. C. Wissel, and R. K. Ragade. Genetic Algorithm for Packing Three-Dimensional Non-Convex Objects Having Cavities and Holes. In *Proceedings of the 7th International Conference on Genetic Algortithms*, pages 591–598, East Lansing, Michigan, July 1997. Morgan Kaufmann Publishers.

[8] D. Pisinger. Heuristics for the container loading problem, 2002.

[9] G. Syswerda. A study of reproduction in generational and steady state genetic algorithms. In G. J. E. Rawlins, editor, *Proceedings of Foundations of Genetic Algorithms Conference*, pages 94–101. Morgan Kaufmann, 1991.

[1] Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Kogălniceanu 1, Cluj-Napoca, 400084, Romania

E-mail address: oana_muntean85@yahoo.com

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 201–208

MULTI-AGENT DISTRIBUTED COMPUTING

CAMELIA CHIRA[(1)]

Abstract. The paper takes a multi-agent approach to distributed cooperative virtual environments. Distributed computing systems aim to connect human and information resources in a transparent, open and scalable way. Multi-agent systems are investigated with the aim of compiling a successful solution for distributed computing problems. An agent-based architecture is proposed to support distributed computing by enabling interoperation among distributed resources and knowledge integration. The multi-agent approach should address load balancing and resource management problems. It is envisaged to improve the proposed architecture using stigmergic optimization techniques, particularly ant-based procedures for generating load balancing mechanisms.

1. Introduction

Intelligent agents are regarded as the natural metaphor to address distribution of data or control, legacy systems and open systems [13, 22]. Composed of several interacting agents, multi-agent systems (MAS) have the potential to play a crucial role in a large number of application domains including ambient intelligence, computing, electronic business, semantic web, bioinformatics and computational biology [3, 14, 15, 17].

Current approach investigates MAS with the purpose of identifying a suitable architecture to support distributed computing. The proposed *Multi-Agent Knowledge Management and Support System (MAKS)* enables distributed collaboration, supports interoperation of heterogeneous resources and facilitates knowledge sharing, reuse and integration in a virtual environment. Performance and efficiency of the MAKS multi-agent architecture are evaluated for a distributed design environment. Future work emphasizes possible extensions of the proposed architecture to better address load balancing problems.

2000 *Mathematics Subject Classification.* 68M14, 68T30.
Key words and phrases. Multi-Agent Systems, Distributed Computing, Ontologies.

©2007 Babeş-Bolyai University, Cluj-Napoca

2. MULTI-AGENT SYSTEMS AND ONTOLOGIES

An *agent* refers to a system situated in an environment being able to perceive that environment and to act autonomously in order to accomplish a set of objectives [3, 10, 14, 17]. Agents receive inputs about the state of their environment through sensors and they can perform actions through effectors [13]. The main properties of an agent are autonomy, reactivity, pro-activeness, cooperation, learning and mobility [3, 10, 17].

A *multi-agent* approach to developing complex systems involves the employment of several agents capable of interacting with each other to achieve objectives [6]. The benefits of such an approach include the ability to solve large and complex problems, interconnection and interoperation of multiple existing legacy systems and the capability to handle domains in which the expertise is distributed [14, 18]. A MAS is composed of several autonomous and possibly heterogeneous agents [14]. Each agent within the MAS has a limited set of capabilities or incomplete information to solve the problem. The MAS approach implies that there is no global system control, data is decentralized and computation is asynchronous [14].

The interoperation among autonomous agents of MAS is essential for the successful location of a solution to a given problem. Agent-oriented interactions span from simple information interchanges to planning of interdependent activities for which cooperation, coordination and negotiation are fundamental. *Coordination* is necessary in MAS because agents have different and limited capabilities and expertise [16]. The foremost techniques to address coordination in MAS include organisational structuring, Contract Net Protocol, multi-agent planning, social laws and computational market-based mechanisms [3, 16]. *Negotiation* is essential within MAS for conflict resolution and can be regarded as a significant aspect of the coordination process among autonomous agents [14, 16, 19]. Agents within MAS need to *communicate* in order to exchange information and knowledge or to request the performance of a task as they only have a partial view over their environment [14]. Considering the complexity of the information resources exchanged, agents should communicate through an agent communication language (ACL) [9, 17] such as the Knowledge Query and Manipulation Language [9] and FIPA ACL [12].

A meaningful communication process among agents requires a common understanding of all the concepts exchanged by agents. *Ontologies* represent one of the most significant technologies to support this requirement being capable of semantically managing the knowledge from various domains [7, 20]. "Ontologies are explicit formal specification of a shared conceptualization" [21]. Ontologies describe concepts and relations assumed to be always true independent from a particular domain by a community of humans and/or agents that commit to that view of the world [1, 11].

3. MULTI-AGENT KNOWLEDGE MANAGEMENT AND SUPPORT FOR DISTRIBUTED COMPUTING

Emerging enterprise models involve multiple users distributed in a virtual environment who have to cooperate using the software tools available in order to solve problems. Being highly heterogeneous, these users (or teams of people) can be geographically, temporally, functionally and semantically distributed over the enterprise [6]. A computer-based communication network is the work environment where interoperation has to take place [3].

The proposed *Multi-Agent Knowledge Management and Support System (MAKS) architecture* employs multi-agent systems to manage human and information resources in distributed computing environments. Content-related support is ensured by the use of ontologies that handle the information circulated in the environment. The MAKS architecture is composed of the following four major classes of agents (see Figure 1): *(1) User Agents* represent the interface between the system and the end user; *(2) Application Agents* integrate heterogeneous tools by making the application-specific information globally available; *(3) Ontology Agents* manage the information resources of the distributed environment; and *(4) Interoperation Agents* supervise the functionality of the system ensuring agents are meaningfully interconnected and allocation of resources is appropriate.

MAKS Ontology Library composes the machine-enabled framework in which the system's information resources are circulated and stored. The aim is to establish a joint terminology between members of the distributed environment (either humans or agents) by defining concepts, relations and inference rules [3]. MAKS Multi-Agent plane of the proposed architecture specifies the types and behaviours of the software agents required to enable the system's functionality (see [4]). User Agents provide different services to the user and respond to queries and events initiated by the user (or on behalf of the user) with the help of the ontological agents. Examples of User Agents include a User Profile Manager agent (which should act autonomously to manage the profile of the user and should learn user preferences over time) and a User Interface Controller agent (which should provide a customizable graphical user interface based on the user profile). Application Agents are in charge of retrieving information from the software applications called by the user and forward it for storage to the ontological agents. Ontology Agents provide ontology management services in communication networks. They are able to access, retrieve, add, modify and delete information from the Ontology Library. Besides the agents that can read, write and update information (Ontology Reader, Ontology Broker, Ontology Reviser, Component Receiver agents), the ontology agent society should contain agents that are able to supervise the ontology management process ensuring the consistency of the ontology and the delivery of the requested ontology-related services (Ontology Manager agents). The fourth class of agents refers to Interconnection Agents that supervise and support the interoperation

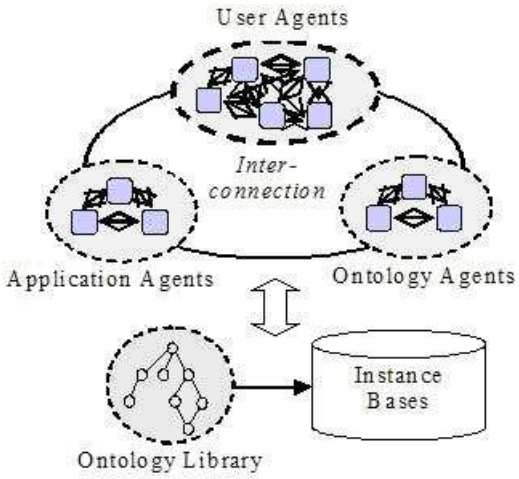


FIGURE 1. MAKS Architecture for Distributed Computing Environments

process among the other agents. Examples of Interconnection Agents include System Manager agents that supervise the overall functionality of the multi-agent system and Directory Facilitator agents that facilitate service discovery.

The MAKS architecture manages resources by employing Application Agents and Ontology Agents to deal with distributed information and knowledge and User Agents to make knowledge easily accessible and shared among dispersed computers. Interoperation Agents address load balancing problems by distributing processing activities across a computer network (creating and activating agents in different locations). Furthermore, mobile Ontology Agents can move around the network to spread information.

4. MAKS EVALUATION

The proposed MAKS architecture has been implemented for the distributed design domain. Designers are able to access information using the proposed multi-agent system in a web format (*MAKS Agent Web Portal*) or based on graphical user interfaces (*MAKS Agent Interface*).

All MAKS agents have been implemented using a Java-based environment for MAS development and are able to take the initiative (i.e. pro-activeness) and interoperate (i.e. cooperation) with other agents in order to achieve their objectives. Moreover, some of the MAKS agents (e.g. User Profile Manager, Application Controller, Ontology Manager, System Manager) should be able to operate on their own without the intervention of users or other agents. Figure 2 presents a possible deployment of the MAKS agents in a distributed engineering design environment.

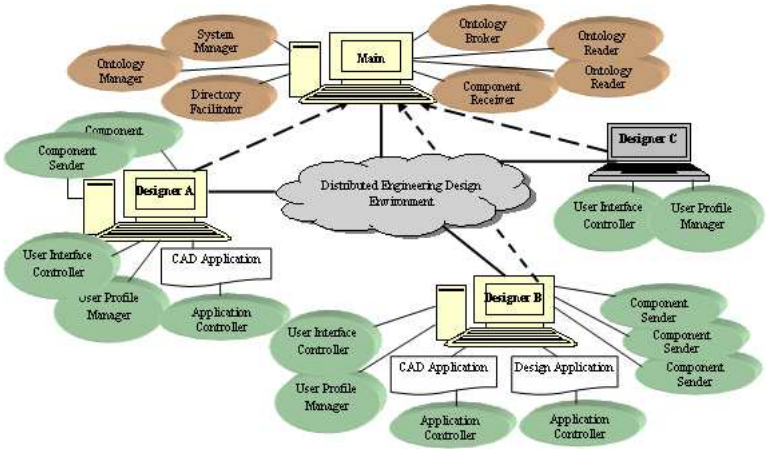


FIGURE 2. MAKS Agents deployment in a distributed design environment

The computer labelled *'Main'* in Figure 2 represents the main platform containing the MAKS manager agents e.g. System Manager, Ontology Manager that supervise the entire agent interoperation process. The other computers in the network are used by different designers each served by a User Interface Controller agent and a User Profile Manager agent (that will have to register with the System Manager and optionally with the Directory Facilitator). These User agents can be accessed through either MAKS Agent Interface or MAKS Agent Web Portal. Furthermore, some designers have one or more Application Controller and Component Sender agents active depending on the number of software applications integrated in MAKS (e.g. the information handled by Designer A using a CAD application is also organized by an Application Controller agent).

To make the design knowledge manageable by MAKS agents, the distributed design domain has been mapped to a library of ontologies defining concepts such as product, property, material, resource and process. The key concept defined in the engineering design ontology is that of a Product considered the final outcome of the design process. Figure 3 shows the UML-based ontology diagram describing the concept of a Product. Each product is viewed as a hierarchy of assemblies and parts, with each assembly being made-up of further assemblies and parts defined in terms of their characteristics (e.g. name, mass, version) and relations (has_author, has_manager, has_feature, has_material).

The testing phase of MAKS for distributed design uses the *protocol analysis (PA)* technique to evaluate the proposed system when used by a single designer or by a team of designers in a distributed environment to perform a given set of tasks. The subjects were videotaped while using the system (MAKS Agent Web Portal

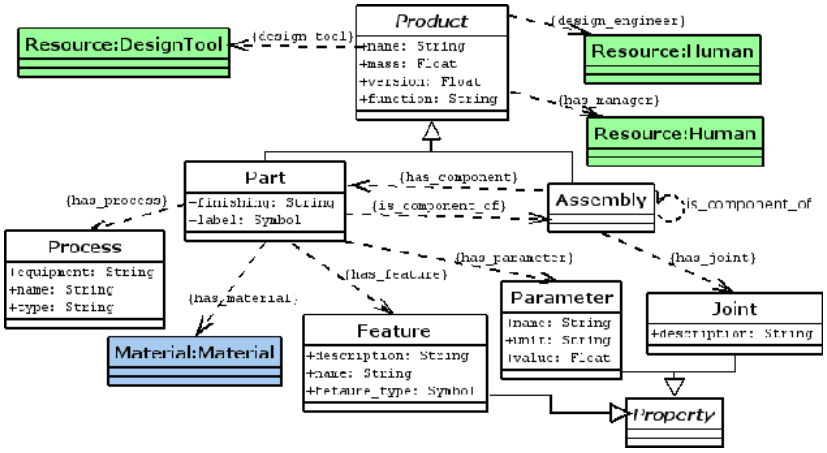


FIGURE 3. UML-based ontology diagram of a product

and MAKS Agent Interface) and verbalizing their thoughts or communicating with other designers (depending on the task). Besides the proposed multi-agent system, subjects were asked to use traditional groupware technologies to complete similar tasks. The intention was to evaluate the proposed system itself using the PA approach and furthermore to compare it with groupware technologies currently used by designers (in a best-case real scenario) to share information in a distributed design environment. The groupware technology selected for this reason is Lotus Sametime Document Repository, which allows logged users to manage documents through a web-based interface.

The transcripts of each PA session were designed to support the capture and analysis of the subject's exact verbalization, the observer's notes and the records of user's actions. The segmentation of episodes is based on the steps and different screens used by the subject in order to complete the given tasks. Figure 4 shows the episode times for each subject in a PA session where the given tasks refer to the retrieval of specific information about a component in a given assembly.

Each subject used the Sametime Document Repository and the proposed multi-agent system (through the MAKS Agent Interface and the MAKS Agent Web Portal) to retrieve the requested information. It is clear that the groupware technology (see Figure 4) was more difficult to be used whereas the MAKS Agent Interface and Web Portal have about the same amount of time allocated.

The PA test results show that agent properties such as autonomy, pro-activeness, cooperation and mobility are highly beneficial to the distributed designer during

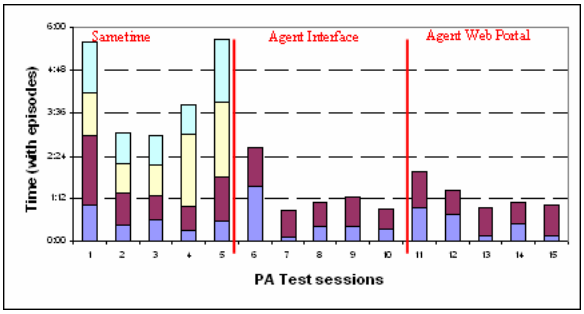


FIGURE 4. MAKS evaluation results: PA episodes for each subject

the information-intensive problem solving process of design. Compared to traditional groupware technologies, the multi-agent approach has clear potential benefits including reliability, robustness and faster access to required information.

5. CONCLUSIONS AND FUTURE WORK

Distributed computing systems aim to connect resources in a transparent, open and scalable way. The agent-based architecture proposed in the current paper employs multi-agent systems for interoperation among distributed resources and ontologies for knowledge sharing, reuse and integration. The proposed system exploits agent properties such as autonomy, cooperation, learning and pro-activeness in a semantic approach to support a process that involves dispersed heterogeneous resources and multidisciplinary people. The main issues addressed by the agent-based architecture include resource management and information flow in communication networks.

Future work focuses on the improvement of MAKS using emerging metaheuristics such as Ant Colony Systems and Evolutionary Computing techniques. Ongoing research focuses on Ant Colony Optimization [5, 8] algorithms and their employment for solving load balancing problems in the enhanced extension of the MAKS system.

REFERENCES

[1] P. Borst, H. Akkermans, J. Top, Engineering Ontologies, International Journal of Human-Computer Studies, Vol. 46, No. Special Issue on Using Explicit Ontologies in KBS Development, 1997, pp. 365-406.

[2] J.M. Bradshow, An Introduction to Software Agents, in Software Agents, J.M. Bradshow, MIT Press, 1997.

[3] C. Chira, The Development of a Multi-Agent Design Information Management and Support System, PhD Thesis, Galway-Mayo Institute of Technology, 2005.

[4] C. Chira, O. Chira, A Multi-Agent System for Design Information Management and Support, International Conference on Computers, Communications and Control (ICCCC 2006), Baile Felix Spa Oradea, Romania, 2006.
[5] C. Chira, C-M. Pintea, D. Dumitrescu, Stigmergic Agent Optimization, Romanian Academy Journal of Information Science and Technology, vol.9, no.3, pp.175-183, 2006.
[6] O. Chira, C. Chira, D. Tormey, A. Brennan, T. Roche, An Agent-Based Approach to Knowledge Management in Distributed Design, Special issue on E-Manufacturing and web-based technology for intelligent manufacturing and networked enterprise interoperability, Journal of Intelligent Manufacturing, Vol. 17, No. 6, 2006.
[7] V.O. Chira, Towards a Machine Enabled Semantic Framework for Distributed Engineering Design, PhD Thesis, Galway-Mayo Institute of Technology, 2004.
[8] M. Dorigo, G.D. Caro, The Ant Colony Optimization Meta-Heuristic, ed; New Ideas in Optimization; ed. D. Corne, M. Dorigo. F. Glover; 1999.
[9] T. Finin, Y. Labrou, J. Mayfield, Kqml as an Agent Communication Language, in Software Agents, B.M. Jeffrey, Ed., MIT Press, 1997.
[10] S. Franklin, A. Graesser, Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996, Berlin, Germany, 1996.
[11] N. Guarino, Formal Ontology and Information Systems, Formal Ontology in Information Systems. FOIS'98, 6-8 June 1998., Trento, IOS Press,; 1998.
[12] http://www.fipa.org, Foundation for Intelligent Physical Agents.
[13] N.R. Jennings, On Agent-Based Software Engineering, Artificial Intelligence,, 2000.
[14] N.R. Jennings, K.P. Sycara, M. Wooldridge, A Roadmap of Agent Research and Development, Journal of Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, 1998, pp. 7-36.
[15] M. Luck, P. McBurney, C. Preist, Agent Technology: Enabling Next Generation Computing, AgentLink, ISBN 0854 327886, 2003.
[16] H. Nwana, L. Lee, N. Jennings, Coordination in Software Agent Systems, BT Technology Journal, Vol. 14, No. 4, 1996, pp. 79-88.
[17] H.S. Nwana, Software Agents: An Overview, Knowledge Engineering Review, Vol. 11, No. 3, 1996, pp. 1-40.
[18] S. Park, V. Sugumaran, Designing Multi-Agent Systems: A Framework and Application, Expert Systems with Applications, Vol. 28, No., 2005, pp. 259-271.
[19] A.S. Rao, M.P. Georgeff, Bdi Agents: From Theory to Practice, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA, 1995.
[20] P. Spyns, R. Meersman, M. Jarrar, Data Modelling Versus Ontology Engineering. ACM SIGMOD Record, 2002.
[21] R. Studer, V.R. Benjamins, D. Fensel, Knowledge Engineering: Principles and Methods, Data and Knowledge Engineering, Vol. 25, No. 1-2, 1998, pp. 161-197.
[22] M. Wooldridge, P. Ciancarini, Agent-Oriented Software Engineering: The State of the Art, ed; Agent-Oriented Software Engineering; ed. P. Ciancarini., M. Wooldridge; AI Volume 1957; 2001.

[1] Department of Computer Science Babes-Bolyai University Cluj-Napoca 1B M. Kogalniceanu, 400084, Romania
E-mail address: cchira@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 209–216

AN EVOLUTIONARY MODEL FOR SOLVING MULTIPLAYER NONCOOPERATIVE GAMES

RODICA LUNG[(1)] AND D. DUMITRESCU[(2)]

Abstract. Computing equilibria of multiplayer noncooperative normal form games is a difficult computational task. In games having more equilibria mathematical algorithms are not capable to detect all equilibria at a time. Evolutionary algorithms are powerful search tools for solving difficult optimization problems. It is shown how an evolutionary algorithm designed for multimodal optimization can be used for solving normal form games.

1. Introduction

Game theory is one of the fields of mathematics that has the largest impacts in the economic and social fields.

What economists call game theory psychologists call the theory of social situations, which is an accurate description of what game theory is about [1]. There are two main branches of game theory: *cooperative* and *non cooperative* game theory. Non cooperative game theory deals largely with how intelligent individuals interact with one another in an effort to achieve their own goals. That is the branch of game theory discussed here.

Solving multiplayer normal form games presenting multiple Nash equilibria is a difficult task that is addressed here by using evolutionary algorithms (EAs). The aim is to show that EAs can be used to detect multiple solutions of a game by transforming the game into a multimodal optimization problem.

2. Prerequisites

Notations and basic notions related to game theory that are necessary for this work are presented in this section.

A finite strategic game is defined by $\Gamma = ((N, S_i, u_i), i = 1, N)$ where:

- N represents the number of players;

2000 *Mathematics Subject Classification.* 91A06, 91A10,68T20.
Key words and phrases. Nash equilibria, evolutionary multimodal optimization.

©2007 Babeş-Bolyai University, Cluj-Napoca

- for each player $i \in \{1, ..., N\}$, S_i represents the set of actions available to him, $S_i = \{s_{i1}, s_{i2}, ..., s_{im}\}$; $S = S_1 \times S_2 \times ... \times S_N$ is the set of all possible situations of the game;
- for each player $i \in \{1, ..., N\}$, $u_i : S \to R$ represents the payoff function.

The following notations are based on [6]

Let $\mathcal{P}_i$ be the set of real valued functions on S_i. The notation $p_{ij} = p_i(s_{ij})$ is used for elements $p_i \in \mathcal{P}_i$.

Let $\mathcal{P} = \times_{i=1,...,N}\mathcal{P}_i$ and $m = \sum_{i=1}^{n} m_i$. Then $\mathcal{P}$ is isomorphic to R^m.

We denote elements in $\mathcal{P}$ by $P = (P_1, P_2, ..., P_N)$ where $P_i = (p_{i1}, p_{i2}, ..., p_{im_i})$.

If $p \in \mathcal{P}$ and $P_i' \in \mathcal{P}_i$ then (P_i', P_{-i}) stands for the element $Q \in \mathcal{P}$ that satisfies $Q_i = P_i'$ and $Q_j = P_j$ for $j \neq i$.

Let Δ_i be the set of probability measures on S_i. We define $\Delta = \times_{i=1,...,N}\Delta_i$. Elements $p_i \in \Delta_i$ are real valued functions on S_i: $p_i : S_i \to R$ and it holds that

$$\sum_{s_{ij} \in S_i} p_i(s_{ij}) = 1,\ p_i(s_{ij}) \geq 0,\ \forall s_{ij} \in S_i.$$

We use the abusive notation S_{ij} to denote the strategy $P_i \in \Delta_i$ with $p_{ij} = 1$. Hence, the notation (S_{ij}, P_{-i}) represents the strategy where player i adopts the pure strategy S_{ij} and all other players adopt their components of P.

The payoff function u_i is extended to have domain R^m by the rule

$$u_i(P) = \sum_{s \in S} P(s)u_i(s),$$

where

$$P(s) = \prod_{i=1}^{N} P_i(s_i).$$

A strategy profile $P* = (P_1*, P_2*, ..., P_N*) \in \Delta$ is a *Nash equilibrium (NE)* if for all $i \in \{1, ..., N\}$ and all $P_i \in \Delta_i$, we have

$$u_i(P_i, P_{-i}*) \leq u_i(P*).$$

Thus a strategy profile $P*$ is a Nash equilibrium if no player can unilaterally increase its payoff when all the other keep theirs unchanged. A normal form game can present more than one NE. Nash [8] proved that there exists at least one NE for any normal form game.

The problem of finding the NEs of a normal form game can be formulated as the problem of detecting all the global minima of a real valued function [5]. This function is constructed using three functions: x, z and g, all defined on $\mathcal{P}$ and having values in R^m. We define the ijth value for this functions for any $p \in \mathcal{P}$, $i \in \{1, ..., N\}$ and $S_{ij} \in S_i$ as

$$x_{ij}(P) = u_i(S_{ij}, P_{-i}) \tag{1}$$

$$z_{ij}(P) = x_{ij}(P) - u_i(P) \tag{2}$$

$$g_{ij}(P) = \max(z_{ij}(P), 0) \tag{3}$$

We define the real valued function $v : \Delta \to R$ by

$$v(P) = \sum_{i=1}^{N} \sum_{j=1}^{m_i} (g_{ij}(P)^2).$$

Function v is continuous, differentiable and satisfies the inequality $v(P) \geq 0$ for all $P \in \Delta$.

A strategy profile $P*$ is a NE if and only if is a global minimum of v, i.e. $v(P^*) = 0$ [5, 4].

3. Evolutionary multimodal optimization

The main problem in dealing with multimodal optimization is to detect and preserve both local and global solutions.

Over the years, various population diversity mechanisms have been proposed that enable Evolutionary algorithms (EAs) to evolve and maintain a diverse population of individuals throughout its search, so as to avoid convergence of the population to a single peak and to allow EAs to identify multiple optima in a multimodal domain. However, various current population diversity mechanisms have not demonstrated themselves to be very efficient as expected. The efficiency problems, in essence, are related to some fundamental dilemmas in EAs implementation. Any attempt of improving the efficiency of EAs has to compromise these dilemmas, which include:

- The elitist search versus diversity maintenance dilemma: EAs are also expected to be global optimizers with unique global search capability to guarantee exploration of the global optimum of a problem. So the elitist strategy is widely adopted in the EAs search process. Unfortunately, the elitist strategy concentrates on some "super' individuals, reduces the diversity of the population, and in turn leads to the premature convergence.
- The algorithm effectiveness versus population redundancy dilemma: For many EAs, we can use a large population size to improve their effectiveness including a better chance to obtain the global optimum and the multiple optima for a multimodal problem. However, the large population size will notably increase the computational complexity of the algorithms and generate a lot of redundant individuals in the population, thereby decrease the efficiency of the EAs.

Two main evolutionary approaches to multimodality have been adopted:

- *Implicit approaches* that impose an equivalent of either geographical separation or of speciation
- *Explicit approaches* that force similar individuals to compete either for resources or for survival

4. Roaming optimization

A recent evolutionary approach to multimodal optimization called Roaming optimization (RO)[2] is presented.

Within Roaming, the tasks of exploitation and exploration are separated. The first one is performed by a group of elitist individuals belonging to an external population called *the archive* while the second one is realized by subpopulations evolving in isolation.

One of the problems facing multimodal optimization techniques is how to decide when an optimum has been detected. Roaming surpasses this problem by introducing a stability measure for subpopulations. This stability measure enables the characterization of subpopulations as stable or unstable.

A subpopulation is considered stable if no offspring is better in terms of fitness function than the best individual in the parent population. Subpopulations that produce offspring better than the best parent are considered unstable and evolve in isolation until they reach stability.

The best individual in a stable subpopulation is considered to be a potential local optimum and included into the archive using a special archiving strategy.

The number of subpopulations is a parameter of the algorithm and it is not related to the expected number of local optima. This confers flexibility and robustness to the search mechanism.

The archive contains individuals corresponding to different optimum regions. The exploitation task is realized by refining the elite individuals in the archive.

The output of the algorithm is represented by the archive - the set of elitist individuals containing local optima.

5. Deflection technique

The deflection technique [3] is an alternative technique that allows the detection of multiple optima during a single run of an optimization algorithm. Let $f : D \to R$, $D \subset R^n$ be the original objective function under consideration.

Let $x_i^*, i = 1, ..., k$ be the k optima (minima) of f. The deflection technique defines a new function $F : D \to R$ as follows:

$$F(x) = T_1(x; x_1^*, \lambda_1)^{-1} \cdot ... \cdot T_k(x; x_k^*, \lambda_k)^{-1} f(x)$$

where λ_i, $i = 1, ..., k$ are relaxation parameters and T_i, $i = 1, ..., k$ are appropriate functions in the sense that the resulting function has exactly the same optima as f except at points x_i^*, $i = 1, ..., k$.

The functions

$$T_i(x; x_i^*, \lambda_1) = \tanh(\lambda_i ||x - x_i^*||),\ i = 1, ..., k,$$

satisfy this property, known as the deflection property as shown in [3].

When an algorithm detects a minimum x_i^* of the objective function, the algorithm is restarted and an additional $T_i(x; x_i^*, \lambda_i)$ is included in the objective function $F(x)$.

6. Experimental results

Roaming optimization is used to solve several normal form games presenting multiple NE. Results are compared with those obtained by two types of heuristics - differential evolution (DE) and particle swarm optimization (PSO) - adapted to detect multiple solutions using the deflection technique. Experimental set-ups and results regarding DE and PSO presented in [9] are used here. Six variants of DE and two variants of PSO were used.

Results are also compared with those obtained using the state-of-art software GAMBIT (ver. 0.2007.01.31) [7], which computes NE by solving systems of polynomial equations.

6.1. **Test problems.** The following test problems presenting multiple NE, available with the GAMBIT software are considered.

GAME1. This is a four players each having two strategies available normal form game. GAME1 has three NE. The corresponding GAMBIT file is $2 \times 2 \times 2 \times 2$.nfg.

GAME2. This is a game with four players, each having two strategies available, having five NE. The corresponding GAMBIT file is g3.nfg.

GAME3. This is a five player game, with two strategies available to each player, having five NE. The corresponding GAMBIT file is $2 \times 2 \times 2 \times 2 \times 2$.nfg.

GAME4. This is a three player game, with two strategies available to each of them, having nine NE. The corresponding GAMBIT file is $2 \times 2 \times 2$.nfg.

6.2. **Experimental set-up.** The parameter setting for RO are presented in table 1. Common parameters used to run DE1-6, PSOc and PSOi are presented in table 2.

TABLE 1. Parameter settings for Roaming

Parameter	GAME1	GAME2	GAME3	GAME4
Subpopulations number	10	10	10	30
Size of subpopulations	10	5	5	3
Number of generations	200	200	300	500
Iteration parameter	1	1	1	1

TABLE 2. Parametter settings for DE and PSO

Problem	Pop. size	Iterations/restart	No. restarts
GAME1	20	1000	8
GAME2	20	1000	10
GAME3	50	2000	10
GAME4	10	1000	15

6.3. **Dealing with constraints.** In order for a point $X = (x_{ij})_{i=1,...,N;j=1,...,m_i}$ to be a NE it must satisfy the constraints naturally arising from the condition $X \in \Delta$, which is

$$\sum_{j=1}^{m_i} x_{ij} = 1, \ \forall i = 1, ..., N.$$

To evaluate the fitness of each individual X the following normalization is used:

$$x'_{ij} = \frac{||x_{ij}||}{\sum_{j=1}^{m_i} ||x_{ij}||},$$

which ensures that $X' \in \Delta$. This normalization is used only to compute the fitness value of individuals and not to constraint the population to lie in Δ.

6.4. **Results.** Descriptive statistics presenting the mean, standard deviation, min and max number of NE obtained for each problem by each method over 30 runs are presented in tables 3-6.

7. CONCLUSIONS

Detecting multiple Nash equilibria of multi-player games is a difficult task that most of the times is addressed by applying an algorithm several times. When the number of equilibria or the number of player increases classical approaches are difficult to apply and not always successful.

Evolutionary algorithms designed to detect multiple optima can be used to find Nash equilibria because solving a normal form game is equivalent to finding all the minima of a function constructed from the game.

TABLE 3. GAME1 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**3**	0	3	3
DE1	2.97	0.18	2	3
DE2	2.93	0.25	2	3
DE3	2.97	0.18	2	3
DE4	3	0	3	3
DE5	3	0	3	3
DE6	3	0	3	3
PSOc	2.97	0.18	2	3
PSOi	3	0	3	3
GAMBIT	3	0	3	3

TABLE 4. GAME2 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**5**	0	5	5
DE1	4.73	0.45	4	5
DE2	4.30	0.47	4	5
DE3	4.63	0.49	4	5
DE4	4.33	0.48	4	5
DE5	0.87	0.51	0	2
DE6	4.47	0.51	4	5
PSOc	4.67	0.48	4	5
PSOi	4.90	0.31	4	5
GAMBIT	5	0	5	5

TABLE 5. GAME3 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**5**	0	5	5
DE1	3.10	0.55	2	4
DE2	1.20	0.71	0	3
DE3	3.17	0.75	2	4
DE4	3.03	0.72	2	5
DE5	1.63	0.76	0	3
DE6	2.57	0.82	1	4
PSOc	3.00	0.69	2	4
PSOi	3.37	0.72	2	5
GAMBIT	5	0	5	5

TABLE 6. GAME4 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**8.83**	0.46	7	9
DE1	6.70	1.09	4	9
DE2	7.17	1.05	5	9
DE3	7.27	0.87	6	9
DE4	7.90	0.76	7	9
DE5	6.80	1.13	4	9
DE6	7.57	0.90	5	9
PSOc	7.03	0.76	5	9
PSOi	6.90	0.96	5	9
GAMBIT	7	0	7	7

Thus six instances of the Differential Evolutionary algorithm and two of the Particle Swarm Optimization algorithm have been adapted using a deflection technique to detect multiple optima. Results are compared with an evolutionary algorithm designed for multimodal optimization called Roaming optimization.

Numerical experiments indicate that EAs are efficient in this task. Among evolutionary techniques, Roaming proved to have the best results for the test problems taken into account.

REFERENCES

[1] last accessed May 2006 levine.sscnet.ucla.edu/general/whatis.htm.
[2] Rodica Ioana Lung and D. Dumitrescu. A new subpopulation model for multimodal optimization. In *IEEE proceedings, SYNASC05*, pages 339–342, Timisoara, September 2005.
[3] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis. On the alleviation of the problem of local minima in back-propagation. *Nonlinear Anal.*, 30(7):4545–4550, 1997.
[4] R. McKelvey and A. McLennan. Computation of equilibria in finite games. 1996.
[5] R. D. McKelvey. A Liapunov function for Nash equilibria. Technical report, 1991.
[6] Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In H. M. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1 of *Handbook of Computational Economics*, chapter 2, pages 87–142. Elsevier, 1 1996.
[7] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory. Technical report, Version 0.2007.01.07, 2006.
[8] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.
[9] N. G. Pavlidis, K. E. Parsopoulos, and M. N. Vrahatis. Computing nash equilibria through computational intelligence methods. *J. Comput. Appl. Math.*, 175(1):113–136, 2005.

[1] BABEŞ-BOLYAI UNIVERSITY
E-mail address: srodica@cs.ubbcluj.ro

[2] BABEŞ-BOLYAI UNIVERSITY
E-mail address: ddumitr@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 219–226

ON SOFTWARE ATTRIBUTES RELATIONSHIP USING A NEW FUZZY C-BIPARTITIONING METHOD

HORIA F. POP AND MILITON FRENŢIU

Abstract. A new data analysis method is introduced, fuzzy bipartitioning method, aimed at producing a set of fuzzy biclusters, that are pairs of fuzzy clusters of items and variables. The paper continues the data analysis of the dependence between software attributes performed in a former paper [14], as a case study to illustrate the importance of this new clustering method. The studied data set is formed by a number of projects written by second year students as a requirement in their curriculum.

1. Introduction

The main purpose of Software Metrics is to improve the Software development process [9]. Software Metrics are also useful to evaluate the quality of a software product [18]. And, as we show in this paper, Software Metrics are useful in education. The future programmer will respect an adequate programming methodology if he is taught to do so. The dependency between some software product attributes was discussed by many authors [1, 2, 25].

This study comes as a continuation of the previous work of both authors [10, 11, 12, 13, 14, 15].

2. The experiment

The study is based on 29 projects produced by second year undergraduate students as part of their requirements curriculum. These projects were analysed observing the attributes described in Table 1. Due to space constraints, the primary data is not given here, but it can be found in [14].

The attributes A10 and A11 were measured automatically by computer. All the others were estimated by postgraduate students. All metrics have the values in the interval [0, 10], where 0 stands for "very bad" (or not present at all), and 10 for "excellent". These values are the subjective evaluation of students. This

2000 *Mathematics Subject Classification.* 68N30.

Key words and phrases. Software metrics, Measurement, Biclustering, Fuzzy Clustering, Data Analysis Techniques, Education.

©2007 Babeş-Bolyai University, Cluj-Napoca

Attribute	Description	Attribute	Description
A1:	requirements description	A16:	readability
A2:	good specification	A17:	comprehensibility
A3:	function points	A18:	changeability (modifiability)
A4:	design clarity	A19:	structuredness
A5:	design correctness	A20:	testability
A6:	design completeness	A21:	reliability
A7:	design diagrams	A22:	efficiency
A8:	modules specification	A23:	extensibility
A9:	algorithms description	A24:	adaptability
A10:	lines of code	A25:	documentation clarity
A11:	no. of comments	A26:	documentation completeness
A12:	good use of comments	A27:	maintainability
A13:	good use of free lines	A28:	simplicity
A14:	indentation	A29:	quality
A15:	good names		

TABLE 1. Attributes description

subjectivity does not affect the attributes relationships, all values for a project being given by the same person. After all, "subjective measures are cheap and worth using" [7]. The definitions of the used attributes are inspired from and can be found in [9].

The attribute A12 refers to the documentation done by comments. It is not based on the number of comment lines of the programs. We may write as many comment lines as we like and sometimes the comments contradict the code, or do not reflect what the code does. The measure for this attribute takes in account if the specification of each module is reflected through comments, if the meaning of each variable and object is explained by comments, if the invariants and other important explanations are given by comments.

3. Overview of biclustering methods

Biclustering is a data mining technique that allows simultaneous clustering of rows and columns. The technique has originally been introduced in 1972 by J.A. Hartigan [16], and the term was first used in 2000 by Cheng and Church [4], in gene expression analysis. The concept of two-mode clustering, with the same meaning, has been introduced in 2004 by Van Mechelen, Bock and De Boeck [24].

Given a set of m rows in n columns, the biclustering algorithm generates biclusters, i.e. a subset of rows that exhibit similar behavior across a subset of columns, and vice-versa. Different biclustering algorithms have different definitions of bicluster.

Considering the relationships among the data, we identify:

- biclusters with constant values;
- biclusters with constant values on rows or columns;
- biclusters with coherent values;
- biclusters with coherent evolutions.

Considering the relationships between biclusters, the following bicluster structures may be obtained:

- exclusive row and column biclusters;
- non-overlapping biclusters with checkerboard structure;
- exclusive-rows biclusters;
- exclusive-columns biclusters;
- non-overlapping biclusters with tree structure;
- non-overlapping non-exclusive biclusters;
- overlapping biclusters with hierarchical structure;
- arbitrarily positioned overlapping biclusters.

For an excellent survey of biclustering algorithms see the survey paper of Madeira and Oliveira [17].

The paper [8] introduces a hybrid genetic fuzzy biclustering algorithm aimed at discovering value-coherent biclusters.

A different approach to biclustering, named cross-clustering, has been introduced in [6] and further studied and improved in [23].

A full-scale comparison of crisp and fuzzy cross-clustering and biclustering algorithms is beyond the purpose of this paper, and will be approached separately.

4. Fuzzy bipartitioning algorithm

The theory of fuzzy sets was introduced in 1965 by Lotfi A. Zadeh [26] as a natural generalization of the classical set concept. Let X be a data set, composed of n data items characterized by the values of s characteristics. A fuzzy set on X is a mapping $A : X \to [0, 1]$. The value $A(x)$ represents the membership degree of the data item $x \in X$ to the class A. The advantage of this approach is that it allows a data item x to be a member of more classes, with different membership degrees, according to certain similarity criteria.

Clustering algorithms based on fuzzy sets have proved their superiority due to their ability to deal with imprecise sets, imprecisely-defined boundaries, isolated points, and other delicate situations. The class of fuzzy clustering algorithms based on fuzzy objective functions [3] provides a large share of geometrical prototypes and combinations thereof, to be used according to the data substructure. On the other hand, the Fuzzy Divisive Hierarchical scheme [5, 19] provides an in-depth analysis of the data set, by deciding on the optimal subcluster cardinality and the optimal cluster substructure of the data set.

Let us consider the main point of biclustering algorihtms. They produce independent clusters formed by a selection of items and a selection of variables, such that the selected items are most similar by considering only the selected variables. Our aim is to generalize this approach in two different ways.

- we aim at producing fuzzy biclusters, not crisp ones;
- we aim at producing a set of biclusters, not only one.

Let us recall here that the fuzzy clustering algorithms of the FCM type use item-prototype dissimilarities based on distances between the item and the geometric prototype.

The main idea behind a fuzzy bicluster is that it is composed by two fuzzy sets: a fuzzy set of items and a fuzzy set of variables. The fuzzy set of variables actually define variables weights to be used in a weighted distance function when producing the fuzzy set of items. Similarly, the fuzzy set of items define the items weights to be used in a weighted distance function when producing the fuzzy set of variables.

This remark suggests an iterative procedure composed by two calls to various fuzzy clustering algorithms using adaptive metrics.

The method proposed in this paper is the following. We start by running a fuzzy horizontal variables clustering method [21]. In this way we will have a fuzzy partition formed by c fuzzy sets of variables.

For each of these c fuzzy sets, we run a fuzzy pointwise regression method[22, 20] (Fuzzy 1-Means), using a distance function weighted by the fuzzy membership degrees of the fuzzy variables set. This, actually, mean using a norm-induced distance with a diagonal norm matrix, with the fuzzy values on the diagonal.

We have now, a set of c fuzzy items sets, each one produced using one of the c fuzzy variables sets. For each of these c fuzzy items sets, we run a fuzzy pointwise regression method (Fuzzy 1-Means), using a distance function weighted by the fuzzy membership degrees of the fuzzy items set. This, actually, mean using a norm-induced distance with a diagonal norm matrix, with the fuzzy values on the diagonal.

At this point we have a set of c fuzzy variables sets, each one produced using one of the c fuzzy items sets. We are going to use these fuzzy variables sets in the same manner in order to produce a new set of fuzzy items sets.

This dual scheme continues until the two sets of fuzzy sets produced at an iteration are close enough to the fuzzy sets produced at the previous iteration.

Let us call this method *Fuzzy C-Bipartitioning (Regression) Method* and let us state it formally (the XT notation denotes the transpose of X).

```
Subalgorithm FuzzyCBipartitioningRegr (X, n, s, c, A, B) is
   Input:  X - data set with n items and s variables
           c - number of clusters to be produced, c>1
```

```
   Output: A - a set of c fuzzy sets of items, A[1], ..., A[c]
           B - a set of c fuzzy sets of variables, B[1], ..., B[c]

   Call FuzzyHorizontalClustering(XT, s, n, c, B)
   Repeat
      Let B' := B;
      For i := 1 to c do
         Call FuzzyPointRegression(X, n, s, B'[i], A[i]);
      End for
      For i := 1 to c do
         Call FuzzyPointRegression(XT, s, n, A[i], B[i]);
      End for
   Until |B-B'| < eps
End subalg
```

A few remarks are in order. Firstly, this method does not produce fuzzy partitions. The sets A_i, $i = 1, \ldots, c$, do not form a fuzzy partition of X, and the same is valid for B_i, $i = 1, \ldots, c$. This is actually not a problem, but it may even be an advantage.

The spread of the fuzzy sets produced using fuzzy regression is controlled by the fuzzy regression method itself. We recall here that these fuzzy regression methods use an input parameter to denote the smallest fuzzy membership value to be assigned.

Finally, our fuzzy bipartitioniong method requires as input the number of fuzzy biclusters to be produced. While this may be considered a major drawback, it is actually not an issue. Let us recall that we are constructing biclusters and bipartitions because, on one side, we do have an idea about the fuzzy cluster substructure of a data set, and we need more info not on the number of relevant clusters, but the relationship between particular data clusters and the variables clusters that group the variables most important to explain these data clusters.

A variation of this algorithm would imply running Fuzzy Clustering instead of Fuzzy Regression at each step. However, this time an adaptive metric would be used on a per class basis. Due to the use of full clustering procedures, this algorithm will construct complete fuzzy partitions both for the data items, and for the variables.

We call this method *Fuzzy C-Bipartitioning (Clustering) Method*. Its formal description follows.

```
Subalgorithm FuzzyCBipartitioningClust (X, n, s, c, A, B) is
   Input:  X - data set with n items and s variables
           c - number of clusters to be produced, c>1
```

```
   Output: A - a partition of c fuzzy sets of items A[1],...,A[c]
           B - a partition of c fuzzy sets of variables B[1],...,B[c]

   Call FuzzyHorizontalClustering(XT, s, n, c, B)
   Repeat
      Let B' := B;
      Call FuzzyHorizontalAdaptiveClustering(X, n, s, B', A);
      Call FuzzyHorizontalAdaptiveClustering(XT, s, n, A, B);
   Until |B-B'| < eps
End subalg
```

5. Case study. Data analysis of software attributes

Based on the analyses performed in the previous papers, we have selected for our case study the use of the Fuzzy 5-Bipartitioning (Clustering), i.e. we aim at a partition of five classes. The bipartition obtained by defuzzyfication from the final fuzzy bipartition is available in Table 2. Due to space constraints, we are not giving here the table of fuzzy membership degrees to the five biclusters. This data is available upon request from the authors. The clustering error used for this case study is $eps = 10^{-5}$.

Class	Projects	Attributes
1	9 12 16 18 19 20 27	1 3 6 8 10 11 13 14 18 26 27 28 29
2	1 2 4 5 6 7 8 13 14 15 17 21 22 23 24 25 26 28 29	5 15 20 23 24
3	10	2
4	11	16 17
5	3	4 7 9 12 19 21 22 25

Table 2. The final fuzzy bipartition for five classes

A few comments with respect to the validity of these results. The first issue we should note is that we are discussing a fuzzy clustering method. As such, the whole set of advantages of fuzzy sets come as well with this method. There are many items with important fuzzy membership degrees to more than one class. See, for example, projects 16, 18, 20, and with a lesser extent, projects 8, 9, 12, 14, 19, 25, 27, all of these with respect to classes 1 and 2. As well, see attributes 29, 24, 20, 18, 15, 13, 5 (with respect to classes 1 and 2), attribute 7 (with respect to classes 1 and 5), attribute 25 (with respect to classes 2 and 5). These fuzzy membership degrees illustrate mixed behavior, not possible to be seen using crisp clustering methods.

A special mention for the qualitative attributes 18 to 29. They are distributed among three classes, many of them with important fuzzy membership degrees. The subjectivity of qualitative evaluations suggested by these attributes is indicated very clearly here by the fact that attribute 29, 'quality', has membership degrees of 0.556 and 0.436 to classes 1 and 2.

Similar correlations between numerical attributes, and between the design attributes, may be seen as well.

For project 10, the attribute 2 is considered very good, meanwhile the majority of the other attributes are very less important.

The clustering of projects 11 and 3 in separate classes is supported by an analysis of their attributes values. They show a lack of correlation between closely related attributes, issues that are normally not found among other projects. For example, project 3 has an overall quality grade of 5, even if all individual quality grades are 5, 6 or 7, the majority of them being greater than 5.

6. Further work and Concluding remarks

The method highlighted here is subject to be refined into a large family of fuzzy bipartitioning algorithms, each such method oriented towards a particular issue mentioned above. Further papers will, of course, have to consider:

(a) hierarchic fuzzy bipartitioning algorithms;
(b) refined algorithms producing sets of biclusters;
(c) refined algorithms producing actual fuzzy partitions;
(d) algorithms using different adaptive metrics, suitable for clusters of non-spherical shapes.

References

[1] Baecker R., Marcus A., *Design Principles for the Enhanced Presentation of Computer Program Source Text*, CHI'86 Proceedings, 51–58

[2] Basili V.R., Selby R.W., Hutchens D.H., *Experimentation in Software Engineering*, IEEE Transactions on Software Engineering, Vol. Se-12 (1986), no.7, 733–743

[3] Bezdek J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981

[4] Cheng Y., Church G.M., *Biclustering of expression data*, Proceedings of the 8-th International Conference on Intelligent Systems for Molecular Biology (2000), 93-103

[5] Dumitrescu D., *Hierarchical pattern classification*, Fuzzy Sets and Systems 28 (1988), 145–162

[6] Dumitrescu D., Pop H.F., Sârbu C., *Fuzzy hierarchical cross-classification of Greek muds*, Journal of Chemical Information and Computer Sciences 35 (1995), 851–857

[7] Dunsmore A., Roper M., *A Comparative Evaluation of Program Comprehension Measures*, EfoCS-35-2000, Department of Computer Science, University of Strathelgde, Glasgow, 2000

[8] Fei X., Lu S., Pop H.F., Liang L.R., *GFBA: A Genetic Fuzzy Biclustering Algorithm for Discovering Value-Coherent Biclusters*, Proceedings of the 2007 International Symposium on Bioinformatics Research and Applications, Georgia State University, Atlanta, Georgia, May 6-9, 2007

[9] Fenton N.E., *Software Metrics*. A Rigorous Approach, Int. Thompson Computer Press, London, 1995
[10] Frenţiu M., *On programming style – program correctness relation*, Studia Univ. Babes-Bolyai, Series Informatica 45, 2 (2000), 60–66
[11] Frenţiu M., *The Impact of Style on Program Comprehensibility*, "Babeş-Bolyai" University of Cluj-Napoca, Research Seminar on Computer Science, 2002, pp. 7–12
[12] Frenţiu M., Lazăr I., Pop H.F., *On individual projects in software engineering education*, Studia Universitatis Babes-Bolyai, Series Informatica 48, 2 (2003), 83–94
[13] Frenţiu M., Pop H.F., *A study of licence examination results using Fuzzy Clustering techniques*, Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Research Seminars, Seminar on Computer Science, 2001, 99–106
[14] Frenţiu M., Pop H.F., *A study of dependence of software attributes using data analysis techniques*, Studia Universitatis Babes-Bolyai, Series Informatica 47, 2 (2002), 53–66
[15] Frenţiu M., Pop H.F., *Tracking mistakes in software measurement using fuzzy data analysis*, In The 4-th International Conference RoEduNet Romania (Sovata, Târgu-Mures, Romania, May 20-22 2005), pp. 150–157
[16] Hartigan J.A., *Direct clustering of a data matrix*, Journal of the American Statistical Association 67, 337 (1972), 123–129
[17] Madeira S.C., Oliveira A.L., *Biclustering Algorithms for Biological Data Analysis: A Survey*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1 (2004), 24–45
[18] McConnell S., *Software Quality at Top Speed, Software Development*, 1996, `http://www.construx.com/stevemcc`
[19] Pop H.F., *SAADI: Software for fuzzy clustering and related fields*, Studia Universitatis Babeş-Bolyai, Series Informatica 41, 1 (1996), 69–80
[20] Pop H.F., *Development of robust fuzzy regression techniques using a fuzzy clustering approach*, Pure Mathematics and Applications 14, 3 (2004), 221–232.
[21] Pop H.F., *Data analysis with fuzzy sets: a short survey*, Studia Universitatis "Babes-Bolyai", Series Informatica 49, 2 (2004), 111–122.
[22] Pop H.F., Sârbu C., *A new fuzzy regression algorithm*, Analytical Chemistry 68, 5 (1996), 771–778.
[23] Pop H.F., Sârbu C., *The fuzzy hierarchical cross-clustering algorithm. Improvements and Comparative study*, Journal of Chemical Information and Computer Sciences 37, 3 (1997), 510–516.
[24] Van Mechelen I., Bock H.H., De Boeck P., *Two-mode clustering methods: a structured overview*, Statistical Methods in Medical Research 13, 5 (2004), 363–94.
[25] Vessey I., Weber R., *Some Factors Affecting Program Repair Maintenance: An Empirical Study*, Comm. A.C.M., 26 (1983), no. 2, 128–134.
[26] Zadeh L.A., *Fuzzy sets*, Information and Control 8 (1965), 338–353.

Babeş-Bolyai University, Department of Computer Science, Faculty of Mathematics and Computer Science, 1 M. Kogălniceanu Street, 400084 Cluj-Napoca, Romania
E-mail address: `hfpop@cs.ubbcluj.ro`
E-mail address: `mfrentiu@cs.ubbcluj.ro`

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 227–235

SOME FORMAL APPROACHES FOR DYNAMIC LIFE SESSION MANAGEMENT

F. BOIAN[(1)], D. BUFNEA[(2)], A. VANCEA[(3)], A. STERCA[(4)], D. COJOCAR[(5)], AND R. BOIAN[(6)]

Abstract. At this moment, the lifetime of a Web session is rigidly established at the level of the Web application and certain implicit constant values are suggested for this duration in the same stiffly manner by most of the Web technologies. This paper introduces some formal models for determining, establishing and dynamic maintaining the lifetime of a HTTP session. To achieve these goals we took into account also the personalized type of work every user does and the particular way in which he or she interacts with the Web application/server. In the following, three different formal approaches will be presented regarding the lifetime management of a Web session.

1. Introduction

If Tim Berners-Lee would have realized the impact and the ampleness which the HTTP protocol would have in the widespread and development of the Internet network at the beginning of '90s (but also the huge number of problems that this protocol had to face) then probably the specifications of this protocol would have been quite different. Today, using the Internet and the Web has acieved a social aspect, being a usual part of the everyday life of many. If at the beginning, the Web and the HTTP protocol were designed for presenting and exchanging documentats, manuals or scientific content information between researchers and academics, once the commercial Internet emerged and spread towards ordinary people, the actual requirements and challenges that this protocol and the HTML language has to face have changed. As the popularity and requirements of web applications grow higher, the HTTP design faces several problems. A set of such problems are related to the stateless dialog model using the HTTP protocol between a client-user and the Web server.

To overcome the drawbacks of the stateless communication model between the client and the server, the web application use the concept of HTTP working session.

2000 *Mathematics Subject Classification.* 90C59, 47N30.
Key words and phrases. dynamic HTTP session management.

©2007 Babeş-Bolyai University, Cluj-Napoca

In its most simple form, an HTTP session may be defined as the set of all the connections issued by a certain client to the Web server involved in solving the same problem at a given moment. The server is responsible for identifying at any given moment to which user a pending request belongs to and for sending back content (dynamically, in most of the cases) accordingly. The returned content varies depending on the requested action and the served user. The most popular mechanisms used to identify the session corresponding to a new request are:

- session id tagging of the requested URLs;
- session id tagging of request's HTTP headers (using cookies);
- SSL based (HTTPS) session management.

While these mechanisms solve the request/session matching succesfully, they do not offer any means for controlling the life-time of the session. The inability to determine the end of a session's life-time comes primarily from the stateless design of the HTTP protocol.

2. Previous work

Most Web technologies apply a rigid and inflexible vision (naive in a certain way) when establishing the lifetime of a session, namely to resort to a fix amount of time (we will call it T_{fixed}), usually 15-30 minutes long. If the user does not issue a request (does not interact with the application/Web server during an T_{fixed} time interval), he is considered "idle" and the working session is invalidated. In other words, from a stateful model perspective, the connection through which the client is served is closed. Although most web technologies allow on the server-side modification of the session expiration time, they do not offer any management or information relative to how this duration can be efficiently changed. Such a mechanism is much more necessary, because most users do not "officially" close the session (they either simply "forget" to click the logout button and either close the browser or leave the web application open for a long time).

Boian presents in [BB06a] an efficient model for calcualting the session lifetime. The mechanism proposed in [BB06a] is based on requiring explicit feedback from the client regarding its status. The time intervals at which this feedback is required are usually much smaller than the session's lifetime, so a fine time granularity is possible for session lifetime calculation. The feedback request intervals remain however rigid, fixed at a constant value s. Therefore, in [BB06a], Boian identifies two types of interactions between the client and the Web server as below (see figure 1):

- Business type actions initiated by the direct interaction between the user and the Web application (GET requests, POST or GET submits);
- *Web-ping* type actions which consist of periodically accessing (at fixed time intervals of s seconds) an URL u by the client, this invocation notifying the server relative to actively maintaining the working session from the client's part.

Client's notification regarding the necessity of accomplishing a *Web-ping* type action is done by using HTTP headers, inserted in the answering pages dynamically generated by the Web server, by means of a field which has the following form: <meta http-equiv="refreh" content="s;url=u">

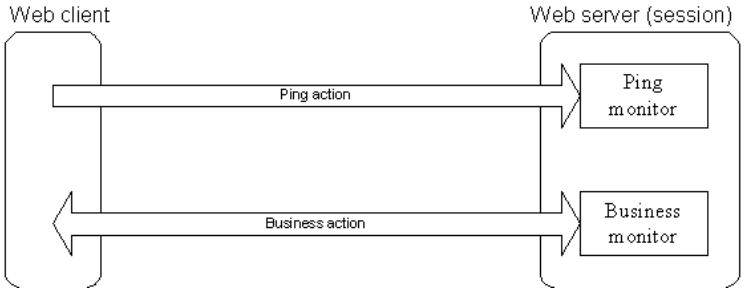


FIGURE 1. Web application architecture with *Web-ping* requests

For the *Web-ping* type actions not to affect the session's lifetime, when receiving such an action by the Web server/application, the T_{fixed} value is diminished by the time elapsed from the moment of the last performed action until the actual moment. For the server to decide if the client (browser) is inactive, it waits maximum r successive *Web-pings*, which follow one after the other without any business action between them. The r value is heuristically established, such that $r \times s << T_{fixed}$.

3. CONTRIBUTIONS

There is a significant number of situations in which the client - Web application interaction assumes frequently repeating different actions initiated by the user - one can identify a certain pattern in the user's behavior. Examples of such activities are: reading the e-mails using a Web-based client, browsing the pages inside a forum, assigning student's grades using the AMS portal [BB06b] etc. In such cases, based on identifying the user's type and time of answer, one can anticipate the user's behavior at the level of the Web server and determine in a more rigorous way a more flexible lifetime (depending on the user/application). The flexibility of the proposed model will be much more significant as the client is explicitly informed about the necessity of offering a feedback in a certain time interval, if it does not directly interact with the Web application.

Anticipating the user's behavior can be done either at the level of the Web application, or at a lower level inside of the application server which hosts that particular application. The application server may export this information to other Web applications by means of an API which the application can decide to

use it or not. The level of reusability, portability and the impact of the proposed mechanism are thus enhanced.

The present paper introduces some formal models for the dynamic management of the session's lifetime and for determining (also as a dynamic value) the time intervals at which the client must notify the Web server/application about its presence. In the exposed approaches we will take into account also the personalized type of work every user has and the particular way in which he interacts with the application.

4. Formal models

We consider until the time moment n, the time intervals elapsed between two user's consecutive interactions (business actions) with the Web application (such an interaction assumes a simple GET request, a submit action either by the GET method either by the POST one). We consider the length of these intervals to be $t_1, t_2, \ldots, t_n$, where t_i is the length of the time interval from the time moment i (the time elapsed between the business actions $i-1$ and i).

4.1. **Statistical approaches.** In [BB06a] the s value was established as an application constant. In the following, after each business action we will compute the maximum probable interval of time in which a business action should normally occur (we will try to anticipate the client's next business action). This value depends on two factors:

- the time moments of client's business actions from the start of the session up to the present time;
- the succession speed of the latest business actions.

Having as input data the random variable $T = (t_1, t_2, \ldots, t_n)$, our goal is to estimate the value t_{n+1}. We will use the following statistics elements [GS97]: mean, standard deviation and linear regression.

The estimation technique is illustrated in figure 2. The figure plots the eleven time intervals between the twelve requests posted by a user to a web application. The linear regression and mean of the elevent points are displayed in continuous lines. Teh dashed lines are parallel to the mean and linear regression lines, but are higher by three standrd deviations.

Let $m_n = \frac{1}{n} \times \sum_{i=1}^{n} t_i$ and $\sigma_n = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^{n} (t_i - m_n)^2}$ be the mean, respectively the standard deviation of random variable T. We also denote by $y = a_n \times X + b_n$ the regression line where $\sum_{i=1}^{n} (a_n \times i + b_n - t_i)^2 \to$ minimum. This line is the best least squares fitting line for points t_i.

Without going into too many details, through simple calculus, we obtain the formulas for computing m_n, σ_n, a_n and b_n as a function of m_{n-1}, σ_{n-1}, a_{n-1}, b_{n-1} and t_n.

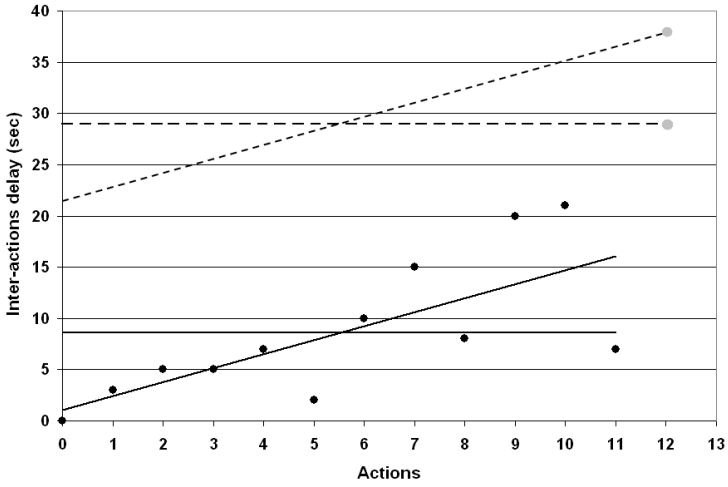


FIGURE 2. Statistical estimation of the latest moment when the next request will arrive

$$M1_n = \sum_{i=1}^{n} t_i = M1_{n-1} + t_n \tag{1}$$

$$M2_n = \sum_{i=1}^{n} t_i^2 = M2_{n-1} + t_n^2 \tag{2}$$

$$MI_n = \sum_{i=1}^{n} i \times t_i = MI_{n-1} + n \times t_n \tag{3}$$

$$m_n = \frac{1}{n} M1_n \tag{4}$$

$$\sigma_n = \sqrt{\frac{1}{n-1}(M2_n - 2m_n \times M1_n + n \times m_n^2)} \tag{5}$$

The simple regresion line $y = a_n x + b_n$ has the following coefficients:

$$a_n = \frac{6[2MI_n - (n+1)M1_n]}{n(n+1)(n-1)} \tag{6}$$

$$b_n = \frac{2M1 - a_n^n(n+1)}{2n} \tag{7}$$

The standard deviation $\overline{\sigma_n}$ of the Z, $z_i = a_i + b - t_i$, random variable is:

$$\overline{\sigma_n} = \sqrt{\frac{1}{n-1}(\frac{a_n^2 n(n+1)(2n+1)}{6} + nb_n^2 + M2_n + n \times \overline{m_n}^2 + \overline{a_n b_n n(n+1) - 2a_n MI_n - a_n \overline{m_n} n(n+1) - 2b_n M1_n - 2nb_n \overline{m_n} + \overline{2\overline{m_n} M1_n)}}}$$

where:

$$\overline{m_n} = \frac{a_n(n+1)}{2} + b_n + \frac{M1_n}{n} \tag{8}$$

The well known three σ rule from statistics says: "the next value of a random variable will be in the interval $[m - 3\sigma, m + 3\sigma]$ with a probability of over 99%". According to this rule, we can expect that $t_{n+1} < m_n + 3\sigma$.

Also, according to a well known heuristic principle taken from operating systems theory [BV07], if at some point in time a certain resource location (memory location, disk location etc.) is being accessed, then the next access will be in the vicinity of the previous one with a very high probability.

In our context, this principle translates to: "if the frequency of the latest business actions' occurrence is high, then it is expected that the frequency of occurrence of the next business actions remains high in the following period". The example from figure 2 depicts such a situation. If this frequency was low, it is expected to remain low. Due to these reasons, we intend to consider the regression line which gives the slope of business actions occurrences frequency. According to this observation we can expect that $t_{n+1} < a_n \times (n+1) + b_n + 3\overline{\sigma_n}$, where $\overline{\sigma_n}$ is the data's standard deviation from the regression line.

Combining the above two principles, we will express t_{n+1} as:

$$t_{n+1} = max(m_n + 3\sigma, a_n \times (n+1) + b_n + 3\overline{\sigma_n})$$

4.2. **A transport level approach for dynamic life session management.** A different approach for the same goal can be considered. It implies the use of a model similar to the one used in the TCP protocol for estimating the nominal value of the round-trip time.

Having at time n an estimation, E_n, for the time that must pass until the user will interact with the web application again, we approximate the length of the time interval between time moment n and time moment $n+1$ as:

$$E_{n+1} = \alpha \times E_n + (1 - \alpha) \times t_n$$

where the interaction time estimation at moment E_i has a higher weight in anticipating the interaction time at moment E_{n+1} than the length of the time interval between time moment $n-1$ and n ($\alpha > 0.5$). This approach eliminates the noise which occurs in estimating the client's interaction time with the web application/server and, on the long run, for a relatively constant behavior of the client makes the estimation value converge to the real value of the interaction time. For an optimal implementation of the above formula, in practice, the value $\alpha = 0.875$ will be used (the two multiply operations will reduce to a bit shift and a substract operation).

For a best fit of this estimated time moment, we use an approximation error interval around the length of the time interval which is computed above. Thus, we approximate the time moment t_{n+1} as:

$$t_{n+1} = E_{n+1} + l_{n+1}$$

where l_{n+1} is a value which increases proportionally to the variation of client's interaction time values (time moment t_{n+1} will be approximated in a bigger time interval):

$$l_{n+1} = \beta \times l_n + (1-\beta) \times (t_n - t_{n-1})$$

where $\beta = 0.875$.

4.3. **The use of the t_{n+1} estimation.** The formal models presented above can be implemented at application level in a web application server to dynamically track the life time of a session depending on the custom behavior of each client. Moreover, once the next interaction time with the web application/server is estimated, the web server can notify the client through an HTTP header (cookie - [FG99]) about the necessity of an explicit feedback (keep-alive message) from him. This feedback can be performed either by a direct action of the user or through a *Web-ping* mechanism like the one described in [BB06a].

In the absence of such a feedback, at time moment $n+1$, in the most conservative approach, the web server/application can decide to invalidate/close the user session even if the life time of such a session is set to a value of $T_{fixed} > t_{n+1}$. A more flexible approach, identical to the one proposed in [BB06a], implies the cancellation of a user session only after $r \times t_{n+1}$ seconds if the web application/server receives no message (action) from the client in this time interval. This approach allows also the cancellation of a user session if the client (browser) does not act.

For example, if x ping messages are received and no business message, then one can consider cancelling the user session for inactivity reasons. The values r and x has to be have heuristic values (we will address this in future papers).

Figure 3 presents real values time evolution vs. estimated ones time evolution.

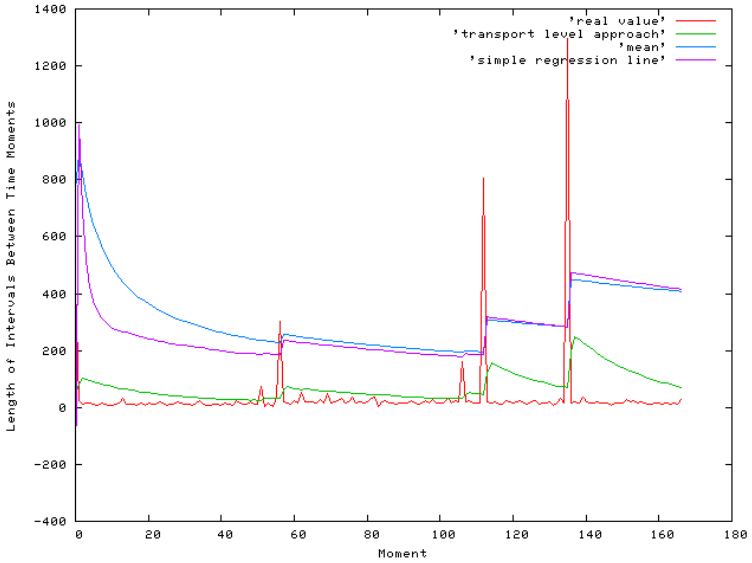


FIGURE 3. Session lifetime estimation results

5. Conclusion and future work

Our paper introduces two formal models for dynamically tracking the life time of an HTTP session, models intended to replace the rigid way of setting a session's life time to a constant value. Also, it proves that a couple of concepts and mechanisms applied to some layers of the TCP/IP stack can be adapted and used at other layers of the stack (in our case, a transport layer mechanism was used at the application layer).

As future work, we intend to implement the proposed models in a web application which is used frequently, e.g. [BB06b]. Also, we want to create an API that extends the functionality of an application server and offers the suggested functionality to all interested applications (migrate the business logic of the proposed models from the web application to the application server layer).

References

[BB06a] F.M. Boian, D. Bufnea, A. Vancea, A. Sterca, D. Cojocar, R. Boian, *A Model for Efficient Session Object Management in Web Applications* in Proceedings of the Symposium "Colocviul Academic Clujean de Informatică" Cluj-Napoca, 2006, pp. 131-136;

[BB06b] F.M. Boian, R. Boian, A. Vancea, *AMS: An Assignment Management System for Professors and Students* in Proceedings of the Symposium "Colocviul Academic Clujean de Informatică" Cluj-Napoca, 2006, pp. 137-142;

[FG99] R. Fielding, R. J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616, June 1999;

[WWW1] *Improved Session Tracking*, http://www.mojavelinux.com/blog/archives/2006/09/improved_session_tracking/, September 2006;

[PA00] V. Paxson, M. Allman, *Computing TCP's Retransmission Timer*, IETF RFC 2988, November 2000;

[GS97] Charles M. Grinstead, J. Laurie Snell, *Introduction to Probability*, American Mathematical Society, July 1997;

[BV07] F. Boian, Al. Vancea, D. Bufnea, C. Cobârzan, A. Sterca, D. Cojocar, *Sisteme de operare*, Editura Risoprint 2006, ISBN 973-751-220-0, 978-973-751-220-8.

(1) Babeş-Bolyai University
E-mail address: florin@cs.ubbcluj.ro

(2) Babeş-Bolyai University
E-mail address: bufny@cs.ubbcluj.ro

(3) Babeş-Bolyai University
E-mail address: vancea@cs.ubbcluj.ro

(4) Babeş-Bolyai University
E-mail address: forest@cs.ubbcluj.ro

(5) Babeş-Bolyai University
E-mail address: dan@cs.ubbcluj.ro

(6) Babeş-Bolyai University
E-mail address: rares@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 236–243

MANAGEMENT OF WEB PAGES USING XML DOCUMENTS

LEON ŢÂMBULEA AND HORIA F. POP

Abstract. The management of a web site is a very difficult task. The paper describes a way of automatic management by memorizing in a database the information sources in different pages. To describe the way to generate a page, a page context is used, which is stated through an XML document.

1. Motivation

Many models are proposed to allow the management of complex web sites (see [1, 2, 3, 4, 5]). In [6] a model allowing collaboration/approval of activity of many users in the development of site is introduced.

This paper describes a modality for automatic management of complex web sites through memorising the site contents in a database and stating the contents of a web page through an XML document.

2. Contents of a web site

The visualisation of a web site (by an Internet browser) meas the succesive display of more pages. We start from an initial page, and following the occurence of a certain event we continue with a different page.

A **page** is built (generated) from a set of elements (sources of information). These elements have different types:

- static elements: blocks of HTML text, XML documents, documents of different types (DOC, RTF, XLS, PPT, and so on);
- menus;
- result of the execution of server scripts;
- and so on.

For each **element type** more elements (realisations of this element type) may exist. Such an element may have, among others:

- an identification (id and type);
- a contents (a sequence of characters that defines this element).

2000 *Mathematics Subject Classification.* 68P99.
Key words and phrases. web sites, management, content management systems, XML.

©2007 Babeş-Bolyai University, Cluj-Napoca

These values may be memorised in independent files (and the identification is made through the file name), or in a database.

For an **element** e, a **value** $v(e)$ may be determined. This value is used to view the element. The way to obtain the value depends on the element type: for a static element, the value is the contents, and for a server script, the value is obtained by interpreting (executing) the script.

The value $v(e)$ may include all the information required by the browser to view. Alternatively, when viewing extrainformation is used, memorised in a **viewing style** s asociated to the element e.

In what follows we will denote by E the set of all elements, disregarding their type, and by S the set of all styles, defined in order to be associated to differet elements in E.

When generating a **web page** the following information is considered:

- A set of necessary elements in the page (subset of E). For each element e its value $v(e)$ (text to be interpreted by the browser) is determined. When generating the element value, a viewing style may be used.
- Positioning the value of each element in the generated page. Such a value will occupy a 2D area on the display where the page will be viewed.
- A viewing style for the whole page, which completes the viewing styles associated to elements.

After a page p is generated and displayed, a new page p' will be generated following an event. The event in page p (for example, the use of a link) appears in the value of an element e in this page. When defining the element a reference to an element $e' \in E$ is needed, or a link to an external document is produced. If there is a reference to an alement e', then from e' the page p' will have to be generated. This generation of the page p' may be done in two different ways:

(1) Each referred element e' is ssociated a set of elements, $A(e') \subset E$. From these elements the new page p' is generated. This association is accompanied by the position each element in $\{e'\} \cup A(e')$ will use on the newly generated page, p'. It is definitely possible that the same page to be generated by events in different elements, so all positioninginformation should be repeated. The main advantage of this approach is that the same elements may generate distinct views, based on the generating event.

(2) A set of **viewing contexts** C are defined. A context $c \in C$ contains:
   (a) a set of elements $c(e) \subset E$;
   (b) a [positioning of the vaues of these elements in the generated page,
   (c) a style associated to the conext.

   Using this information for a context, a page p may be generated. An element e is associated a context $c(e) \in C$. If $e \in c(e)$, then the page is generated using the context $c(e)$. If $e \notin c(e)$, then we need to state a

position of the value of element e in the page generated byu the context $c(e)$.

In what follows we will consider the second alternative. Thus, for each element $e \in E$, the following information is associated:

- a **context** $c(e) \in C$, if e is referred by a certain event;
- the value **null**, if the element e will be included in a viewing context, but is not referred by any possible event (for example, a commercial, banner or block of text in an HTML page, that appear in a certain context, but they are not pointed to by any links in the generated pages.

A context may be associated to more elements, i.e. by referring different events, the same displayed page is obtained.

Viewing a page actually means viewing a context. The first step in this process is to determine the value of each element in the context and to include this value in a certain area in the generated page.

In many situations, the value of an element does not have a constant length, especially if the element is a server script and the value of this element depends on the contents of a database. So, in the generated page, the element value will ocupy an area with no exact area. If, when generating the value of such a context, values of more elements are used, then the HTML text generated from the context, used to view the context, will be formed by HTML tags `<table> ... </table>`. So, we will get a table where the cells will hold the value of an element, or a new included table, corresponding to a subcontext.

The size of a cell holding the value of an element:

- May be determined by the browser based on the values of the other cells in the table and the size of the parent area where the table is included;
- One of the sizes (width, height) is stated through an absolute or relative value with respect to the parent area. The other size will be determined by the browser.
- Both sizes are stated. If the value does not fit in the reserved area, then one of the sizes will be automatically increased by the browser.

3. Description of a web context as an XML document

Let us consider the context depicted in Figure 1, where we denoted by 1, 2, ..., 5a, ..., 10, elements of a particular type t, and the parentheses specify the width of the area where the value is included. The value p% specifies that the width of the cell is relative to the width of the parent area.

The HTML text that allows the extraction of this context should be generated from the "context description", defined as a suitable data structure. This data structure should allow:

(1) Easy run of operations to modify the context:
   (a) insert elements,

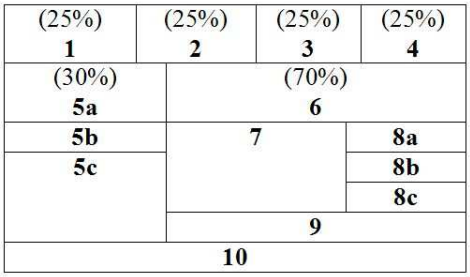


FIGURE 1. Example of a web site context

(b) remove elements of sets of elements that occupy a certain 2D area in the page,
(c) move elments.

(2) HTML text generation by determining the values of included elements and their positioning in the generated page, according the the pattern established by the context.

We will now describe a possibility to memorize contexts by using XML documents.

If two lines in a table may have different structures, i.e. different mumbers of included elements and/or different sizes for these elements, if a line has more than one cell, such a line requires the inclusion of an independent table.

The HTML text that will generate the context view will have `<table> ... </table>` tags. If the use of unions of columns or lines in the table is not esired (`rowspan` and `colspan` attributes in `td`), then not any context may be generated using `<table> ... </table>` tags. See for example the context in Figure 2.

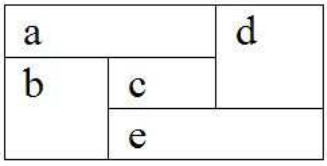


FIGURE 2. A web site context that cannot be generated with table tags and without unions of rows and columns

We are hereby assuming that the tables do not use unions of lines or columns. This particularity for tables is useful in order to easily and quickly perform context management operations (remove, insert, move elements). If, still, situations like that described in Figure 2 are needed, we may define a new element, to contain, by

its definition, a table with this structure, and the context will include this complex element.

Let us enumerate the possbilities to memorize the elements positions in a context.

(1) For the case mentioned above (table with no unions) an XML document may be used. This possibility will be analysed below.
(2) We start from a 2D grid over the display, and the columns may have szes associated. A rectangular area in tis 2D grid is stated for each element included in the contex. This situation is considered with the development of interfaces in 2D visual environments. With such a structure, elements overlaps are possible if the 2D grid has cells of fixed size. Insertions of new elements are more difficult with this way of memorizing the elements positioning.
(3) Even `<table>` tags may be memorized. In this way, the context includes elements, instead of their value.

We are analyzing here the first alternative. This choice does not restrict the possibilities to use this approach, because the largest amount of information on Internet corresponds to this alternative.

We are going to describe the way to generate the HTML text for a context, and the correspondence between the XML tags and HTML tags.

For the XML document we consider the tag `<page [attributes]>...</page>` corresponding to the context generated page. The context, as it has been considered above, has oneor more independent lines. The structure of a line does not generally depend on the structure of another line. When generating the HTML text for a page, a table structure is needed. When generating the table, we need to consider that each line has one value. If one line has more values, then the line will be extracted as an included table. So, two types of lines are possible:

- Line that has one column only, i.e. the value of one element or one included subcontex. The generated HTML text will have `<tr>...</tr>`, and for the XML document we will consier a tag `<rowe [attributes]> ... </rowe>`.
- Line with more than one columns. Will generate an HTML text of the form `<tr><td><table><tr> ... </tr></table></td></tr>`. For such lines, we will consider the XML tag `<rowc [attributes]> ... </rowc>`.

With these considerations, for the context in Figure 1, the following associated XML document will be generated:

```
<table>                                  <page>
  line 1, as table with 1 line, 4 cols     <rowc> line 1 </rowc>
  line 2, as table with 1 line, 2 cols     <rowc> line 2 </rowc>
  line 3, with value of one element        <rowe> line 3 </rowe>
</table>                                 </page>
```

The third line has only one value, so the HTML text and the XML document are:

`<tr><td> v('t',10) </td></tr>`	`<rowe><elem type='t' id='10' /><rowe>`

In the HTML text, the value of an element of type 't' will be denoted by $v('t', id)$. In the XML document, the reference to an element 'id' of type 't' will be done by attributes `type='t' id='id'`.

The first line has four values. So, the HTML text will include a table, and this table, with only one line, will have four columns with the values of one element each:

`<tr>`
`  <td>`
`  <table>`
`  <tr>`
	`<rowc>`

`  <td> v('t','1') </td>`
`  <td> v('t','2') </td>`
`  <td> v('t','3') </td>`
`  <td> v('t','4') </td>`
	`  <elem type='t' id='1' />`
	`  <elem type='t' id='2' />`
	`  <elem type='t' id='3' />`
	`  <elem type='t' id='4' />`

`  </tr>`
`  </table>`
`  </td>`
`</tr>`
	`</rowc>`

The second line in the original table has a more complex structure. To describe a coluumn with more elements, we will consider a tag `<col [attributes]> ... </col>`. For this tag we will create a table cell that includes a new table. So, the generated HTML text is of the form `<td [attributes]><table> ... </table></td>`. In this subcontext, the first two columns are formed by values of more elements, so context tables are needed.

From the examples above, the correspondence between HTML tags and XML tags for memorizing contexts is deduced. This correspondence is given in Figfure 3.

XML tag	HTML tag
<page [attributes]> </page>	<table [attributes]> </table>
<rowe [attributes]> </rowe>	<tr [attributes]> </tr>
<rowc [attributes]> </rowc>	<tr><td><table [attributes]><tr> </tr></table></td></tr>
<elem [attributes] type='t' id='id' />	<td [attributes]> value of element of type 't' with identifier 'id' </td>
<col [attributes]> </col>	<td [attributes]><table> </table></td>

FIGURE 3. The correspondence between HTML tags and XML tags for memorizing contexts

By considering the tag attributes as well, we obtain the correspondence between the XML document and the HTML text from Figure 4 for the context in Figure 1.

```
<table border=1 width="90%">
  <tr><td><table border=1 width="100%"><tr>
    <td width="25%" valign="top">v('t','1')</td>
    <td width="25%" valign="top">v('t','2')</td>
    <td width="25%" valign="top">v('t','3')</td>
    <td width="25%" valign="top">v('t','4')</td>
  </tr></table></td></tr>
  <tr><td><table border=1 width="100%"><tr>
    <td width="30%" valign="top">
      <table border=1 width="100%">
        <tr><td>v('t','5a')</td></tr>
        <tr><td>v('t','5b')</td></tr>
        <tr><td>v('t','5c')</td></tr>
      </table>
    </td>
    <td width="70%" valign="top">
      <table border=1 width="100%">
        <tr><td>v('t','6')</td></tr>
          <tr><td><table border=1 width="100%"><tr>
            <td width="60%" valign="top">v('t','7')</td>
            <td width="40%" valign="top">
              <table border=1 width="100%">
                <tr><td>v('t','8a')</td></tr>
                <tr><td>v('t','8b')</td></tr>
                <tr><td>v('t','8c')</td></tr>
              </table>
            </td>
          </tr></table></td></tr>
        <tr><td>v('t','9')</td></tr>
    </table></td>
  </tr></table></td></tr>
  <tr><td>v('t','10')</td></tr>
</table>
```

```
<page width="90%">
  <rowc width="100%">
    <elem type='t' id='1' width="25%"/>
    <elem type='t' id='2' width="25%"/>
    <elem type='t' id='3' width="25%"/>
    <elem type='t' id='4' width="25%"/>
  </rowc>
  <rowc>
    <col width="30%">
      <rowe><elem type='t' id='5a'/></rowe>
      <rowe><elem type='t' id='5b'/></rowe>
      <rowe><elem type='t' id='5c'/></rowe>
    </col>
    <col width="70%">
      <rowe><elem type='t' id='6'/></rowe>
      <rowc>
        <elem type='t' id='7' width="60%"/>
        <col width="40%">
          <rowe><elem type='t' id='8a'/></rowe>
          <rowe><elem type='t' id='8b'/></rowe>
          <rowe><elem type='t' id='8c'/></rowe>
        </col>
      </rowc>
      <rowe><elem type='t' id='9'/></rowe>
    </col>
  </rowc>
  <rowe><elem type='t' id='10'/></rowe>
</page>
```

FIGURE 4. The correspondence between the XML document and the HTML text for the context in Figure 1

4. CONCLUSIONS

The management of a web site is a very difficult task. The paper describes a way of automatic management by memorizing in a database the information sources in different sources. To describe the way to generate a page, a page context is used, which is stated through an XML document.

The construction of this XML document is difficult to accomplish, especially for pages with a lot of information. For this reason, an application to allow the interactive editing of such an XML document is desired.

REFERENCES

[1] *PhpWebSite project*, Appalachian State University, 2004-2006, http://phpwebsite.appstate.edu.

[2] A. Bonifati, S. Ceri, P. Fraternali, A. Maurino, *Building multi-device, content-centric applications using WebML and the W3I3 Tool Suite*, in Proceedings of Conceptual Modeling for E-business and the Web, Lecture Notes in Computer Science, vol. 1921, 2000, pp. 64–75.

[3] E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, Al. Maedche, B. Motik, D. Oberle, Ch. Schmitz, St. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, V. Zacharias, *KAON – Towards a large scale Semantic Web*, Proc. of the 3rd Intl. Conf. on E-Commerce and Web Technologies (EC-Web 2002), 2002, pp. 304–313.

[4] Y. Jin, S. Decker, G. Wiederhold, *OntoWebber: model-driven ontology-based web site management*, in Proceedings of the first international semantic web working symposium (SWWS'01), Stanford University, Stanford, CA, 29 July – 1 August 2001.

[5] L. Ţâmbulea, H. F. Pop, *Web sites management*, BABES-BOLYAI University of Cluj-Napoca, Faculty of Mathematics and Computer Science, Proceedings of the Symposium "Zilele Academice Clujene", 2006, p. 21–26.

[6] L. Ţâmbulea, H. F. Pop, *Cooperative Model For Web Sites Authoring*, International Workshop in Collaborative Systems, Cluj-Napoca 2006, Annals of the Tiberiu Popoviciu Seminar, Cluj-Napoca, 28–29 October 2006, 329–336.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, 1, M. Kogalniceau Street, 400084 Cluj-Napoca, Romania

E-mail address: leon@cs.ubbcluj.ro

E-mail address: hfpop@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 244–251

A VIEW ON FAULT TOLERANT TECHNIQUES APPLIED FOR MEDIOGRID

DACIAN TUDOR[(1)], VLADIMIR CRETU[(2)], AND HORIA CIOCARLIE[(3)]

Abstract. In this paper we analyze the characteristics of fault tolerance for grid systems, which serves as a basis to identify the features and techniques of the Globus framework and the way these can be applied for the MedioGrid project. Based on the general MedioGrid architecture and the fault tolerant analysis, we suggest an extension of dynamic replication strategies to address data life cycle using dynamic policies. Finally, we highlight some directions in order to enhance the fault tolerance at the MedioGrid application level, both simple and parallel.

1. Introduction

Classic systems approach fault tolerance by either avoiding error conditions, through structured programming and certified component reuse, or using error reduction techniques by enforcing strong testing techniques. In case of grid systems which exhibit dynamic and unforeseen interactions between their heterogeneous components, which reside in different administrative and privileged domains, such approaches are not feasible and insufficient. One of the main reasons is complexity as grid applications have a high degree of asynchronous protocol interactions. By its unpredictable nature, grid systems supply an execution environment where execution guarantees can be hardly satisfied. Another factor which makes complicates execution guarantee is the high execution time as some applications might be running even days or weeks until the results are obtained. In addition, applications might require multiple resources which are located in different administrative domains which can be error prone too. Defects might be exacerbated as well by the service composition, specifically, the fact that a grid service might invoke several other grid services. The failure of one of the invoked service might turn against the caller service, leading to return an error on its caller too. As the service composition protocol is non-deterministic, the potential grid errors are non-deterministic too.

2000 *Mathematics Subject Classification.* 68M14.
Key words and phrases. Grid computing, Fault Tolerance, MedioGrid.

©2007 Babeş-Bolyai University, Cluj-Napoca

All these new introduced condition in grid systems lead us to the idea that for grid systems, one must rely on a more complex fault model. In this paper we analyze a fault tolerance model for distributed systems through the grid perspective, then we are highlighting the Globus approach for the fault tolerance model and finally we evaluate the applicability of fault tolerance techniques on the MedioGrid project.

2. A VIEW ON FAULT TOLERANCE FOR THE GRID

When referring to a fault tolerant systems, we refer to a system which supplies a set of services to its clients, according to a well defined contract, in spite of error presence, through detecting, correcting and eliminating errors, while the systems continues to supply an acceptable set of services [1]. A fault tolerance model highlights possible causes and conditions where errors might appear, with the goal of improving system characteristics do detect and eliminate errors.

The main approach to attack fault tolerance is rollback technique [2], which implies application state logging at a certain time interval and restoring the last stable state in case the application is detected as entering a critical state. The used techniques are either check pointing types [3] where the application state is expected, or logging techniques [4] which implies application message logging and handling. For data grid systems, one of the most common and widespread fault tolerance techniques is provided by replication techniques, at both data provider and computing resources. In the later case, a certain application can be running in parallel on multiple resources and in case of error conditions, computation is continued on the healthy and active resources. Another approach is process migration when the executive state is becoming critical.

Based on [12], we present and discuss the main classes of errors that might appear in the grid systems.

2.1. **Network errors.** Network errors are environmental errors caused by the communication channel and basically refer to package losses on the transmission path or corrupted incoming packages on the receiving path. These errors can be corrected by the network transmission protocol and in cases where no correction can be applied the communication path between the two endpoints is considered broken.

2.2. **Timing errors.** Timing errors are errors that can occur either at the beginning of the communication as a result of the impossibility to establish a connection, or during the communication flow when for example the response time of the caller exceeds the response time expected by the caller. In case of grid systems which exhibit large and variable communication latencies, such timing conditions add a nondeterministic component to the expected approximate time.

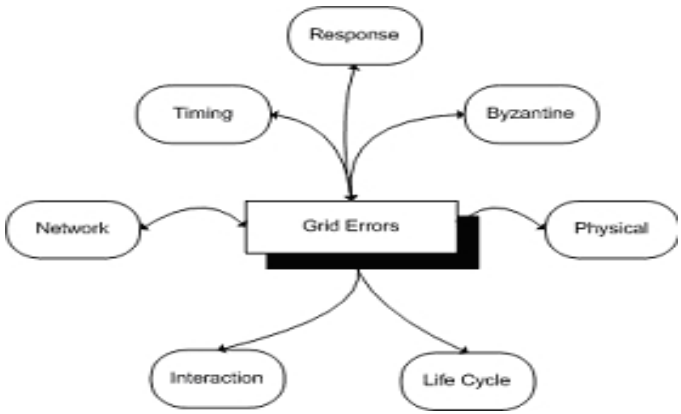


FIGURE 1. Grid Software Error Classes

2.3. **Response errors.** Response errors are caused by a service which returns values outside of the expected boundaries by the caller. In such situations, components have to be able to validate a certain response and to appropriately handle the exceptions. A system that is designed as a state machine, can execute uncontrolled transitions in the state space which can be further propagated to other services as a result of the grid service composition.

2.4. **Byzantine errors.** Byzantine errors are arbitrary errors that could appear during the execution of an application. They refer to catastrophic conditions such as crashes and omission errors. A system entering in a Byzantine state has an undefined behavior which might be caused either by the execution impossibility or erroneous execution, or by arbitrary execution outside of the one specified by design.

2.5. **Physical errors.** Physical errors refer to critical conditions of the physical resources such as processor, memory, storage or communication medium. Such errors have to be detected and corresponding resources be declared as non-functional.

2.6. **Life cycle errors.** Life cycle errors are particular to components which expose services which can expire at a certain moment. They can apply to component versioning as well. An example of this condition is updating a service while its clients expect that the service is working properly according to its previous specification. Service changes could be both syntactical and structural with different implications on the service callers.

2.7. **Interaction errors.** Interaction errors are caused by incompatibilities at the communication protocol stack level, security, workflows or timing. These are the most common errors in large scale grid systems because all these conditions appear

while running the applications and the environmental and interaction states cannot be reproduced during the application testing phases. We expect that for complex grid applications to observe a high probability of interaction error occurrence. Some of these, as for example the ones due to different security levels, could be isolated and eliminated during the testing phases in a high percentage as there is a limited number of calls between virtual organizations.

3. FAULT TOLERANCE IN GLOBUS

Globus Toolkit [4] offers a reference implementation of grid standardized protocols, together with a set of tools and helper services in order to facilitate grid application development and deployment. In terms of the fault tolerance conditions presented in the previous section, one of the most important services are data transfer services, replication and grid execution management services.

The most basic data transfer service which serves as a basis for all other data manipulation services is GridFTP service, which represents an implementation of the grid extended well known FTP protocol. Its extended features for the grid include security level integration, parallel data transfer flows, partial transfers, automatic setting of TCP transfer buffers, transfer flow monitoring and restarting. The most interesting part of the GridFTP component in the context of building fault tolerant data services, is that Globus does not supply a server side library, thus peer-to-peer transfer scenarios cannot be constructed using Globus. In addition, it requires a preinstalled GridFTP server which reduces the degree of resource discovery and automatic replacement in case of physical damage.

Reliable File Transfer service (RFT) provides a Globus service which guarantees a successful file delivery between grid nodes in the presence of failures during a given transfer operation. Transfer status is kept into a database which supplies the necessary information to eventually resume the transfer. Globus uses a SOAP description for the transfer request and the GridFTP protocol to initiate the transfer. RFT offers support for concurrent transfer flows which gives good performance for modest size parallel file transfers.

The Data Replication Service (DRS) is a higher level Globus service which gives support for automatic file replication between multiple sites and it is working as following:

- Grid clients specify a set of files to be replicated to other nodes.
- Each node uses RLS to determine which are the files that are missing locally and where these files are available.
- The missing files are replicated locally by submitting a RFT request to the RFT service.
- After the files are locally replicated they are registered to LRC.

Analyzing data management services, we can notice that the Globus solution is mostly a static solution based on fixed configuration sets of grid components.

Hypothetical situations where a damaged grid node is dynamically replaced by another one reduce to the a priori determination of replacement nodes. Globus does not offer support for peer-to-peer data sharing with automatic discovery of candidate nodes for such a service.

At the execution level, Globus offers an interface to launch and monitor jobs on grid nodes through the GRAM service. The GRAM service takes care for input and output data transfers as well as the security level integration. For execution planning, GRAM supplies a "plug-in" architecture which permits an extension to adapters for local resource schedulers. One can notice that the Globus execution level offers only job state checking during execution. The fault tolerance level has to be assured by the concrete implementation of the resource scheduler. One of the most popular resource schedulers is Condor [6]. Condor offers technical support for fault tolerance by providing job migration and check pointing mechanisms. The fault tolerance level of Condor takes care of restarting monitoring and job submission services in case a local error occurs. In case of network type errors, Condor can restore the communication flow as soon as the nodes become available and can restore the previous state through persistent state logging mechanisms. In cases where the grid nodes which execute the job are detected as faulty nodes, Condor implements a job migration policy to other available nodes which are continuously monitored through a dedicated service.

4. FAULT TOLERANCE IN MEDIOGRID

In case of the MedioGrid system [6] , we consider the main error sources as being caused by communication path corruption, failure of the processing or data units and application specific errors.

Communication path failure can lead to the impossibility to communicate with the parallel processes spawn across the grid to solve a certain environmental problem. It is desired that the entire operation is not restarted as a whole, but only specific computations that are assigned to the processing units who cannot communicate anymore with the rest of the systems. Using a fault tolerant scheduler like Condor [6] removes the impossibility to continue computations on an alternative resource. The GRAM architecture and the interaction with the scheduler are presented in Figure 2. We consider communication path failure as an extreme failure condition that can be compensated by providing alternative communication paths which requires more research into the problem of network design which is not in the scope of the MedioGrid project.

Referring to computing resources and excluding the ones that are managing data, any error at this level is detected and managed by the Condor job scheduler. The support offered by Condor is sufficient to get the job execution to its end, even if nodes are signalling errors.

Data unit failures are mainly addressed by the data model and the level of data replication [8]. The proposed replication solution, which is supported by Globus

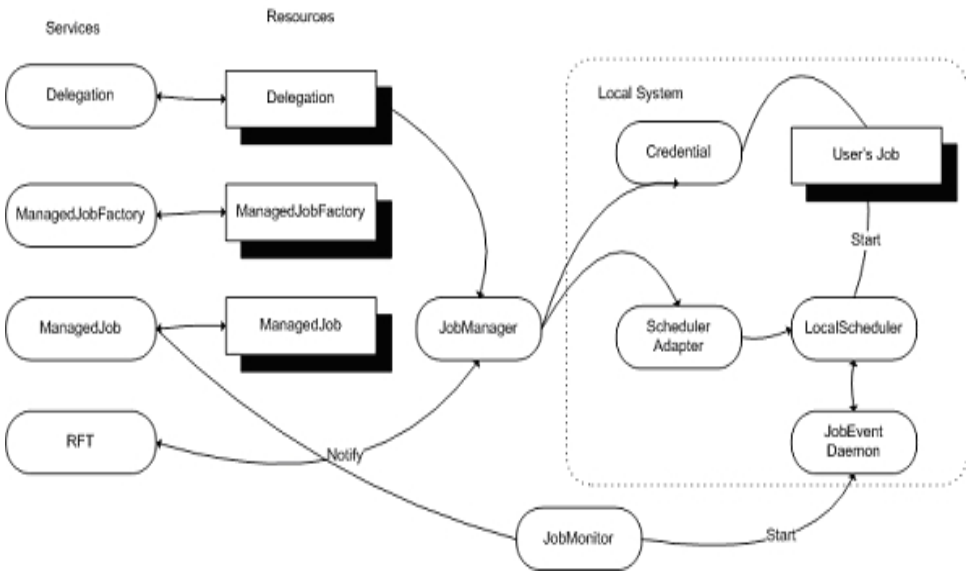


FIGURE 2. GRAM Architecture, as presented in Globus [4]

toolkit, represents a static solution which can handle the failure of up to two data storage units, considering the primary replication group composed of three data resources. A solution which offers a configurable degree of fault tolerance and also addresses the data life cycle is to supply a data management service which guarantees the existence of a file f in the MedioGrid network as being replicated in n locations. At the present form of the MedioGrid system, n is equal to 3 and replication is fixed.

The proposed service achieves a rule based variable replication. The rules determine the replication degree in respect to the data use rate and data age. It is expected that users are accessing more recent data than older ones. These policies come to enforce the data life cycle management in contrast to the current system that keeps data at three fixed locations. Such a service can be implemented based on existing Globus services and existing higher level dynamic replication strategies for grids as the ones presented in [13].

In addition to the availability aspect of data replication in MedioGrid, which has been a goal from the beginning, we would like to extend the dynamic replication scope to address the data life cycle as well. One of the drawbacks of the fixed replication solution is that data is always replicated with the same ratio even if its usage rate is low. Such solution is simply wasting storage that could be used for newer data items.

At the application level, fault tolerance has to components: execution context and the application code itself. At the execution context, the analysis boils down to the resource scheduler and its capabilities to support fault tolerance. As far as the application code is concerned, we distinguish two cases: when the applications runs stand alone and when the application is a parallel application comprising several grid concurrent jobs.

The fault tolerance level for a simple application can be achieved by using grid check-pointing strategies [9]. The idea of the project is to provide transparent check-pointing support for Java applications, by capturing and restoring local and global states through the use of an execution coordinator. There are more particular approaches which aim to offer a certain degree of fault tolerance for Java, which consist basically of a fault tolerant RMI layer. Such layer replaces remote references automatically in case a reference is in the impossibility to execute an operation over a connection. The applicability of such strategies is dependent on the algorithms and decided at this point in time.

In case of a parallel application containing more collaborative or independent tasks which are running in parallel, the Condor job scheduler mentioned in the previous section is being able to launch only one job. The are a series of efforts towards a MPI-like fault tolerant solution [10, 11], but they are still considered immature to be used in a real grid application. Techniques based on GridRPC to obtain a fault tolerant service have main target applications that are running uninterruptible computations, where stopping the calculus for a limited duration is not critical. One the other pole, MedioGrid applications aim to provide the result immediately, which reduces the applicability of these techniques for our project. Of course, in case of some grid operations which have to be completed without any timing constraints, such techniques could be applied for MedioGrid too.

5. CONCLUSIONS

In this paper we have presented a view on fault tolerance for grid systems and we have evaluated the main solutions and concepts provided by the Globus toolkit to construct fault tolerant grid applications. As a result of the analysis, we concluded that Globus provides limited support for fault tolerant services, both at data management and task execution, but provides the necessary basic concepts to build higher level services. One of the major drawbacks of the Globus solution is the built-in static configurations which limits dynamic service construction using Globus components. Based on our analysis of the main grid error classes, we have assessed the MedioGrid system in terms of fault tolerance and we suggested a dynamic data replication extension based on configurable life-cycle policies. Depending on the MedioGrid applications, we have given a few directions towards adopting supplemental fault tolerance levels in terms of parallel applications.

References

[1] Avizienis, A.,"The N-version Approach to Fault-Tolerant Software" - IEEE Transactions on Software Engineering - vol. 11 1985

[2] Manivannan, D., Singhal, M.,"Quasi-synchronous checkpointing: Models, characterization, and classification". In: IEEE Transactions on Parallel and Distributed Systems. Volume 10. (1999) 703-713

[3] Alvisi, L., Marzullo, K., "Message logging: Pessimistic, optimistic, causal, and optimal". Software Engineering 24 (1998) 149-159

[4] Globus toolkit homepage, http://www.globus.org, Globus Alliance 2006

[5] Condor homepage, http://www.cs.wisc.edu/condor/

[6] Ordean, M., Melenti, C., and Gorgan, D., "Mediogrid system in meteorological and environment applications". International Conference on Advances in the Internet, Processing, Systems and Interdisciplinary Research, IPSI - 2005 Amalfi, Italy, pp: 203-207, ISBN: 86-7466-117-3, 2005

[7] Muresan, O. and Gorgan, D., "Arhitectura retelei MedioGrid". Atelier de Lucru MEDIOGRID vol 1, ISBN: 973-713-090-1, Ed MEDIAMIRA Cluj-Napoca, 2006.

[8] Colesa, A., Ignat, I., Opris, R., "Providing High Data Availability in MEDIOGRID", 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, September 26-29, 2006

[9] Stone, N., Simmel, D., and Kielmann, T., "GWD-I: An architecture for grid checkpoint recovery services and a GridCPR API". Grid Checkpoint Recovery Working Group Draft 3.0, Global Grid Forum, http://gridcpr.psc.edu/GGF/docs/draft-ggf-gridcpr-Architecture-2.0.pdf, May 2004.

[10] Graham, E. F., and et al., "HARNESS and fault tolerant MPI", Parallel Computing, vol. 27, pp. 1479-1496, 2001.

[11] Bosilca, G., and et al., "Mpich-v: Toward a scalable fault tolerant mpi for volatile nodes", in Proceedings of Supercomputing, 2002.

[12]]Townend, P., Xu, J., "Fault Tolerance within Grid environment", Proceedings of AHM2003, page 272, 2003

[13] Ranganathan, K., Foster, I.T., "Identifying Dynamic Replication Strategies for a High-Performance Data Grid", Proceedings of the Second International Workshop on Grid Computing Vol. 2242, pages: 75-86, 2001

(1) Computer Science and Engineering Department, "Politehnica" University of Timisoara, V. Parvan street, no. 2, 30023, Timisoara, Romania
E-mail address: dacian@cs.utt.ro

(2) Computer Science and Engineering Department, "Politehnica" University of Timisoara, V. Parvan street, no. 2, 30023, Timisoara, Romania
E-mail address: vcretu@cs.utt.ro

(3) Computer Science and Engineering Department, "Politehnica" University of Timisoara, V. Parvan street, no. 2, 30023, Timisoara, Romania
E-mail address: horia@cs.utt.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 252–260

A NEW GRAPH-BASED APPROACH IN ASPECT MINING

GABRIELA ŞERBAN[1] AND GRIGORETA SOFIA COJOCAR[2]

Abstract. *Aspect mining* is a process that tries to identify crosscutting concerns in existing software systems. The goal is to refactor the existing systems to use aspect oriented programming ([3]), in order to make them easier to maintain and to evolve. This paper aims at presenting a new graph-based approach in aspect mining. We define the problem of identifying the crosscutting concerns as a search problem in a *graph* and we introduce *GRAM* algorithm for solving this problem. We evaluate based on some quality measures the results obtained by applying *GRAM* algorithm from the aspect mining point of view. The proposed approach is compared with a clustering approach in aspect mining ([5]) and a case study is also reported.

1. Introduction

1.1. **Aspect Mining.** *Separation of concerns* ([1]) is a very important principle of software engineering that, in its most general form, refers to the ability to identify, encapsulate and manipulate those parts of software that are relevant to a particular concept, goal, or purpose. Some of the benefits of a good separation of concerns are reduced software complexity, improved comprehensability, limited impact of change, easy evolution and reuse.

Aspect Oriented Programming (AOP) ([3]) provides means to encapsulate concerns which cannot be modularized using traditional programming techniques. These concerns are called *crosscutting concerns*. Logging and exception handling are well known examples of crosscutting concerns. Aspect oriented paradigm offers a powerful technology for supporting the separation of crosscutting concerns. Such a concern is explicitly specified as an *aspect*. Aspects encapsulate the implementation of a crosscutting concern. A special tool, called *weaver*, integrates a number of aspects to obtain the final software system. *Aspect mining* is a relatively new research direction that tries to identify crosscutting concerns in already developed software systems, without using AOP. The goal is to identify them and then to refactor them to aspects, to achieve a system that can be easily understood, maintained and modified.

2000 *Mathematics Subject Classification.* 68N99, 68R10.
Key words and phrases. Software Engineering, Aspect Mining, Graph.

©2007 Babeş-Bolyai University, Cluj-Napoca

Crosscutting concerns in non AO systems have two symptoms: *code scattering* and *code tangling*. *Code scattering* means that the code that implements a crosscutting concern is spread across the system, and *code tangling* means that the code that implements some concern is mixed with code from other (crosscutting) concerns.

1.2. **Related Work.** Several approaches have been considered for aspect mining until now. Some approaches use clone detection techniques to identify duplicate code that might indicate the presence of crosscutting concerns ([12], [13]). Another approach uses metrics to identify crosscutting concerns that have the scattering symptom ([8]). There are also two approaches that use dynamic analysis to discover crosscutting concerns: one that analyzes the program traces to discover recurring execution relations ([11]), and one that uses formal concept analysis to analyze the execution traces ([14]).

A few aspect mining techniques proposed in the literature use *clustering* in order to identify crosscutting concerns ([5], [6], [7]). In [6] a vector space model based clustering approach in aspect mining is proposed. This approach is improved in [5], by defining a new *k-means* based clustering algorithm in aspect mining (kAM). In [7] the methods are clustered based on their names, and then the user can navigate among the clusters, visualize the source code of the methods and identify the crosscutting concerns.

In this paper we propose a new graph-based approach, as an alternative to the clustering approach in aspect mining.

The paper is structured as follows. A theoretical model on which we base our approach is introduced in Section 2. Section 3 presents a new graph based approach in aspect mining. An experimental evaluation of our approach, based on some quality measures, is presented in Section 4. The obtained results are compared with the ones obtained by applying kAM algorithm ([5]). Some conclusions and further work are outlined in Section 5.

2. Theoretical Model

In this section we present the problem of identifying *crosscutting concerns* as a problem of identifying a partition of a software system.

Let $M = \{m_1, m_2, ..., m_n\}$ be the software system, where $m_i, 1 \leq i \leq n$ is a method of the system. We denote by n ($|M|$) the number of methods in the system.

We consider a crosscutting concern as a set of methods $C \subset M$, $C = \{c_1, c_2, \ldots, c_{cn}\}$, methods that implement this concern. The number of methods in the crosscutting concern C is $cn = |C|$. Let $CCC = \{C_1, C_2, ..., C_q\}$ be the set of all crosscutting concerns that exist in the system M. The number of crosscutting concerns in the system M is $q = |CCC|$. Let $NCCC = M \setminus (\bigcup_{i=1}^{q} C_i)$ be the set

of methods from the system M, methods that do not implement any crosscutting concerns.

Definition 1. ***Partition of a software system*** M**.**
The set $\mathcal{K} = \{K_1, K_2, ..., K_p\}$ *is called a* ***partition*** *of the system* $M = \{m_1, m_2, \ldots, m_n\}$ *iff* $1 \leq p \leq n$, $K_i \subseteq M, K_i \neq \emptyset, \forall 1 \leq i \leq p$, $M = \bigcup_{i=1}^{p} K_i$ *and* $K_i \cap K_j = \emptyset$, $\forall i, j, 1 \leq i, j \leq p, i \neq j$.

In the following we will refer to K_i as the i-th *cluster* of $\mathcal{K}$.

In fact, the problem of aspect mining can be viewed as the problem of finding a partition $\mathcal{K}$ of the system M. If the result of an AM technique is a *partition* of the software system, we will call it **partitioning aspect mining technique**.

We propose the following steps for identifying the crosscutting concerns that have the scattered code symptom:

- **Computation** - Computation of the set of methods in the selected source code, and computation of the attribute set values, for each method in the set.
- **Filtering** - Methods belonging to some data structures classes (like *ArrayList, Vector*) are eliminated. We also eliminate the methods belonging to some built-in classes like *String, StringBuffer, StringBuilder, etc.*
- **Grouping** - The remaining set of methods is grouped in order to obtain a partition of the software system M (in our approach using *GRAM* algorithm).
- **Analysis** - A part of the obtained clusters are analyzed in order to discover which clusters contain methods belonging to crosscutting concerns.

We mention that at the **Grouping** step, a partition of the software system can be obtained using a clustering algorithm ([5]) in aspect mining, or using *GRAM* algorithm, that will be introduced in the next section.

3. A New Graph-Based Approach In Aspect Mining

In this section we present the problem of obtaining a *partition* (Definition 1) of a software system as a search problem in a *graph*. This graph based approach is, in fact, a method to identify the clusters in the system and can be viewed as an alternative to a *clustering* algorithm in aspect mining ([5]).

In our approach, the objects to be grouped (clustered) are the methods from the software system: $m_1, m_2, \ldots, m_n$. The methods belong to the application classes or are called from the application classes.

Based on the vector space model, we will consider that the vector associated with the method m is $\{FIV, CC\}$, where FIV is the fan-in value ([8]) of m (the number of methods that call m) and CC is the number of calling classes for m.

In our approach, we will consider the *Euclidian distance* between methods as a measure of dissimilarity between them.

After a partition of the software system is determined using a **partitioning aspect mining technique**, the clusters are sorted by the average distance from the point 0_2 in descending order, where 0_2 is the two-dimensional vector with each component 0 (in our case two is the dimension of the vector space model). Then, we analyze the clusters whose distance from 0_2 point is greater than a given threshold.

3.1. **The Graph Of Concerns.** In this section we introduce the concept of *graph of concerns* and auxiliary definitions needed to define our search problem. The concept of *graph of concerns* introduced below is different from the concept of *concerns graph* defined in [2] by Robillard and Murphy and it is used in a different context.

We mention that the idea of constructing the *graph of concerns* is specific to aspect mining and will be explained later.

Definition 2. *Let $M = \{m_1, m_2, \ldots, m_n\}$ be a software system and d_E the* Euclidian distance *metric between methods in a multidimensional space. The* ***graph of concerns*** *corresponding to the software system M, denoted by $\mathcal{GC}_M$, is an undirected graph defined as follows: $\mathcal{GC}_M = (\mathcal{V}, \mathcal{E})$, where:*

- *The set $\mathcal{V}$ of vertices is the set of methods from the software system, i.e., $\mathcal{V} = \{m_1, m_2, \ldots, m_n\}$.*
- *The set $\mathcal{E}$ of edges is $\mathcal{E} = \{(v_1, v_2) \,|\, v_1, v_2 \in \mathcal{V},\ v_1 \neq v_2,\ \ d_E(v_1, v_2) \leq$ **distMin**$\}$, where **distMin** is a given threshold.*

We have chosen the value 1 for the threshold *distMin*. The reason for choosing this value is the following: if the distance between two methods m_i and m_j is less or equal to 1, we consider that they are similar enough to be placed in the same (crosscutting) concern. We mention that, from the aspect mining point of view, using *Euclidian distance* as metric and the vector space model proposed above, the value 1 for $distMin$ makes the difference between a crosscutting concern and a non-crosscutting one.

In Definition 3 below we will define the problem of computing a partition of the software system M.

Definition 3. *Let $M = \{m_1, m_2, \ldots, m_n\}$ be a software system, d_E (*Euclidian distance*) the metric between methods in a multidimensional space and $\mathcal{GC}_M$ the corresponding graph of concerns* (Definition 2)*. We define the problem of computing a partition $\mathcal{K} = \{K_1, K_2, ..., K_p\}$ of M as the problem of computing the connected components of $\mathcal{GC}_M$.*

3.2. ***GRAM* Algorithm.** In this subsection we briefly describe *GRAM* algorithm for determining a *partition* $\mathcal{K}$ of a software system M. This algorithm will be used in the **Grouping** step (Section 2) for identification of crosscutting concerns.

Let us consider a software system $M = \{m_1, m_2, \ldots, m_n\}$ and the *Euclidian distance* d_E between methods in a multidimensional space, and the problem introduced in Definition 3.

The main steps of *GRAM* algorithm are:

(i) Create the *graph of concerns*, $\mathcal{GC}_M$, as shown in Definition 2. We mention that the threshold $distMin$ used for creating the edges in the graph is chosen to be 1. The reason for this choice was explained above.
(ii) Determine the connected components of $\mathcal{GC}_M$. These components give a partition $\mathcal{K}$ of the software system M.

3.3. **Example.** In the following, we present a small example that shows how methods are grouped in clusters by *GRAM* algorithm. If we have the classes shown in Table 1, the values of the attribute set are presented in Table 2 and the corresponding graph of concerns is shown in Figure 1. The obtained clusters are given in Table 3.

```
public class A {
   private L l;
   public A(){l=new L(); methB();}
   public void methA(){ l.meth(); methB();}
   public void methB(){ l.meth();}
}
public class L {
   public L(){}
   public void meth(){}
}
public class B {
   public B(){}
   public void methC(L l){ l.meth();}
   public void methD(A a){a.methA();}
}
```

TABLE 1. Code example.

Method	FIV	CC
A.A	0	0
A.methA	1	1
A.methB	2	1
B.B	0	0
B.methC	0	0
B.methD	0	0
L.L	1	1
L.meth	3	2

TABLE 2. Attribute values.

Cluster	Methods
C1	{ L.meth }
C2	{A.methA, A.methB, L.L }
C3	{ A.A, B.B, B.methC, B.methD }

TABLE 3. The obtained clusters.

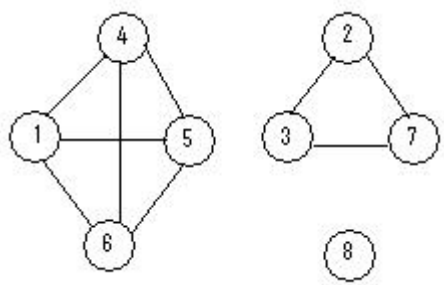


FIGURE 1. Graph of concerns.

4. Experimental Evaluation

In order to evaluate the results of *GRAM* algorithm from the aspect mining point of view, we use a set of quality measures defined in [4].

These measures will be applied on a case study (Subsection 4.2). The obtained results will be reported in Subsection 4.2. Based on the obtained results, *GRAM* algorithm will be compared with *kAM* algorithm proposed in [5].

4.1. **Quality Measures.** In this subsection we present three quality measures. These measures (*DIV*, *ACT* and *PAN*) evaluate a partition from the aspect mining point of view.

DIV is a measure already defined in [4], but *ACT* and *PAN* are newly defined. All these measures evaluate a partition of a software system from the aspect mining point of view.

In the following, let us consider a partition $\mathcal{K} = \{K_1, \ldots, K_p\}$ of a software system $M = \{m_1, m_2, \ldots, m_n\}$ and $CCC = \{C_1, C_2, ..., C_q\}$ the set of all crosscutting concerns from M (Section 2). The partition $\mathcal{K}$ can be obtained using *GRAM* algorithm or using a clustering algorithm, like *kAM* ([5]).

$DIV(CCC, \mathcal{K})$ is introduced in [4] and defines the degree to which each cluster contains methods from different crosscutting concerns or methods from other concerns. $DIV(CCC, \mathcal{K})$ takes values in $[0, 1]$ and larger values for DIV indicate better partitions with respect to CCC, meaning that DIV has to be maximized.

Definition 4. ***ACcuracy of a partitioning based aspect mining Technique - ACT.***

Let $\mathcal{T}$ be a partitioning aspect mining technique.

The accuracy of $\mathcal{T}$ with respect to a partition $\mathcal{K}$ and the set CCC, denoted by $ACT(CCC, \mathcal{K}, \mathcal{T})$, is defined as:

$$ACT(CCC, \mathcal{K}, \mathcal{T}) = \frac{1}{q} \sum_{i=1}^{q} act(C_i, \mathcal{K}, \mathcal{T}).$$

$act(C, \mathcal{K}, \mathcal{T})$ is the accuracy of $\mathcal{T}$ with respect to the crosscutting concern C, and is defined as:

$$act(C, \mathcal{K}, \mathcal{T}) = \sum_{j=1}^{r} \frac{|C \cap K_j|}{|C|}$$

where r $(1 \leq r \leq p)$ is the last cluster analyzed by $\mathcal{T}$.

For a given crosscutting concern $C \in CCC$, $act(C, \mathcal{K}, \mathcal{T})$ defines the percentage of methods from C that were discovered by $\mathcal{T}$.

In all partitioning aspect mining techniques, only a part of the clusters are analyzed, meaning that some crosscutting concerns or parts of them may be missed.

Based on Definition 4, $ACT(CCC, \mathcal{K}, \mathcal{T}) \in (0, 1]$. Larger values for ACT indicate better partitions with respect to CCC, meaning that ACT has to be maximized.

Definition 5. ***Percentage of ANalyzed methods for a partition - PAN.***

Let us consider that the partition $\mathcal{K}$ is analyzed in the following order: $K_1, K_2, \ldots, K_p$.

The percentage of analyzed methods for a partition $\mathcal{K}$ with respect to the set CCC, denoted by $PAN(CCC, \mathcal{K})$, is defined as:

$$PAN(CCC, \mathcal{K}) = \frac{1}{q} \sum_{i=1}^{q} pan(C_i, \mathcal{K}).$$

$pan(C, \mathcal{K})$ is the percentage of the methods that need to be analyzed in the partition $\mathcal{K}$ in order to discover the crosscutting concern C, and is defined as:

$$pan(C, \mathcal{K}) = \frac{1}{n} \sum_{j=1}^{s} |K_j|$$

where $s = max\{t \,|\, 1 \leq t \leq p \text{ and } K_t \cap C \neq \emptyset\}$ is the index of the last cluster in the partition $\mathcal{K}$ that contains methods from C.

$PAN(CCC, \mathcal{K})$ defines the percentage of the number of methods that need to be analyzed in the partition in order to discover all crosscutting concerns that are in the system M. We consider that a crosscutting concern was discovered when all the methods that implement it were analyzed.

Based on Definition 5, it can be proved that $PAN(CCC, \mathcal{K}) \in (0, 1]$. Smaller values for PAN indicate shorter time for analysis, meaning that PAN has to be minimized.

4.2. **Results.** In order to evaluate the results of *GRAM* algorithm, we consider as case study JHotDraw, version 5.2 ([9]).

This case study is a Java GUI framework for technical and structured graphics, developed by Erich Gamma and Thomas Eggenschwiler, as a design exercise for using design patterns. It consists of **190** classes and **1963** methods.

In this subsection we present the results obtained after applying *GRAM* algorithm described in Subsection 3.2, for the vector space model presented in Section 3, with respect to the quality measures described in Subsection 4.1, for the case study presented above.

The results obtained by *GRAM* are compared with the results obtained by *kAM* algorithm proposed in [5].

In Table 1 we present the comparative results.

Algorithm	DIV	ACT	PAN
GRAM	0.857	0.299	0.346
kAM	0.842	0.278	0.361

TABLE 4. The values of the quality measures for JHotDraw case study.

From Table 1 we observe, based on the properties of the quality measures defined in the above subsection, that *GRAM* algorithm provides **better** results from the aspect mining point of view, than **kAM** algorithm.

In our view, the vector space model has a significant influence on the obtained results. We are working on improving the vector space model in order to handle the tangling code symptom, too.

So far no comparison between existing aspect mining techniques was reported in the literature. No comparison between *GRAM* and other similar approaches is provided for the following reasons:

- some techniques are dynamic and they depend on the data used during executions ([11], [14]);
- for the static techniques ([8], [7]) only parts of the results are publicly available;
- there is no case study used by all these techniques.

5. Conclusions and Future Work

We have presented in this paper a new *graph-based approach* in aspect mining. For this purpose we have introduced *GRAM* algorithm, that identifies a partition of a software system. This partition is analyzed in order to identify the crosscutting concerns from the system. In order to evaluate the obtained results from the aspect mining point of view, we have used a set of quality measures. Based on these measures, we showed that *GRAM* algorithm provides **better** partitions than *kAM* algorithm (previously introduced in [5]).

Further work can be done in the following directions:

- To apply this approach for other case studies, like JEdit ([10]).
- To compare the results provided by *GRAM* with the results of other approaches in aspect mining.
- To identify a choice for the threshold $distMin$ that will lead to better results.

- To improve the results obtained by *GRAM*, by improving the vector space model used.

References

[1] David. L. Parnas. On The Criteria To Be Used in Decomposing Systems Into Modules. *Communications of the ACM*, 15(12), 1972, pp. 1053–1058.

[2] Robillard, M.P., Murphy, G.C.: Concern graphs: finding and describing concerns using structural program dependencies. In: *Proceedings of the 24th International Conference on Software Engineering.* Orlando, Florida (2002) 406–416

[3] Kiczales, G., Lamping, J., Menhdhekar, A., Maeda, C., Lopes, C., Loingtier, J.M., Irwin, J.: Aspect-Oriented Programming. In: *Proceedings European Conference on Object-Oriented Programming.* Volume 1241. Springer-Verlag (1997) 220–242

[4] Moldovan, G.S., Serban, G.: Quality Measures for Evaluating the Results of Clustering Based Aspect Mining Techniques. In: *Proceedings of Towards Evaluation of Aspect Mining(TEAM), ECOOP.* (2006) 13–16

[5] Serban, G., Moldovan, G.S.: A new k-means based clustering algorithm in aspect mining. In: *Proceedings of 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'06).* (2006) 60–64

[6] Moldovan, G.S., Serban, G.: Aspect Mining using a Vector-Space Model Based Clustering Approach. In: *Proceedings of Linking Aspect Technology and Evolution (LATE) Workshop.* (2006) 36–40

[7] Shepherd, D., Pollock, L.: Interfaces, Aspects, and Views. In: *Proceedings of Linking Aspect Technology and Evolution (LATE) Workshop.* (2005)

[8] Marin, M., van, A., Deursen, Moonen, L.: Identifying Aspects Using Fan-in Analysis. In: *Proceedings of the 11th Working Conference on Reverse Engineering (WCRE2004).* IEEE Computer Society (2004) 132–141

[9] JHotDraw Project: http://sourceforge.net/projects/jhotdraw (1997)

[10] jEdit Programmer's Text Editor: http://www.jedit.org (2002)

[11] Breu, S., Krinke, J.: Aspect Mining using Event Traces. In: *Proceedings of International Conference on Automated Software Engineering.* (2004) 310–315

[12] Bruntink, M., van Deursen, A., van Engelen, R., Tourwé, T.: An Evaluation of Clone Detection Techniques for Identifying Crosscutting Concerns. In: *Proceedings International Conference on Software Maintenance(ICSM 2004).* IEEE Computer Society (2004)

[13] Morales, O.A.M.: Aspect Mining Using Clone Detection. Master's thesis, Delft University of Technology, The Netherlands. (2004)

[14] Tonella, P., Ceccato, M.: Aspect Mining through the Formal Concept Analysis of Execution Traces. In: *Proceedings of the IEEE Eleventh Working Conference on Reverse Engineering (WCRE 2004).* (2004) 112–121

[1] Department of Computer Science, Babeş-Bolyai University, 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: gabis@cs.ubbcluj.ro

[2] Department of Computer Science, Babeş-Bolyai University, 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: grigo@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 261–268

INTRODUCING DATA-DISTRIBUTIONS INTO *POWERLIST* THEORY

VIRGINIA NICULESCU[(1)]

Abstract. *PowerList* theory is well suited to express recursive, data-parallel algorithms. Its abstractness is very high and assures simple and correct design of parallel programs. We try to reconcile this high level of abstraction with performance by introducing data-distributions into this theory. One advantage of formally introducing distributions is that it allows us to evaluate costs, depending on the number of available processors, which is considered as a parameter. Also, the analysis of the possible distributions for a certain function may lead to an improvement in the design decisions. Another important advantage is that after the introduction of data-distributions, mappings on real parallel architectures with limited number of procesing elements could be analyzed.

1. Introduction

*PowerList*s are data structures introduced by Misra [3], which can be successfully used in a simple and provable correct, functional description of parallel programs, that are *divide and conquer* in nature. They allow working at a high level of abstraction, especially because the index notations are not used. To assure methods that verify the correctness of the parallel programs, algebras and structural induction principles are defined on these data structures. Based on the structural induction principle, functions and operators, which represent the parallel programs, are defined. Generally, unbounded parallelism (the number of processes is not limited) is analyzed using these structures. Still, the most practical approach of bounded parallelism may be introduced, and so, the distributions, too.

Mappings on hypercubes have been analyzed for the programs specified based on these notations [3, 2]; they are based on Gray code. The analysis assumes that the hypercube has more nodes that the lists which are mapped onto, and this is an unrealistic assumption.

2000 *Mathematics Subject Classification.* 65Y05, 68Q85.

Key words and phrases. parallel computation, abstraction, design, distribution, data-structures.

©2007 Babeş-Bolyai University, Cluj-Napoca

The PowerList notation has been proved to be a very elegant way to specify parallel algorithms and prove their correctness. The main advantage of this model is that offers a simple, formal and elegant way to prove correctness. By formally introducing data distribution in this model, we enhance it with the possibility of formally evaluating costs for the case of bounded parallelism. In this way the high level of abstraction of these theories is reconcile with the performance.

2. *PowerList* Theory

A *PowerList* is a linear data structure whose elements are all of the same type. The length of a *PowerList* data structure is a power of two. The type constructor for *PowerList* is:

$$PowerList : Type \times \mathbb{N} \to Type \tag{1}$$

and so, a *PowerList* l with 2^n elements of type X is specified by $PowerList.X.n$ $(n = loglen.l)$. A *PowerList* with a single element a is called *singleton*, and is denoted by $[a]$. If two *PowerList* structures have the same length and elements of the same type, they are called *similar*s.

Two *similar PowerList*s can be combined into a *PowerList* data structure with double length, in two different ways:

- using the operator *tie* $p \mid q$; the result contains elements from p followed by elements from q
- using the operator *zip* $p \natural q$; the result contains elements from p and q, alternatively taken.

Therefore, the constructor operators for *PowerList* are:

$$\begin{array}{lcl} [.] & : & X \to PowerList.X.0 \\ .|. & : & PowerList.X.n \times PowerList.X.n \to PowerList.X.(n+1) \\ .\natural. & : & PowerList.X.n \times PowerList.X.n \to PowerList.X.(n+1). \end{array} \tag{2}$$

Functions are defined based on the structural induction principle. For example, the high order function *map*, which applies a scalar function to each element of a *PowerList* is defined as follows:

$$\begin{array}{l} map.f.[a] = [f.a] \\ map.f.(p \natural q) = map.f.p \natural map.f.q \end{array} \tag{3}$$

Another example is the function $flat$ that is applied to a PowerList with elements which are in turn PowerLists, and it returns a simple PowerList:

$$\begin{array}{l} flat.[l] = l \\ flat.(p \natural q) = flat.p \natural flat.q \quad \text{or } flat.(p \mid q) = flat.p \mid flat.q \end{array} \tag{4}$$

The reduction of the list to an element by using an associative operator $\oplus$ is defined by:

$$\begin{array}{l} red.\oplus.[a] = a \\ red.\oplus.(p|q) = red.\oplus.p \oplus red.\oplus.q \end{array} \tag{5}$$

Binary associative operators on scalar types can be extended to *PowerList*.

3. Distributions

The ideal method to implement parallel programs described with *PowerLists* is to consider that any application of the operators *tie* or *zip* as deconstructors, leads to two new processes running in parallel, or, at least, to assume that for each element of the list there is a corresponding process. That means that the number of processes grows linearly with the size of the data. In this ideal situation, the time-complexity is usually logarithmic (if the combination step complexity is a constant), depending on *loglen* of the input list.

A more practical approach is to consider a bounded number of processes n_p. In this case we have to transform de input list, such that no more than n_p processes are created. This transformation of the input list corresponds to a data distribution.

Definition 1. *$D = (\delta, A, B)$ is called a (one-dimensional) distribution if A and B are finite sets, and δ is a mapping from A to B; set A specifies the set of data objects (an array with n elements that represent the indices of data objects), and the set B specifies the set of processes, which is usually $\overline{p}$. The function δ assigns each index $i (0 \leq i < n)$, and its corresponding element, to a process number* [4].

One advantage of *PowerList* theory is that it does not need to use indices, and this simplifies very much reasoning and correctness proving. So, we will introduce distributions not as in the definition above, but as functions on these special data structures.

The distribution will transform the list into a list with n_p elements, which are in turn sublists; each sublist is considered to be assigned to a process.

3.1. *PowerList* Distributions.

We consider *PowerList* data structures with elements of a certain type X, and with length such that $loglen = n$. The number of processes is assumed to be limited to $n_p = 2^p$ $(p \leq n)$.

Two types of distributions – linear and cyclic, which are well-known distributions, may be considered. These correspond in our case to the operators *tie* and *zip*. Distributions are defined as *PowerList* functions, so definitions corresponding to the base case and to the inductive step have to be specified:

- linear

$$\begin{aligned} & distr^l.p.(u|v) = distr^l.(p-1).u \mid distr^l.(p-1).v \\ & distr^l.0.l = [l] \\ & distr^l.p.x = [x], \text{ if } loglen.x < p. \end{aligned} \tag{6}$$

- cyclic

$$\begin{aligned} & distr^c.p.(u\natural v) = distr^c.(p-1).u \natural distr^c.(p-1).v \\ & distr^c.0.l = [l] \\ & distr^c.p.x = [x], \text{ if } loglen.x < p. \end{aligned} \tag{7}$$

The base cases, transform a list l into a singleton, which has the list $[l]$ as its unique element.

3.1.1. *Properties.* If we consider $u \in PowerList.X.n$, then $distr.n.u = \overline{u}$, where $\overline{u}$ is obtained from the list u by transforming each of its elements into a singleton list.

Also, we have the trivial property $distr.0.u = [u]$.

The result of the application of a distribution $distr.p$ to a list $l \in PowerList.X.n$, $n \geq p$ is a list that has 2^p elements each of these being a list with 2^{n-p} elements of type X.

The properties are true for both linear and cyclic distributions.

3.1.2. *Function Transformation.* We consider a function f defined on $PowerList.X.n$ based on operator *tie* with the property that

$$f.(u|v) = \Phi(f.x_0, f.x_1, \ldots, f.x_m, u, v), \tag{8}$$

where $x_i \in PowerLists.X.k, k = loglen\ n - 1$, and $x_i = e_i.u.v, \forall i : 0 \leq i \leq m$, and e_i and Φ are expressions that may use scalar functions and extended operators on PowerLists. If the function definition Φ is more complex and uses other functions on *PowerList*s, then these functions have to be transformed first, and the considered function after that.

A scalar function f has zero or more scalars as arguments, and its value is a scalar. The function f is easily extended to a PowerList by applying f "pointwise" to the elements of the PowerList. A scalar function that operates on two arguments could be seen as an infix operator, and it also could be extended to PowerLists.

The extensions of the scalar functions on PowerLists could be defined either using the operator tie or zip. Some properties of these functions could be find in [3]. For the sake of the clarity, we will introduce the notation f^1 that specifies the corresponding extended function on PowerLists of the scalar function f defined on the scalar type X. For the case of one argument the definition is:

$$\begin{array}{l} f^1 : PowerList.X.n \to PowerList.X.n \\ f^1.[a] = [f.a] \\ f^1.(p|q) = f^1.p|f^1.q \ \text{ or } f^1.(p\natural q) = f^1.p\natural f^1.q \end{array} \tag{9}$$

Further, f^2 (which is the notation for the extension of f on PowerLists with elements which are in turn PowerLists) could be defined:

$$\begin{array}{l} f^2 : PowerList.(PowerList.X.m).n \to PowerList.(PowerList.X.m).n \\ f^2.[a] = [f^1.a] \\ f^2.(p|q) = f^2.p|f^2.q \ \text{ or } f^2.(p\natural q) = f^2.p\natural f^2.q \end{array} \tag{10}$$

We intend to show that

$$f.u = flat \circ f^p.(dist^l.p.u),$$

where

$$
\begin{array}{ll}
 & f^p.(u|v) = \Phi^2(f^p.x_0, f^p.x_1, \ldots, f^p.x_m, u, v) \\
(11) & f^p.[l] = [f^s.l] \\
 & f^s.u = f.u
\end{array}
$$

Function f^p corresponds to parallel execution, and function f^s corresponds to sequential execution.

Lemma 1. *Given a scalar function $f : X \to X$, and a distribution function distr.p, defined on PowerList.X.n, then the following equality is true*

$$dist.p \circ f^1 = f^2 \circ dist.p \tag{12}$$

Proof. To prove this lemma we use induction on p.

We prove the case of the linear distribution, but the case of the cyclic distribution $dist^c.p$ is similar.

Base case$(p = 0)$

$$
\begin{array}{ll}
 & f^2.(dist^l.0.u) \\
= & \{p = 0 \Rightarrow dist^l.p.u = [u]\} \\
 & f^2.[u] \\
= & \{f^2 \text{ definition}\} \\
 & [f^1.u] \\
= & \{distr^l.0 \text{ definition}\} \\
 & distr^l.0.(f^1.u)
\end{array}
$$

Inductive step

$$
\begin{array}{ll}
 & f^2.(dist^l.p.(u|v)) \\
= & \{ \text{ definition of } distr^l\} \\
 & f^2.(dist^l.(p-1).u|dist^l.(p-1).v) \\
= & \{f^2 \text{ definition}\} \\
 & f^2.(dist^l.(p-1).u)|f^2.(dist^l.(p-1).v)) \\
= & \{\text{induction assumption}\} \\
 & distr^l.(p-1).(f^1.u)|distr^l.(p-1).(f^1.u) \\
= & \{distr^l \text{ definition}\} \\
 & distr^l.p.(f^1.u|f^1.v) \\
= & \{f^1 \text{ definition}\} \\
 & distr^l.p.(f^1.(u|v))
\end{array}
$$

□

The previous result is naturally extended to scalar functions with more arguments, such as infix operators.

Scalar binary associative operators $(\oplus)$, could be also extended on PowerLists as reduction operators – $red(\oplus)$. They transform a PowerList into a scalar. For

them, similar extensions as for scalar functions may be done.

$$\begin{array}{l} red^1(\oplus) : PowerList.X.m \to X \\ red^1(\oplus).[a] = a \\ red^1(\oplus).(p|q) = red^1(\oplus).p \oplus red^1(\oplus).q \end{array} \tag{13}$$

$$\begin{array}{l} red^2(\oplus) : PowerList.(PowerList.X.m).n \to PowerList.X.0 \\ red^2(\oplus).[l] = [red^1(\oplus).l] \\ red^2(\oplus).(p|q) = [red^1(\oplus).\big(red^2(\oplus).p|red^2(\oplus).q\big)] \end{array} \tag{14}$$

Also, a similar property in relation to distributions is obtained:

$$distr.p \circ red^1(\oplus) = red^2(\oplus) \circ distr.p \tag{15}$$

Theorem 1. *Given a function f defined on $PowerList.X.n$ as in Eq. 8, a corresponding distribution $distr^l.p, (p \leq n)$, and a function f^p defined as in Eq. 11, then the following equality is true*

$$f = flat \circ (f^p \circ dist^l.p) \tag{16}$$

Proof. We will proof the following equation

$$\begin{array}{c} (distr^l.p. \circ f).u = (f^p \circ dist^l.p).u \\ \text{for any } u \in PowerList.X.n \end{array} \tag{17}$$

which implies the equation 16. To prove this, we use again induction on p.

Base case$(p = 0)$

$$\begin{array}{ll} & f^p.(dist^l.0.u) \\ = & \{p = 0 \Rightarrow dist^l.p.u = [u]\} \\ & f^p.[u] \\ = & \{f^p \text{ definition}\} \\ & [f^s.u] \\ = & \{f^s \text{ definition}\} \\ & distr^l.p.(f.u) \end{array}$$

Inductive step

$$
\begin{array}{ll}
 & f^p.(dist^l.p.(u|v)) \\
= & \{ \text{ definition of } distr^l \} \\
 & f^p.(dist^l.(p-1).u|dist^l.(p-1).v) \\
= & \{f^p \text{ definition, scalar functions properties}\} \\
 & \Phi^2(f^p.(e_0^2.(dist^l.(p-1).u).(dist^l.(p-1).v)), \ldots, \\
 & \qquad f^p.(e_m^2.(dist^l.(p-1).u).(dist^l.(p-1).v)), dist^l.(p-1).u, dist^l.(p-1).v) \\
= & \{e_i \text{ are simple expressions – use scalar functions}\} \\
 & \Phi^2(f^p \circ\ distr^l.(p-1).(e_0.u.v), \ldots, \\
 & \qquad f^p \circ\ distr^l.(p-1).(e_m.u.v), dist^l.(p-1).u, dist^l.(p-1).v) \\
= & \{\text{induction assumption, and scalar functions properties}\} \\
 & (distr^l.p \circ \Phi)(f.(e_0.u.v), \ldots, f.(e_m.u.v), u, v) \\
= & \{f \text{ definition}\} \\
 & distr^l.p.(f.(u|v))
\end{array}
$$

□

For cyclic distribution the proof is similar; the operator *tie* is replaced with the operator *zip*.

3.2. **Time Complexity.** Considering a function defined on *PowerList*, and a distribution $distr.p.\cdot$, the time-complexity of the resulted program is the sum of the parallel execution time and the sequential execution time:

$$T = \alpha T(f^p) + T(f^s)$$

where α reflects the costs specific to parallel steps (communication or access to shared memory). The evaluation considers that the processor-complexity is 2^p ($O(2^p)$ processors are used).

Example.(**Constant-time combination step**) If the time-complexity of the combination step is a constant $T_s(\Phi) = K_c, K_c \in \mathbb{R}$, and considering the time-complexity of computing the function on singletons is equal to K_s ($K_s \in \mathbb{R}$ also a constant), then we may evaluate the total complexity as being:

$$T = K_c p\alpha + K_c(2^{n-p} - 1) + K_s 2^{n-p} \tag{18}$$

If $p = n$ we achieve the cost of the ideal case (unbounded number of processes).

For example, for **reduction** $red(\oplus)$ the time-complexity of the combination step is a constant, and $K_s = 0$; so we have

$$T_{red} = K_\oplus(p\alpha + 2^{n-p} - 1) \tag{19}$$

For extended operators $\odot$ the combination constant is equal to 0, but we have the time needed for the operator execution on scalars reflected in the constant K_s. A similar situation is also for the high order function *map*. In these cases the time-complexity is equal to

$$T = K_s 2^{n-p} \tag{20}$$

4. Conclusions

The *PowerList* theory forms an abstract model for parallel computation. It is very efficient for developing divide&conquer parallel programs. The abstractness is very high, but we may reconcile this abstractness with performance by introducing bounded parallelism, and so distributions. The necessity of this kind of reconciliation for parallel computation models is argued by Gorlatch in [1], and also by Skillicorn and Talia in [7].

We have proved that the already defined functions on *PowerLists* could be easily transformed to accept bounded parallelism, by introducing distributions. The functions defined based on operator *tie* have to use linear distributions, and the functions defined based on operator *zip* have to use cyclic distributions.

It can be argued that introducing distributions in this theory is not really necessary since we may informally specify that when the maximal number of created processes is achieved, the implementation transforms any parallel decomposition into a sequential one. Still, one advantage of formally introducing the distributions is that it allows us to evaluate costs, depending on the number of available processors - as a parameter. Also, the analysis of the possible distributions for a certain function may lead to an improvement in the design decisions. Another advantage is that we may control the parallel decomposition until a certain level of tree decomposition is achieved; otherwise parallel decomposition could be done, for example, in a 'deep-first' manner, which could be disadvantageous.

Also, after the introduction of distribution functions, mapping on real architectures with limited number of processing elements (e.g. hypercubes) could be analyzed.

References

[1] Gorlatch, S.,: *Abstraction and Performance in the Design of Parallel Programs*, CMPP'98 First International Workshop on Constructive Methods for Parallel Programming, 1998.

[2] Kornerup, J.: *Data Structures for Parallel Recursion.* PhD Thesis, Univ. of Texas, 1997.

[3] Misra, J.: *PowerList: A structure for parallel recursion. ACM Transactions on Programming Languages and Systems*, Vol. 16 No.6 (1994) 1737-1767.

[4] Niculescu, V.: *On Data Distributions in the Construction of Parallel Programs*, The Journal of Supercomputing, Kluwer Academic Publishers, 29(1): 5-25, 2004.

[5] Niculescu, V.: *Designing a Divide&Conquer Parallel Algorithm for Lagrange Interpolation Using* Power*,* Par*, and* P *Theories*, Proceedings of the Symposium "Zilele Academice Clujene", 2004, pp. 39-46.

[6] Skillikorn, D.B.: *Structuring data parallelism using categorical data types.* In Programming Models for Massively Parallel Computers, pp. 110-115, 1993, Computer Society Press.

[7] Skillicorn, D.B. and Talia, D.: *Models and Languages for Parallel Computation.* ACM Computer surveys, 30(2): 123-136, June 1998.

[1] Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca

E-mail address: vniculescu@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 269–273

THE STABLE SETS OF A G-COMPLEX OF MULTI-ARY RELATIONS AND ITS APPLICATIONS

SERGIU CATARANCIUC

Abstract. We define the notion of internally stable set and externally stable set for the G-complex of multi-ary relations. We define also a simple game with two players, where the solution is established by the kernel of G-complex of multi-ary relations.

The notion of complex of multi-ary relations was defined as a discrete structure, which consists of a special sequences of elements of a set X (see papers [1]-[3]). Generalizing some classical structure as graphs, complex of multi-ary relations is useful as a model to solve some problems in projection of integrating network, in informatics, economics etc. If the elements of the complex of multi-ary relations are sequences with repetitions we obtain a new structure called G-complex of multi-ary relations.

Let $X = (x_1, x_2, \ldots, x_r)$ be a set of elements, $r \geq 2$, and $X = X^1, X^2, \ldots, X^n, \ldots, 1 \leq n \leq r$, a sequence of cartesian products of the set X : $X^{m+1} = X^m \times X$, $m = 1, 2, \ldots, n$. Any not empty subset $R^m \subset X^m$, $m \geq 1$, is said to be m-ary relation of elements from X. The set $R^1 \subset X^1$ define a subset of elements from X. The m-ary relation R^m consists of a family of sequence with m elements from X in a given order. Now, consider the finite subset of relations $R^1, R^2, \ldots, R^{n+1}$ of the infinite set mentioned above. Require that this subset satisfies the conditions:

I. $R^1 = X^1 = X$;

II. $R^{n+1} = \emptyset$;

III. Any sequence $[x_{j_1}, x_{j_2}, \ldots, x_{j_l}]$, $1 \leq l \leq m \leq n+1$ of the sequence $[x_{i_1}, x_{i_2}, \ldots, x_{i_m}]$, which represents sequence, is contained in R^l.

Definition 1. *A family of relations $R^1, R^2, \ldots, R^{n+1}$ which satisfies the conditions I-III is said to be* ***finite-complex of multi-ary relations*** *and is denoted by $\mathcal{R}^{n+1} = (R^1, R^2, \ldots, R^{n+1})$.*

2000 *Mathematics Subject Classification.* 18F15, 32Q60, 32C10.

Key words and phrases. G-complex of multi-ary relations, internally stable set, externally stable set, kernel.

©2007 Babeş-Bolyai University, Cluj-Napoca

According to the conditions I-III it results that any set R^m of the complex of relations $\mathcal{R}^{n+1}$ is not empty.

By according to the study of G-complex of multi-ary relations in paper [4] is defined the basic notions of G-subcomplex, connectivity, path etc. A complex of multi-ary relations $\mathcal{R}^{n+1} = (R^1, R^2, \dots, R^{n+1})$ can be represented as a family of abstract simplexes and is denoted by $K^n = (\mathcal{S}^0, \mathcal{S}^1, \dots, \mathcal{S}^n)$ (see paper [4]).

Now let's examine the G-complex formed by the characteristic faces of the simplex $S^m \in K^n$, $m > 0$, which will be denoted $[S^m]$ and will be written $K_s^{m-1} = [S^m]$.

Definition 2. *Let K^n be a G-complex of multi-ary relations, an arbitrary simplex $S^m = [x_{i_0}, x_{i_1}, \dots, x_{i_m}] \in K^m$, $m > 0$, and $[S^m]$ – the complex formed by S^m. The difference $S^m \backslash [S^m]$ will be called **vacuum** with the dimension m and will be notated by $\overset{\circ}{S}{}^m = S^m \backslash [S^m] = (x_{i_0}, x_{i_1}, \dots, x_{i_m})$.*

Let's say F represents a set of simplexes from K^n. We will denote by T_F the family of all simplexes from K^n which does not contains the own face of simplexes of F, but are incident with F. We will denote by T_F the set of all vacuums (with different dimensions) of these elements (simplexes). Let's examine the difference $K^n \backslash (F \cup T_F)$. So, to simplify the notations, in the place of $K^n \backslash (F \cup T_F)$ we will use $K^n \backslash F$.

Denoted by $st\, S^m$ the set of all simplexes of dimension $m+1$, for which S^m is a common face, and all faced of these simplexes. The set $st\, S^m$ is said to be **star** of simplex S^m.

Let S^m be a simplex of G-complex of multi-ary relations K^n and K_s be a subcomplex of K^n that contains S^m, all proper faces of S^m, all simplexes of K^n incedented to S^n, and all his faces. The subcomplex $K_s \subset K^n$ is called **superstar** of S^m and is denoted by $St\, S^m$. The subcomplex $R(S^m) \subset K^n$ which consists of the set of all nonincidented simplexes to S^m of the superstar $St\, S^m$ is called **the representative of this superstar** and is denoted by $R(S^m)$.

Let's assume we have a connected G-complex of multi-ary relations K^n, an arbitrary simplex $S^m \in K^n$, a superstar $St(S^m)$ and a representative $R(S^m)$ of a superstar. We know that $R(S^m) \cap S^m = \emptyset$.

Now let's conseder a point to set mapping

$$\Gamma : K^n \to K^n,$$

with the property:

$$\Gamma(S^m) \subset R(S^m),$$

for all simplexes $S^m \in K^n$, $m = 0, 1, \dots, n$.

Definition 3. *Let's say I is a family of simplexes of K^n that satisfies the condition: for any $S^m \in I$ it holds*

$$\Gamma(S^m) \cap I = \emptyset.$$

The family of simplexes I of the G-complex K^n is said to be ***internally independent set*** *(****internally stable set****) of simplexes of the G-complex K^n.*

Remark 1. *Not every G-complex of multi-ary relations possesses internally stable sets of simplexes relating to Γ, formed by the oriented complex*

$$S^m = \varepsilon(m)[x_{i_0}, x_{i_1}, \ldots, x_{i_m}] \in K^n.$$

Let K^n be a complex that admits independent sets, the J-ensemble of all shown sets of K^n.

Consequence 1. *If $I_1 \in J$ and $I_2 \subset I_1$, then we obtain the relation $I_\Gamma^2 \subset I_\Gamma$.*

Definition 4. *Let's consider K^n and the ensemble J. The value*

$$\alpha(K^n) = \max_{I \in J} \{card\, I\}$$

is to be said the ***internally independent number*** *of G-complex K^n (****the number of internally stability****).*

Definition 5. *Let K^n be a connected G-complex of multi-ary relation, and E a family of simplexes from K^n with the property: for any $S^m \in K^n \backslash E$ holds the relation $\Gamma(S^m) \cap E \neq \emptyset$. The family $E \subset K^n$ is called* ***externally stable set*** *(****externally stable set****) of simplexes of G-complex K^n.*

Let $\mathcal{E}$ be the ensemble of all externally stable sets of simplexes of complex K^n.

Consequence 2. *If $E_1 \in \mathcal{E}$, $E_2 \in \mathcal{E}$, where $E_1 \subset E_2$, then $E_2 \in \mathcal{E}$.*

Definition 6. *Let K^n be a complex of multi-ary relations, and E_Γ – a proper ensemble. The value*

$$\beta(K^n) = \min_{E \in \mathcal{E}} \{card\, E\}$$

is to be said the ***externally independent number*** *of G-complex K^n (****the number of externally stability****).*

Definition 7. *Let consider a connected G-complex of multi-ary relations K^n, application Γ, and $N \subset K^n$ – a family of simplexes with the properties:*

a) N is a set of internally stable simplexes;

b) N is a set of externally stable simplexes of K^n;

The family N is called ***kernel*** *of G-complex K^n.*

Theorem 1. *Let's consider a connected local complete G-complex of multi-ary relations K^n which admits the family of internally stable sets $J \neq \emptyset$. An element $I \in J$ is the kernel of complex K^n if and only if this one is maximal.*

Proof. Let $N \in J$ be the kernel of K^n and let's admit that this one is not maximal internally stable set of K^n. In this case there is at least one simplex $S^m \in K^n$ that satisfied the property $N \cup \{S^m\} = I \in J$ and $S^m \notin N$. By definition 7 (see the property a), if $I \in J$, we have $\Gamma(S^m) \cap I =$. On the other hand, in virtue of definition 7 (see the property b), using the relation $S^m \in K^n \backslash N$, it results the inequality $\Gamma(S^m) \cap N \neq \emptyset$, so $\Gamma(S^m) \cap I \neq \emptyset$.

Let's assume now that $I \in J$ is a maximal internally stable set of simplexes of the complex K^n, and let's demonstrate that I is a kernel of K^n too. We admit the opposite, there is a simplex $S^m \in K^n \backslash I$, such that $\Gamma(S^m) \cap I = \emptyset$. If this equality is satisfied, we obtain immediately that the family $I \cup \{S^m\}$ represents an internally stable set of simplexes of complex K^n, as $\Gamma(S^m) \cap S^m = \emptyset$. The contradiction with the assuming that $I \in J$ is maximal. The theorem 1 is proved.

Definition 8. *The next procedure is to be called* ***a simple game with the players*** *A* ***and*** *B* ***on the*** *G***-complex*** K^n.

1) *A simplex* $S^m \in K^n$ *is determined arbitrary.*
2) *The player A picks up a simplex from the set* $\Gamma(S^m)$ *(if* $\Gamma(S^m) = \emptyset$*, the player A loses). Let* S^{m_1} *be the simplex picked up by A.*
3) *The player B picks up a simplex from* $\Gamma(S^{m_1})$ *(if* $\Gamma(S^{m_1}) = \emptyset$*, the player B loses). Let* S^{m_2} *be the simplex picked up by B.*
4) *and so on*

Definition 9. *The sequence of simplexes* $S^{m_0}, S^{m_1}, \ldots, S^{m_t}$ *of G-complex* K^n *is to be called* ***trajectory*** *of* K^n *related to* Γ*, if the following conditions are satisfied:*

1) *any pair of simplexes* $S^{m_i}, S^{m_{i+1}}$ *is particular,* $i = 0, 1, \ldots, t-1$*;*
2) *for* $\forall S^{m_i} \in K^n$ *it holds* $S^{m_{i+1}} \subset \Gamma(S^{m_i})$, $i = 0, 1, \ldots, t-1$*;*
3) *if* $S^{m_{i+1}}$ *and* $S^{m_{i-j}}$ *are intersected by a simplex* $S^{m_{i+1,i-j}}$*, where* $j = 0, 1, \ldots, i-1$*, then this simplex does not belong to the* $S_{m_{i+1,i-j}}$.

The trajectory $S^{m_0}, S^{m_1}, \ldots, S^{m_t}$ *will be denoted by* $T(0,t)$.
The trajectory $T(0,t)$ *is said to be* ***maximal*** *related to* Γ*, if exists a natural number* t*, such that* $\Gamma(S^{m_t}) = \emptyset$*. The trajectory* $T(0,t)$ *is said to be* ***contour-trajectory*** *related to* Γ*, if exists* t_0*, such that* $S^{m_0} = S^{m_{t_0}}$.
Let's have a connected G-complex of multi-ary relations K^n*, a series of nonnegative integers* N_0 *and a function* $g : K^n \to N_0$*, with the property: for any* $S^m \in K^n$ *it holds* $g(S^m) = min\{N_0 \backslash g(\Gamma(S^m))\}$*. The application g is to be called* ***Grundy function*** *related to* Γ.

Not every G-complex K^n has a Grundy function.

It holds

Theorem 2. *Let* K^n *be a connected G-complex of multi-ary relations, and we consider it exists a Grundy function for* K^n *related to* Γ*. The set of all the*

simplexes S^m of K^n that satisfies the relation $g(S^m) = 0$ represents the kernel of complex K^n.

Proof. Let M be the set of all the simplexes of K^n, which satisfies the relation $g(S^m) = 0$. We will show that:

1) M is an internal stable multitude of simplexes;
2) M is an external stable multitude of simplexes.

1) Let $S^{m_1} \in \Gamma(S^m)$ be a random simplex. From the definition of Γ and the application g, it results that $g(S^{m_1}) \neq 0$, that is $\Gamma(S^m) \cap M = \emptyset$.
2) Let $S^{m_1} \in K^n \backslash M$ be a random simplex. In this case $g(S^{m_1}) \neq 0$, so $\Gamma(S^m) \cap \cap M \neq \emptyset$ and M represents a kernel of the complex K^n.

The theorem 2 is proved.

Theorem 3. *If the G-complex K^n dispose of a contour-trajectory related to Γ, then the simple game on this G-complex leads to the fact that no one of the players A and B lose.*

The proof of theorem 3 is obvious because the players A and B choosing the simplexes on the contour-trajectory. Of cause, they always have the possibility to make the proper movement, if exists a simplex $S^{m_i} \in \Gamma(S^m)$, where S^m is an arbitrary simplex.

References

[1] Soltan P., *On the homologies of multi-ary relations and oriented hipergraphs*, Studii în metode de analiză numerică şi optimizare, vol. 2, nr 1(3), pp.60-81, Chişinău, 2000.
[2] Cataranciuc S., Soltan P., *Hipergrafuri şi omologiile lor*, Trends in the Development of the Information and Communication Technologies in Education and Management, International Conference, March 20-21, pp.294-300, 2003.
[3] Cataranciuc S., *About connectivity of general complexes of multi-ary relations*, The XIV-th Conference on Applied and Industrial mathematics (Satelite Conference of ICM 2006), pp.94-95, Chişinău, August 17-19, 2006.
[4] Cataranciuc S., *G-complexul de relaţii multi-are*, Analele ştiinţifice ale Universităţii de Stat din Moldova, Seria "Ştiinţe fizico-matematice", p.119-122, Chişinău, 2006.

Moldova State University
Faculty of Mathematics and Computer Science
60 A. Mateevici street, Chişinău, MD−2009
Republic of Moldova
E-mail address: caseg@usm.md

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 274–281

MULTI-AGENT SYSTEM FOR COMPETENCE MODELING

ALEXANDRU CICORTAS[1] AND VICTORIA IORDAN[2]

Abstract. Modeling compentences is a real challenge for the people working in a lot of domains like Human Resource (HR), companies, e-Learning related activities, universities and not only. Universities and e-Learning organizations define the prerequisites that must be fulfilled before joining and the competences that will be acquired after successful completion. Based on these are stated the curricula or training programmes. The future students and companies mainly HR departments try to define their own competences and these are compared with those offered by the applicants and universities respectively. The proposed model will be included in the project Cex 05-D8-66/2005. It will offers to the universities, companies and students to define and to present the competencies, making comparisons between them with appropriate scores and evaluations in order to use in an efficient way.

1. Introduction

Due to the IT evolution the people mobility has increased and also the managers and Human Resource departments have more difficulties in deciding the most appropriate qualifications (the right qualifications) to join a job in company or in a project. Some of the major problems that appear most currently for the universities and e-Learning companies is that the competences can be well stated and valuated from simplest ones to the complex ones. From the following examples:

- an applicant needs to posses a Bachelor degree to apply for Master studies. This is the prerequisite;
- in both Bachelor degree and Master studies there is the course of Software engineering but on different levels. The name of the course is the same but their content is different. In order to attend an expert course on a topic, a certification on a basic level may be required;
- for the Human resource departments the need is that to match the applicant experience with the requirements of a job offer, including the mandatory requirements and the desired ones.

2000 *Mathematics Subject Classification.* 68T99, 68U99.
Key words and phrases. competence, modeling.
The research for this paper has been supported by the project Cex 05-D8-66/2005.

©2007 Babeş-Bolyai University, Cluj-Napoca

we can conclude that for representation of competences in a large variety and sufficiently expressive and necessary formats:

- matching the competences (profiles, requirements);
- expressing the inheritance;
- expressing the part of;
- mechanisms for increasing the reusability;
- the relationships between competences.

In [4] and [3] focus on reusable competency definitions. The basic idea is to define the repositories that concentrate the competencies defined for certain communities. These can be referenced by external data structures, allowing the interoperability and reusability.

2. Related Work

Concerning the competence were done some standardization efforts on modeling competences. The efforts focused on aspects related on competency: profiles and relationships among competencies. The IMS Reusable Definition of Competencies or Educational Objective [4] focus on reusable competency definitions. It lacks information on context and proficiency level and does not allow relations or recursive dependencies among competencies. HR-XML focuses on the modeling of information related to human resource tasks. It tries to define profiles in order to use such competency definitions. Here data sets are specified:

- as job requirement profiles - the competencies that a person is required to possess;
- personal competency profiles - that describes the competencies that a person has.

This model does not make a clear distinction between the required and acquired profiles. In [5] the relationships between competencies are modeled. The map can contain information about dependencies or equivalencies among competencies. The capability of composing complex competencies for simpler ones is also taken under consideration. The basic representation is a directed acyclic graph. Due to the fact that the relationships are different meanings i.e., composition, equivalence os order dependency, the model can lead to some confusions.

In [1] the competence (plural competences) is defined as effective performance within a domain context at different levels of proficiency. The competency (plural competencies), is defined in [3] and [4] as any form of knowledge, skill, attitude, ability or learning objective that can be described in a context of learning, education or training. In [2] these definitions are considered to be insufficiently expressive for gap analysis. The context information and the proficiency level scale is not included in the models given in [3], [4]. The model given in [2] competency proficiency level and context are three different dimensions that must be modeled separately in order to maximize their reuse. As an example the same

compentencies may be used in different contexts, or the same proficiency level scales may be reused among different certifications. The same can be applied to the contexts (domain models)that in many situations already exist and therefore may be reused by competences. As a consequence in [2] is modeled the competence (plural competences) as a three dimensional variable: competency (plural competencies), a proficiency level and a context.

As example for illustrating the [2] model given in Figure 1 for *Fluent Business English* that is composed by competency *English* the proficiency *Fluent* and the context *Business*. Also [2] in order to avoid the confusion between the terms competence and competency they propose that competency and skill as being interchangeable but skill is not a synonymous for competency as it covers a part of its scope.

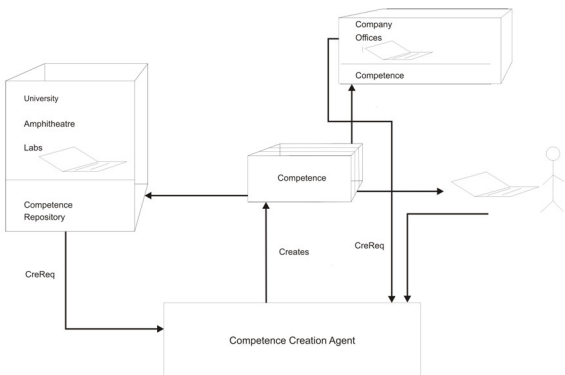


FIGURE 1

3. Requirements for Modeling a Competence

The IEEE Reusable Competency Definitions provide a model for the representation of competencies , the objective being referencing and cataloging an competency not classifying it. The model does not provide any means to specify the relationships between the competencies. The relationships must be take into account that the compentences are it composed from competency proficiency level and context.

Different scales qualitative and quantitative may be used in order to represent proficiency levels. AS an example a computer science curricula want to specify whether a student has acquired a compentence or not whereas an English certification institution may want to classify the students into intermediate, advanced or proficient. Many different scales may be used but it should be possible to reuse them within and across the borders of the institution.

Among elements of one scale of proficiency levels there are implicit relationships. For example a proficiency level may be subsumed by another: *proficient* subsumes *advanced* which subsumes *intermediate*. Such relationships must be modeled due to the fact that these are needed for competence matching. For instance a job requiring someone with intermediate English skill typically has implicit quantifier *at least*, it means that anyone with advanced English will be accepted (being even preferred). One of the possibilities is that to represent as an ordered list the proficiency level scale. In such a list the minimum value (subsumed by any other in the list) is given by the first element and the maximum is given by the last one. Therefore the order in the list represent subsumption relationships, that is, the first element is subsumed by the second one and so on.

In order to improve the interoperability and matching among scales, an optional field is included for mapping to the universal scale (e.g., *[0,1]*). The reason why this mapping field is optional is that even though it would be useful to include it, in some contexts it may not be possible to find a suitable mapping or it may not even be necessary.

Competence descriptions can refer to specific items of these scales in order to represent the proficiency level recquired/acquired. Algorithms could take relationships among proficiency levels into account in order to find out how much training/learning is required to reach a determined employee/learner proficiency level.

The context can be defined:

- interrelated conditions in which something exists or occurs (Webster on line dictionary, http://webster.com [6]) or
- the circumstances and conditions which surround it (Wikipedia, http://en.wikipedia.org [7]).

Regarding to competences, context may refer to different concepts like:

- the specific occupation in which a competence is required;
- a set of topics within a domain;
- even the personal settings related to the student.

And these are contexts which may be part of a competence. Context descriptions con not be defined in general but these depends on the scope and the purpose of the competence descriptions to which they are attached. Also the context definitions may be reused.

Modeling contexts is a complex task, it may coincide with modeling the whole domain knowledge of an institution. Ontologies can capture such knowledge and use arbitrary complex structures from simple sets or tree structures to directed acyclic graphs. The existing relationships between context elements (regarding their use within competences) do not show the need for providing a graph representation or multiple inheritance.

Competences generally can be described as reusable domain knowledge. Any model representing competences describes what a competence is and how it is composed of sub-competences. Due to the fact that the competneces are referenced in different situations like:

- certifications;
- job descriptions;
- personalizing relevant competences for their business that are included in job offers projects descriptions.

Based on these the competence must be adequate represented and described in order to:

- how a competence may be achieved (ex: by acquiring some sub-competences);
- to which level each competence should be acquired;
- whether sub-competences must be all achieved or simply a subset of them;
- if the sub-competences must be acquired in a specific order.

As an other significant problem is that the capability of the model to represent aggregate and alternative structures of the competence. The aggregation allows that the competence is composed from several sub-competences all of them required. Alternative competence can be viewed as a set of compentences and there can be possible to specify by a numeric interval what is the number of alternatives that must be acquired.

Due to the multiple usages of the model it is also important that the equivalence relationships between the competences to be well defined, understood and used by all users.

4. Problem Statement

There are many domains where the competences are used. Our intention is to define a model that can be used by the:

- universities in order to:
  - define the competences acquired by the students after a undergraduate or graduate profile;
  - evaluate the competence in academic media;
  - give the details concerning the conditions and modalities for gaining a competence;
- a student in order to define its own compentences and own conditions for:
  - finding the appropriate university that satisfies its beliefs and desires;
  - offering their own compentences for companies or/and universities;
  - compare their competences with those of companies or/and universities;

- companies that can:
  - define their own requirements concerning the competences for job offers;
  - compare their competences with those of the universities and students for stages or persons who are searching a job.

Our model will have tools that will allow:

- defining the competences and use them under specific needs;
- searching the universities/companies for required competences;
- matching the own compentences with those that were find (offered by universities/companies);
- valuating and ordering the competences after some criteria.

5. Proposed Model

The competences are frequently used in the relations between the universities and the future students, between the companies and the future employees. Our model intend to allow to the universities, students, employees and companies to:

- construct and maintain their own compentences;
- evaluate their competences;
- match their own compentences with the other competences;
- search the desired competences in appropriate domains.

Comparing the competences In order for an efficient usage it is intended to offer a tool that make an exhaustive analysis concerning the competences. It will mainly based on the details that are given for a competence: -the components of the competence. Here the specific agent will compare the occurrence of the competence components scoring the matching between the two competences also the order of components will be taken into the account; -the resources used for gaining a competence. -the effort that must be fulfilled by the student in order to gain certain competence. The students that intend to obtain some qualification (and some competences) can make some suppositions concerning the financial effort and their own effort and time and it will be offered in an adequate manner.

The model is based on a multi-agent system that is constituted from appropriate agents that will fulfill these objectives. The agents are:

- Compentence Creation Agent (CCA);
- Evaluator Agent (EvA);
- Broker Agent (BrA);

The user can be either a university, a company, a student or an employee. He (the user) can submit to the CCA requirements to create competences from basic sub-competences. The CCA creates and furnish the competences to the user that can place them into a Competence Repository or use immediately in new requirements.

The compentece creation can be:

- the simplest one when the compentence is defined from sub-competences;
- the medium one when is given the competence and the details that specify the conditions that must be fulfilled in order that the requester obtain the competence;
- the complex one when we have all details that are needed with the quantification of required conditions and the quality of obtained competence. All these will be valuated after an evaluation.

Every user can construct its own Competence Repository. When a user (student, employee, company) want to find some competence that is placed in an university or company he must furnish the generic competence and eventually the sub-competences, the conditions that allow to acquire the competence and other own requirements concerning the competence. In the Figure 2 is presented the proposed model.

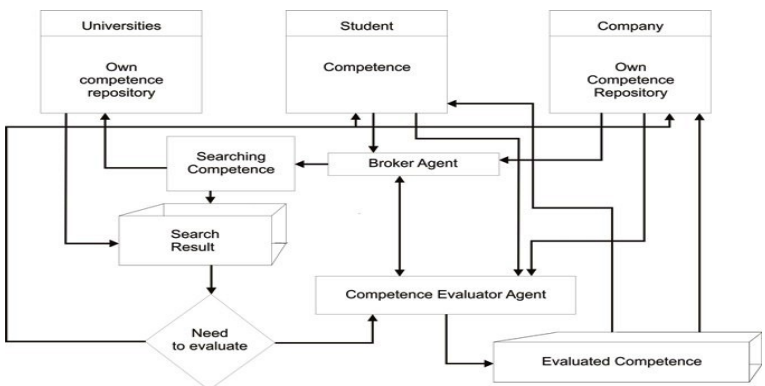


FIGURE 2

All these requirements are taken by the BrA that search (usually on the Internet) and tries to match the user requirement with those that were found. The matching can be in a wide range, starting from a simple matching (the competence name) thur a complex one where a lot of actions are executed by the EvA:

- matching of competences with appropriate scores;
- matching the sub-competences and scoring the matching;
- matching of conditions and giving the scores in some order (preferred by the user or a predefined system order).

The system work as follows. The companies and universities have their Competence Repositories that are posted as web pages. The users: students, employees, applicants, the universities and companies can define the requirements that are

addressed to the system. The system agents try to satisfy the requirements in different levels of details and complexity, as was stated above. The paper purpose is to give some fundamentals for the proposed system that will allows to use agents in order to create, find and compare the competences. Based on this paper in future work and future papers will be stated the specific agent technology will be used in the proposed system.

6. Future Works

The model suppose that the users are able accessing a tool that allows to develop their own compentences, dispose them in a Compentence Repository form the others can access interpret and compare them. As an immediate activity the modules that allow to create and access the compentences will be developed. The next step will allows to refine the search and comparison between the competences will be defined and developed.

References

[1] Cheetam, G., Chivers G., Professions Compentence and Informal Learning, Edgard Elgar Publishing Limited, Journal of Interprofessional Care, Volume 20, Number 5, October 2006, pp. 569-570.

[2] De Coi J.L., Herder, E., Koesling A., Lofi, C., Olmedilla, D., Papapetrou, O., Siberski, W., A Model for Competence GAP Analysis, In WEBIST 2007, Proceedings of the Third International Conference on Web Information Systems and Technologies: Internet Technology / Web Interface and Applications, Barcelona, Spain, Mar 2007. INSTICC Press.

[3] ***, IEEE 1484.20.1/draft - draft standard for Reusable Competency Definitions (RCD), 2005.
http://ieeeltsc.org/wg20Comp/Public/IEEE_1484.20.1.D3.pdf.

[4] ***, IMS Reusable Definition of Competency or Educational Objective (RDCEO), 2002. http://www.imaglobal.org/competencies.

[5] Simple Reusable Compentency Map Proposal 2006,
http//:www.ostyn.com/resource.htm

[6] http://webster.com

[7] http://en.wikipedia.org

(1) Computer Science Department, Mathematics and Computer Science Faculty, West University of Timisoara
E-mail address: cico@info.uvt.ro

(2) Computer Science Department, Mathematics and Computer Science Faculty, West University of Timisoara
E-mail address: iordan@info.uvt.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 282–289

DATA VERIFICATION IN ETL PROCESSES

MARIAN BALTA[1]

Abstract. The ETL processes are responsible for the extraction of the data from the external sources, transforming the data in order to satisfy the integration needs and for loading the data into the data warehouse. On the other hand, in the data mining world, there is a special concern on using the metrics for efficient classification algorithms. One of these approaches is the one that uses metrics on partitions based on the Shannon entropy (or other forms of entropy), to study the degree of concentration of values. In this paper we show how this idea can be used in verification of the consistency of data loaded into the data warehouse by ETL processes. We calculate the Shannon entropy and Gini index on partitions induced by attribute sets and we show that these values can be used to signal a possible problem in the data extraction process.

1. Introduction

In a data warehouse, the periodical integration of the data from the external sources trough the ETL processes produces large amounts of data. Even that each ETL process contains verification stages, it is possible that some particular kind of anomalies may occur and not be detected. These anomalies are not necessarily caused by a malfunction but it is very useful (if not mandatory) to find them. For example, if, at a given moment, one of the external data sources does not provide any data (or less data than the rest of the sources or than usual), then something may be wrong. The detection of such a scenario can easily be done using entropy defined on partitions. The idea was suggested to us by professor Simovici after a presentation he made on a summer school in Iasi [4].

Much work has been done in respect to the use of the entropies into the data mining field. Regarding the use of partition entropy you can find a complete formal description in [5]. Usually, in data mining, the notion of entropy is used to define metrics on partitions sets, in attempt to develop good classification algorithms. However, computing the entropy of a data set in respect to a given partition can

2000 *Mathematics Subject Classification.* 68P15, 68P20.
Key words and phrases. Data warehouse, ETL, Shannon entropy.

©2007 Babeş-Bolyai University, Cluj-Napoca

provide some very useful information regarding the distribution of values over the partition blocks.

The use of metrics in ETL field is not new. In [7], the authors model an ETL scenario as a graph and introduce specific importance metrics. In other works, like [2], the metrics are defined and used for evaluation of real industry ETL products. We use the Shannon entropy, calculated for a partition induced on a relational table by a set of attributes, to verify the consistency of data extracted by the ETL processes from the external sources. We also show that the choice of the attributes set plays a very important role in the efficiency of the method.

The paper is structured as follow. The sections 2 and 3 provide a short introduction in the definition of the Shannon entropy, Gini index and the partitions induced by attributes sets. The section 4 briefly presents the ETL notions, the dimensional model and an example used in section 5 to demonstrate how the entropy can be used to verify the consistency of the data. The paper ends with some concluding remarks.

2. Shannon entropy for partitions

The concept of entropy in information theory describes how much randomness there is in a signal or random event. It is formally defined by Claude E. Shannon in 1948 in his paper "A Mathematical Theory of Communication" [3]. In terms of a discrete random event X, with n possible states, the Shannon entropy is defined as:

$$H(X) = -\sum_{i=1}^{n} p_i \; log \; p_i$$

where p_i is the probability of occurrence of the state i. We call H the entropy of the set of probabilities $p_1, \ldots, p_n$.

The quantity H has a number of interesting properties:

(1) H = 0 if and only if all the p_i but one are zero, this one having the value unity. Intuitively, this means that we know the outcome since only one event can occur. In any other situation, the value of H is positive.
(2) For a given n, H is a maximum and equal to $\log n$ when all p_i pi are equal (i.e., $\frac{1}{n}$). This is also intuitively the most uncertain situation.
(3) Suppose there are two events, X and Y, in question with m possibilities for the first and n for the second. Let *p(i,j)* be the probability of the joint occurrence of i for the first and j for the second. The entropy of the joint event is

$$H(X,Y) = -\sum_{i,j} p(i,j) \; \log p(i,j)$$

while

$$H(X) = -\sum_{i,j} p(i,j)\ \log \sum_{j} p(i,j)$$

$$H(Y) = -\sum_{i,j} p(i,j)\ \log \sum_{i} p(i,j).$$

It is easily shown that

$$H(X,Y) \leq H(X) + H(Y)$$

with equality only if the events are independent (i.e., *p(i,j)=p(i)p(j)*).The uncertainty of a joint event is less than or equal to the sum of the individual uncertainties.

(4) Any change toward equalization of the probabilities $p_1, p_2, \ldots, p_n$ increases H.

These properties qualify H as a measure of the uncertainty of the outcome of the event X.

To extend this to a partition, we have to observe that giving $\pi = \{B_1, \ldots, B_n\}$, a partition on a finite and nonempty set A, we can associate a random variable as follow:

$$X_\pi = \begin{pmatrix} inB_1 & & inB_n \\ \frac{|B_1|}{|A|} & \cdots & \frac{|B_n|}{|A|} \end{pmatrix}$$

The Shannon entropy of π is defined as the Shannon entropy of X_π. This entropy can be used to measure the concentration of values in the partition. For example, if we have a set of 15 elements in A, and a partition having 5 blocks then, for each following cases we have:

- $|B_1| = 11, |B_2| = 1, |B_3| = 1, |B_4| = 1, |B_5| = 1 \Rightarrow H(X_\pi) = 1.3699$
- $|B_1| = 6, |B_2| = 6, |B_3| = 1, |B_4| = 1, |B_5| = 1 \Rightarrow H(X_\pi) = 1.8389$
- $|B_1| = 5, |B_2| = 4, |B_3| = 4, |B_4| = 1, |B_5| = 1 \Rightarrow H(X_\pi) = 2.0662$
- $|B_1| = 3, |B_2| = 3, |B_3| = 3, |B_4| = 3, |B_5| = 3 \Rightarrow H(X_\pi) = 2.3219$

It is known that the value of the Shannon entropy is proportional with the degree at witch the element are equally scattered among the blocks of the partition. The larger the entropy, the more the elements of A are scattered among the blocks of π.

Another way to have a measure of the distribution of the elements between blocks is to calculate Gini's index using the following formula [5]:

$$H_1(X) = 1 - \sum_{i=1}^{n} p_i^2$$

Using the same example from above we obtain the following values:

- $|B_1| = 11, |B_2| = 1, |B_3| = 1, |B_4| = 1, |B_5| = 1 \Rightarrow H_1(X_\pi) = 0.4444$
- $|B_1| = 6, |B_2| = 6, |B_3| = 1, |B_4| = 1, |B_5| = 1 \Rightarrow H_1(X_\pi) = 0.6666$

- $|B_1| = 5, |B_2| = 4, |B_3| = 4, |B_4| = 1, |B_5| = 1 \Rightarrow H_1(X_\pi) = 0.7377$
- $|B_1| = 3, |B_2| = 3, |B_3| = 3, |B_4| = 3, |B_5| = 3 \Rightarrow H_1(X_\pi) = 0.8$

In fact, both formulas are particular cases of the generalized entropy of partitions introduced by Daróczy in the following form:

$$H_\beta(\pi) = \frac{1}{1 - 2^{1-\beta}} \left(1 - \sum_{i=1}^{n} \left(\frac{|B_i|}{|A|} \right)^\beta \right)$$

It is easy to see that for $\beta = 2$ we obtain the Gini index and that $\lim_{\beta \to 1} H_\beta(\pi)$ is Shannon's entropy.

3. Partitions induced by attribute sets

A table T in a relational database is a set ρ of tuples on a set of attributes $A = \{A_1, \ldots, A_n\}$, where each tuple $t \in Dom(A_1) \times \ldots \times Dom(A_n)$. The set ρ is called the content of the table T [6]. Any subset of attributes $K \subseteq A$ induces a partition on the content of the table, denoted by π_K. For each different set of equal values for the projection of the table T on K, we have a corresponding block of the induced partition (Table 1). Some interesting results have been obtained

	$\cdots$	$\leftarrow K \rightarrow$	$\cdots$
t_1	$\cdots$	k_1	$\cdots$
t_2	$\cdots$	k_1	$\cdots$
t_3	$\cdots$	k_1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_h	$\cdots$	k_p	$\cdots$
t_{h+1}	$\cdots$	k_p	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_m	$\cdots$	k_r	$\cdots$
t_{m+1}	$\cdots$	k_r	$\cdots$

Table 1. The partition induced on a table T by the set of attributes K

in data mining field, regarding the classification algorithms using decision trees, using this induced partition [5].

4. ETL and the Dimensional Model

Extract, transform and load (ETL) is a set of processes that include, as the most important parts, the following:

(1) the identification of relevant information in source sistems;
(2) the extraction of that information;

(3) the integration of the information coming from multiple sources into a common format;
(4) the cleaning of the resulting data set, on the basis of database and business rules;
(5) the propagation of the data to the data warehouse.

Despite the popularity of relational normal forms, the data warehouse field has some particularities that need to be considered. This is the reason that has lead to the use of a new conceptual design model - the dimensional model. The dimensional modeling is a technique used in conceptual modeling, and its aim is to present the data in a standardized manner and to allow a very fast access, in order to support analytical processing. The model is, obviously, dimensional and it uses the relational model with some very important restrictions. Each dimensional model is composed from a table with a multiple key, which is called the fact table, and a set of (smaller) tables, called dimensions. Every one of the dimensions has a singular key, usually corresponding to one of the components of the fact table key.

A fact table, because it has a multiple key formed by two or more foreign keys, always represents a many-to-many relation. Another very important aspect of the dimensional model is that the fact table contains one or many numerical columns called measures, attached to the combination of keys which defines each row. An important property of these columns is that one has to be able to aggregate them. The importance reside from the fact that the applications that use this model almost never use a single record; usually they select hundreds, thousands or even millions of rows and submit them to an aggregation.

We can find very good examples of dimensional model in [1] and we will use a version of that example in the following. We consider a data warehouse containing information about sales (Figure 1).

The fact table has a multiple key formed by the *time_key*, *product_key*, *store_key* and *customer_key* attributes. These fields are foreign keys that define the relations between the fact table and the dimensions. The main purpose of this model is to allow fast aggregation of sales over every dimension.

5. Verifying consistency

Every notion mentioned above is well known and used in many areas. In the following section we give a description of the way Shannon entropy and Gini's index can be used to verify the consistency of the data that is extracted from the external sources.

Depending on the nature of the dimensions involved in the schema, we can identify important cases in which the data that enter the data warehouse is distorted by anomalies caused by the inaccessibility of some parts of data. As the time and

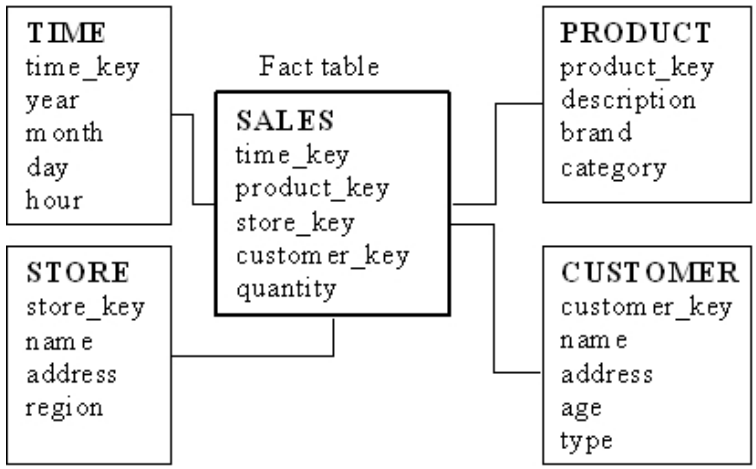


FIGURE 1. The dimensional model

spatial dimensions are always present in a data warehouse, let's consider one of the following situations:

(1) due to technical problems, one of the stores (or one of the regions) was unable to send the necessary information for data warehouse update, or,
(2) the missing data is from a specific period of time (let's say a day).

In the firs case we are interested in detecting if the sales from a store are missing. It is easy to see that we can define the partition induced on the table SALES by the attribute *store_key*. Let's assume that the sales are divided among the five stores, like in Table 2, during three consecutive days.

day \ store	**1**	**2**	**3**	**4**	**5**
1	100,000	110,000	100,005	100,003	120,000
2	130,000	90,000	0	101,000	115,000
3	100,000	110,000	20,000	100,000	120,000

TABLE 2. The distribution of the sales among the stores

If we calculate Shannon entropy and Gini index for the induced partition over these three days, we have:

The results show that, if we carefully choose a level, we can signal as a possible fault any input for which the entropy falls bellow that level. However, the differences in values can have other cause than a technical problem and further testing is in order. For example, in the second day, from the 3^{rd} store there are no records.

day	H	H_1
1	2.3179	0.7989
2	1.9864	0.7453
3	2.1694	0.7684

TABLE 3. The Shannon entropy and Gini index using *store* dimension

We can see that this has a significant impact on the value of the Shannon: it has a smaller value than on the first day. This shortcoming in data can be caused by a technical problem but can easily be a normal situation (an event that required the closing of the second store that day). This is why the verification using the entropy must be followed by a further analysis.

For the second situation mentioned, in which the missing data is from a specific period of time, the counting of the sales has to be done for each store and analyzed in the same manner. The incoming data should be almost evenly distributed in time. So, this time we calculate the Shannon entropy and Gini index for each store, considering the partition induced by the attribute *time_key*. We have:

store	H	H_1
1	1.5734	0.6612
2	1.5788	0.6639
3	0.6502	0.2778
4	1.5849	0.6667
5	1.5847	0.6665

TABLE 4. The Shannon entropy and Gini index using *time* dimension

It is easy to observe that, in this case, the differences between the values of the entropy and Gini index for a regular store and for the store that has problems (3^{rd} store) are greater than in the previous case. This is due to the fact that for each store the values are more equally distributed among each day than they were among stores on a particular day.

From these two examples we can see that the choice of attributes used to define the partition have a great impact on the outcome. The attributes must be selected to assure that, in a normal situation, the number of elements (records) in every block of the induced partition is nearly equal.

6. Conclusions and further work

The notion of entropy and its applications in information theory are very powerful tools. The data mining field is using powerful algorithms obtained using this theory. By proper defining the involved parameters, it can be efficiently used for verification of the data loaded into the data warehouse.

In our approach, the value of the entropy remains high as long as the data from the operational sources is loaded into the warehouse on the regular basis. A disruption in this rhythm causes a change in the value of the entropy. However, we belive that this change can be magnified using some proper prior transformations of the values involved (subject to further work).

The generalization of partition entropy described in [4] could also be an interesting idea of study from the perspective of data warehouse environment.

References

[1] Kimball, R., "Drawing the Line between Dimensional Modeling and ER Modeling Techniques", http://www.dbmsmag.com/9708d15.html, 1997.
[2] Russom, P., Moore C., Teubner, C., "How To Evaluate Enterprise ETL", Forrester Research, 2004.
[3] Shannon, C. E., "A Mathematical Theory of Communication", The Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, July, October, 1948
[4] Simovici, D., "Metric Methods in Data Mining", IDA 2006, Iasi, Romania, June 16, 2006.
[5] Simovici, D., Jaroszewicz, S., "An Axiomatization of Partition Entropy", Transactions on Information Theory, July 2002, vol. 48 (7), pp. 2138–2142.
[6] Ullman, J. D., "Principles of database systems", Computer Science Press, 1980.
[7] Vassiliadis, P., Simitsis, A., Skiadopoulos, S., "Modeling ETL activities as graphs", 4th Intl. Workshop DMDW'2002, Toronto, Canada, May 27, 2002, pp. 52-61.

(1) Computer Science Faculty, "Alexandru Ioan Cuza" University Iasi, General Berthelot, 16, 700483 Iasi, Romania
E-mail address: mbalta@infoiasi.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 290–297

DATA PREDICTIONS USING NEURAL NETWORKS

CĂLIN ENĂCHESCU

Abstract. Financial and economic forecasters have witnessed the recent development of a number of new forecasting models. Traditionally, popular forecasting techniques include regression analysis, time-series, analysis, moving averages and smoothing methods, and numerous judgmental methods. ANN (Artificial Neural Networks) are members of a family of statistical techniques, as are flexible nonlinear regression models, discriminant models, data reduction models, and nonlinear dynamic systems. They are trainable analytic tools that attempt to mimic information processing patterns in the brain. Because they do not necessarily require assumptions about population distribution, economists, mathematicians and statisticians are increasingly using ANN for data analysis.

1. Introduction

Neural computing represents an alternative computational paradigm to the algorithmic one (based on a programmed instruction sequence). Neural computation is inspired by knowledge from neuroscience, though it does not try to be biologically realistic in details [4]. We will deal with neural networks organized in layers, where the information is transmitted from the first layer until the last layer. This type of feedforward multylayer neural networks is called MLP (*MultiLayer Perceptron*) [3]. Some important fact about artificial neural networks:

a) the first layer of neurons is called input layer, it is a simple buffer to store the input data.
b) the input signal is transmitted to the connected (hidden) neurons.
c) the last layer is the output of the system, and is usually called output layer.

MLPs learn in a supervised manner [10]. Learning represents the process in which input patterns are presented repeatedly and the weights are adjusted according to the learning algorithm, which in this supervised case, take the difference between the desired output and the current output into consideration.

2000 *Mathematics Subject Classification.* code, code.
Key words and phrases. Artificial Neural Networks, supervised learning, prediction of data series.

©2007 Babeş-Bolyai University, Cluj-Napoca

2. Mathematical aspects of supervised learning

The training set used for supervised learning has the following form:

$$T = \{(\mathbf{x}_i, \mathbf{z}_i) \mid i = 1, 2, \ldots, N\} \tag{1}$$

where $\mathbf{x}_i \in \mathbf{R}^n$ is the n-dimensional input vector, and $\mathbf{z}_i \in \mathbf{R}^m$ is the m-dimensional target vector that is provided by a trainer. $N \in \mathbf{N}$ is a constant that represents the number of training samples. In the classical supervised learning strategy [5], [10], the trainer is a static agent. Using the probabilistic distribution he selects a certain input vector $\mathbf{x}_i$, and provides the appropriate target vector $\mathbf{z}_i$. The learning algorithm [15] will compute the difference between the output generated by the neural network $\mathbf{y}_i$ and the desired target vector $\mathbf{z}_i$.

The signal error is used to adapt the synaptic weights w_{ji} using a gradient descendent strategy [7]: $w_{ji} = w_{ji} + \eta \dfrac{\partial E}{\partial w_{ji}}$ where $\eta \in (0, 1)$ is the learning rate, controlling the descent slope on the error surface which is corresponding to the error function E [8]: $E = \frac{1}{2} \sum_{i=1}^{N} (y_i - z_i)^2$

3. Data series prediciton with neural networks

Usually, data accessible to the economists are sets of numeric values that describe the situations about the investigated problem in a certain moment. Numeric data sets are called *data series*, and their analysis and values prediction is called the *analysis* respectively the *prediction of data series*. Examples of data series are: the value of monthly unemployment, the value of monthly / annual inflation, the value of different stocks and the value of daily exchange between different currencies [1]. To develop these economic models we have to study the analysis of economic time series, and for the decision process we have to make the prediction for these series, using the developed models. An important class of economic data series is represented by financial data series. These series contain data that represent monetary values of some economic objects or reports on some monetary values of economic objects.

Data processing represents the process of data transformation before building the models. These transformations are conversions, classifications, filtrations or other similar processing. Many times data are not properly structured to build predictive models. In these cases it is important to remove components that represent redundant or irrelevant information [16].

The general purpose of preprocessing is to remove the observable deterministic relations. Theoretically, the purpose is to obtain some data series with mean 0 and a small variation. The first step in data preprocessing is to make comparable the components of the data series. For this purpose it is necessary to rescale the data so that the values of the data series components to have values in the interval

[0,1] or [-1,1]. The second step is to remove the primary deterministic components, that are easy to be observed. Examples of these types of components are trend and seasonality [12]. In the next step we can proceed a data filtering. The purpose of the filtering process is to remove non-trivial periodical components that have dominant effects in data series. To determine those periodical componets we can apply a Fourier transformation of the data series. To filter these componets we can build linear filters. The most common filters are the the low-pass filters, high-pass filters and band-pass filters [10].

Finally, input data vectors are analised to determine possible clustering [9], [13]. This is done through simple classification of data. If we can group the data in precise distinct classes, separate models are built for those classes. It is possible, that after clusterization the models to be equivalent. We must verify model equivalence, and in case of equivalence relations, we must build simplified data general models [14].

By analising the data corresponding to a problem it is possible to build predictive models. It is very important to test the accuracy of the generated model. Through validation we understand the testing of the model and the measurment of its performance using a measure of performance [6]. A method for spliting data in training data and in validation data is the simple division based on data feature. We can consider the data x_t with $t <= T_0$ as training data and data x_t with $t > T_0$ as validation data.

Other techniques for data prediction are based on the auto-regressive model, or on the average sliding model [16].

The prediction of data series is based on the supposition that there exist a functional relation between past, present and future data series values. Usually the supposition is that the functional relation is not completely deterministic, but also contains a stochastic component. In several cases, especially in the case of financial time series, it is assumed that the deterministic component is not dominant, the stochastic component being of great importance.

The performance measurement of predictive models is crucial for their practical application. A common measure regarding predictive models, including neural networks, is the use of square average error. Many times, prediction errors are too large in the context of real applications because of the stochastic components present in the financial series. For this reason, it is necessary to use additional performance measures to test the validity of the predictive models. Neural networks are efficient tools to detect nonlinear relations that rule, at least partially, the behaviour of time series. As the majority of financial data series contain nonlinear components, neural networks are good candidates to predict this type of series.

4. Practical implementation of the neural network

The practical part of this paper is related to the implementation of a neural network that offers the possibility to analyse and predict data series. In this

simulation we have tried to approximate and to predict some financial data series. The parameters that are influencing the learning process are: the training data set, the number of epochs (number of the presentations of the training data set), learning rate, the activation functions for the neurons contained in the hidden layer, the number of neurons in the hidden layer. The architecture of the neural network used in our simulation is corresponding to a MLP neural network with one hidden layer [4]:

- The input layer contains n input neurons, n representing the dimensionality of the input space $x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}) \in \mathbf{R}^n$. The bias can be considered explicitly or implicitly;
- The hidden layer having a number of hidden neurons equal to the dimension of the training set $T = \{x_i, f(x_i)) \mid i = 1, 2, \ldots, N\}$. The activation functions of the hidden neurons are Green functions $G(x - x_k)$ [13]. The dimension of the hidden layer can be reduced using an unsupervised clustering algorithm;
- The output layer contains one single output layer having as activation function a linear function or a special weighted functions of the output values generated by the neurons in the hidden layer [6];

Synaptic weights:

- The weights between the input layer and the hidden layer are included in the form of the activation functions of the hidden neurons. The vector $w = (w_1, w_2, \ldots, w_N)$ represents the weights between the hidden layer and the output layer.

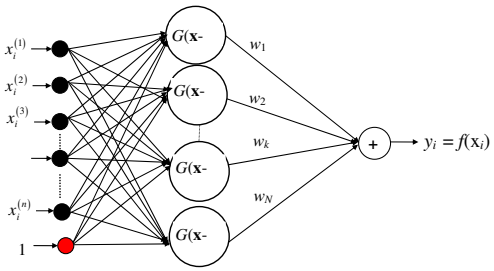


FIGURE 1. Architecture of the neural network used for simulations.

The nerural network was trained using an original learning algorithm, based on the backpropagation learning strategy [8]. For learning we have used a training set containing financial data series, like the exchange rate between RON and EUR or RON and USD. The learning set, corresponds to the time frame January 2004-June 2005, and was obtained from the official Web-site of the National Bank of

Romania http://www.bnr.ro (Banca Nationala a Romaniei). After learning, we have performed a testing phase, in order to measure the accuracy of the trained neural network.

Number of epochs	Learning error
10	0.0289682211941586
50	0.0135544478100808
100	0.00669238729756063
500	0.00268232311020435
1000	0.00192883390014049
5000	0.00163946058914474
10000	0.00157560371972362

TABLE 1. Results of the learning phase: the learning error obtained for different epochs.

In the following graphics (Fig. 2, 3, 4, 5) we have presented the ability of the neural network to approximate and to predict the finnancial data series corresponding to the exchange rate of RON versus EUR. The light-gray curve represents the real exchange rate and the dark-gray curve represents the result generated by the neural network.

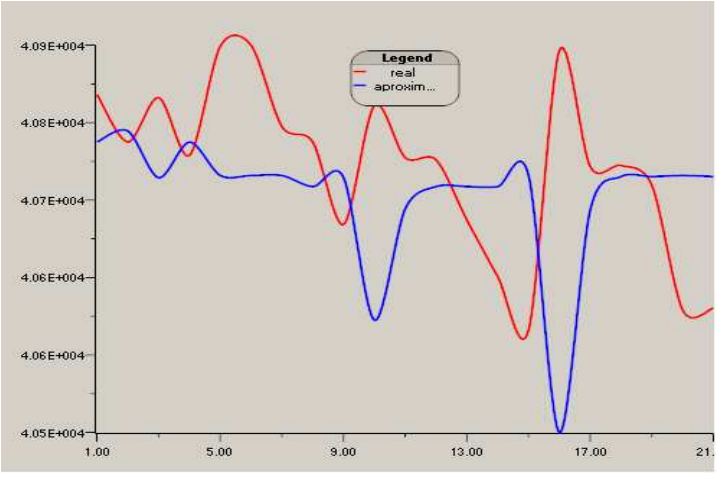


FIGURE 2. Results of the approximation and prediction made by the neural nework, after a learning process of 10 epochs.

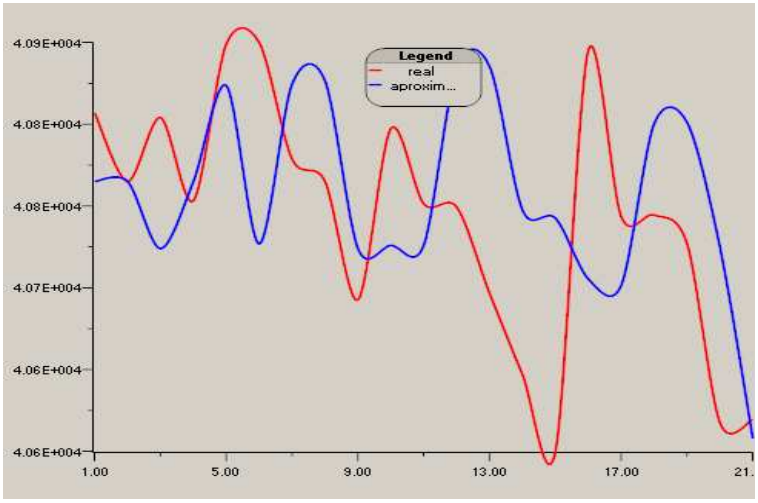


FIGURE 3. Results of the approximation and prediction made by the neural nework, after a learning process of 100 epochs.

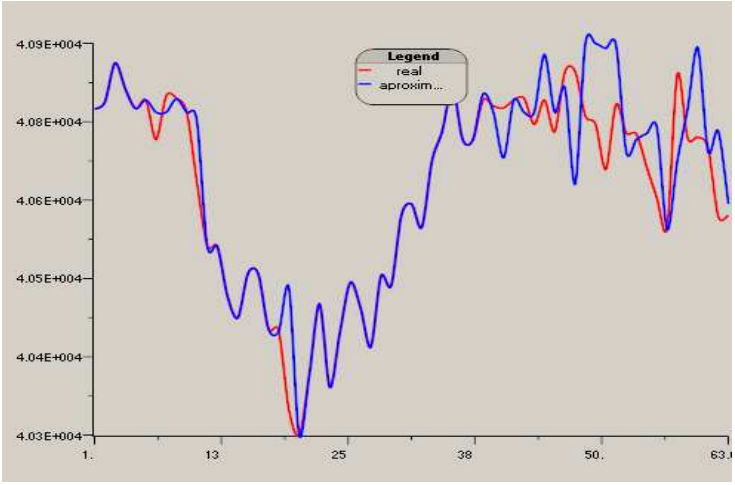


FIGURE 4. Results of the approximation and prediction made by the neural nework, after a learning process of 500 epochs.

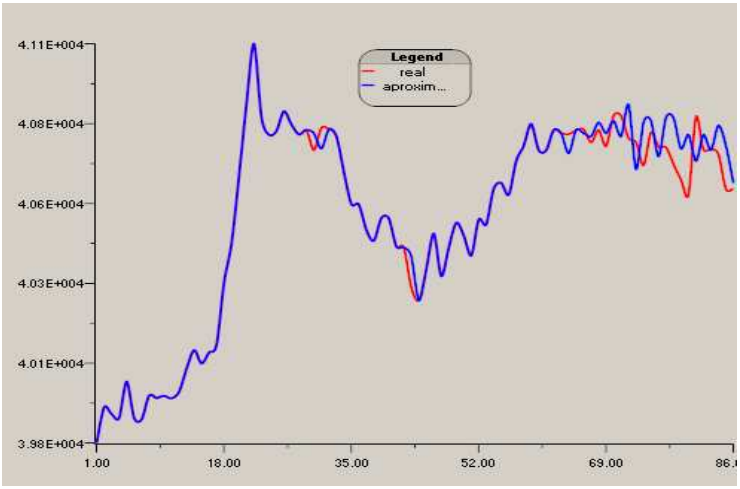


FIGURE 5. Results of the approximation and prediction made by the neural nework, after a learning process of 5000 epochs.

5. Conclusions

The ability to deal with many processing elements makes neural computing faster than conventional computing. In addition, parallelity makes it robust and fault-tolerant in the sense that performance does not degrade significantly even if one of the nodes fails. Researchers are concluding that most economic and financial problems are non-linear; that simple cause-and-effect relationships rarely exist; that, instead, most problems encountered are fuzzy patterns, which relate to multiple variables. There are many useful neural network models for nonlinear data analysis, such as the MLP model, and there is room for many more applications of statistics to neural networks, especially in regard to estimation criteria, optimization algorithms, confidence intervals, diagnostics, and graphical methods. As they do not require an exact specification of the functional equations, emulative neural systems can be applied to predict economic phenomena - especially unrecognized, unstructured, and non-stationary processes. Thus, ANNs (Artificial Neural Networks) are highly suitable for analyzing economic systems. ANNs have proven themselves to be adequate also for searching out and identifying non-linear relationships and for pinpointing those variables that hold the highest predictive value. After extensive training, ANN are able to eliminate substantial amounts of ambiguity in economic forecasts, although never completely overcoming indeterminacy.

References

[1] Adlar, J. Kim, Christian, R. Shelton (2002) Modeling Stock Order Flows and Learning Market-Marking from data, AIM-2002-009, MIT.
[2] András, P. (2000) Reţele neuronale pentru aproximare şi predicţia seriilor de timp, Universitatea "Babeş-Bolyai" Cluj Napoca.
[3] Enăchescu, C. (1998) Bazele teoretice ale reţelelor neuronale, Editura Casa Cărţii de Ştiinţă Cluj-Napoca.
[4] Enăchescu, C. (1997) Elemente de inteligenţă artificială. Calcul neuronal., Universitatea "Petru Maior" Trgu-Mureş, 1997.
[5] Enăchescu, C. (1995) Properties of Neural Networks Learning, 5th International Symposium on Automatic Control and Computer Science, SACCS'95, Vol.2, 273-278, Technical University "Gh. Asachi" of Iasi, Romania.
[6] Enăchescu, C.(1996) Neural Networks as approximation methods. International Conference on Approximation and Optimisation Methods, ICAOR'96, "Babes-Bolyai University", Cluj-Napoca.
[7] Enăchescu, C. (1995) Learning the Neural Networks from the Approximation Theory Perspective. Intelligent Computer Communication ICC'95 Proceedings, 184-187, Technical University of Cluj-Napoca, Romania.
[8] Hecht-Nielsen, R. (1989). Theory of the backpropagation Neural Network. IEEE International Joint Conference on Neural Networks, 1, 593-605.
[9] Girosi, F., T. Pogio (1990). Networks and the Best Approximation Property. Biological Cybernetics, 63, 169-176.
[10] Haykin, S. (1994), Neural Networks. A Comprehensive Foundation. IEEE Press, MacMillian.
[11] Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. Neural Networks, 2, 359-366.
[12] Lackes, R. and Mack D.: Neural Networks. Basics and Applications, Springer-Verlag Berlin Heidelberg, 1998.
[13] Pogio, T., F. Girosi (1990). Networks for Approximation and Learning. Proceedings of the IEEE, Vol. 78, 9, 1481-1497.
[14] Petri, A. J. (1991). A Nonlinear Network Model for continuos learning. Neurocomputing 3, 157-176.
[15] White, D.A., (1989) Learning in artificial neural networks: A statistical perspective. Neural Computation 1, 425-464.
[16] Weigend A.: Time series prediction, Addison-Wesley, 1994.

(1)"Petru Maior" University of Tg. Mures, Nicolae Iorga street nr.1, 540088
E-mail address: ecalin@upm.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 298–306

EVALUATING DYNAMIC CLIENT-DRIVEN ADAPTATION DECISION SUPPORT IN MULTIMEDIA PROXY-CACHES

ADRIAN STERCA, CLAUDIU COBÂRZAN, FLORIAN BOIAN, DARIUS BUFNEA

Abstract. Adaptation of multimedia streams in proxy-caches by usually lowering their quality as a result of a transcoding operation, can yield a lot of benefits in situations when resources like available bandwidth are scarce. Such an operation becomes mandatory when client preferences and/or display capabilities have to be met. We evaluate an alternative to the static specification of terminal capabilities and user preferences inside multimedia proxy-caches, namely the use of *scaling hints*, provided by a protocol still under development, which enables the client to dynamically indicate to proxy-caches or origin servers the appropriate course of action based on network load or user's desires.

1. Introduction

Streamed multimedia data, which continuously gains a bigger percent of the total amount of data transferred over the Internet, tends to stress the existing infrastructure as it demands huge bandwidths (when compared for example with web traffic) and low latencies. Because in most cases, no QoS guaranties can be given, in the rapid fluctuating environment which is the current Internet, availability of multimedia services is certainly a problem. After a successful deployment in the World Wide Web domain, proxy-caches have also been used in the area of multimedia communications (for video on demand services, video broadcasting, etc.). A multimedia cache holds multimedia objects (e.g. movies, audio streams, animations, presentations, etc.) requested by clients in the hope that future requests can be serviced using the already retrieved objects. Such a mechanism can alleviate the availability problem, because multimedia objects are served from the cache through a much better connection (higher bandwidth, smaller delay and jitter), instead of being served from the original streaming server through a fluctuating network line. The gains are not only on the client side but also on the server as an efficient proxy-caching service helps diminish the load imposed on the servers.

2000 *Mathematics Subject Classification.* 68M14, 68M12.
Key words and phrases. Multimedia proxy-caches, Stream adaptation, Streaming protocols.

©2007 Babeş-Bolyai University, Cluj-Napoca

However, there are some situations when the proxy must further reduce its bandwidth usage. These situations can appear when the client's connection to the proxy is overloaded or when the proxy deals with heterogeneous clients (e.g. PDA devices, mobile phones, personal computers, etc.) and dynamically changing user preferences. In these cases, the proxy might choose to adapt the multimedia objects from its cache (instead of sending a new request to the origin server), so that the client is able to use the data it receives and the quality of this data is satisfactory.

The next section presents some other scenarios when adaptation might be useful in order to reduce the total amount of used bandwidth or meet client expectations. Section 3 provides a short overview of some of the techniques that can be used when doing adaptation. Following we present some of the available mechanisms inside streaming protocols that can assist both the client and the proxy-cache/server when deciding to perform an adaptation operation. A description of the test bed we used for the evaluation of those mechanisms as well as a discussion of the obtained results are presented in Section 5 and Section 6. Further away we present our conclusions and indicate possible directions for future work.

2. Use case scenarios

We have already mentioned the case of a multimedia proxy-cache having to service heterogeneous clients (PDAs, mobile phones, desktop computers, tablet PCs, etc.). In order to do this, the proxy-cache has multiple options: request the streams encoded in the appropriate format from the server (the adaptation is done by the server if an appropriate encoded version of the requested file is not available), perform the adaptation operation locally on already cached objects, or, if the object is not available locally and the server is not able to provide the requested quality, perform adaptation on the fly, on the streams it currently receives and delivers to active clients.

Most of the time the decision to perform adaptation is made at the proxy or at the server side before the start of the streaming session, but there are also situations when the client must take such a decision dynamically during the streaming session. During a video conference conference, each participant will receive and playback the streams from other participants and, at the same time, each participant streams the data captured (e.g. from a webcam) at the highest quality permitted by the available bandwidth. As not all the participants speak simultaneously, it would make sense that the client requests that the stream of the active speaker is delivered at full quality while the ones belonging to inactive participants are streamed with minimum quality. In such a case, the proxy could transcode the streams of the inactive clients and deliver them at a lower resolution and/or in grayscale. This would lower the bandwidth consumption while also reducing the burden of displaying all the streams in full quality for the clients with limited

or insufficient computational power and/or with limited display capabilities. Another example is when at some point during the playback of a music video, the user might decide to only listen to the audio track and disregard the video content while busy with another task. At that point he sends an adaptation request to the proxy/server in order to receive only the audio data. A similar scenario, when dynamic client-driven transcoding/adaptation is required, can be identified in a security surveillance system with surveillance cameras sending low quality streams, that switch to a higher quality when triggered by a sensor or by a human operator. Dynamic client-driven adaptation requests can also be used in a client-server/proxy environment as a coarser-granularity replacement for feedback information.

3. Short Survey of Adaptation Techniques

Adaptation of multimedia content means to be able to either enhance or reduce the quality of the data in concordance with the user preferences and the terminal capabilities it specifies.

When it comes to video data (video streams), the most common and frequent operation that is done is to reduce the quality of the video. In the following, we will refer only to situations in which the quality of a video stream is reduced by transcoding operations. By *video transcoding*, one understands the operation of converting a video from one format into another format, where a format is defined by characteristics such as bit-rate, frame-rate, spatial resolution, coding syntax and content. Transcoders can operate both in compressed and pixel-domain, the main difference being that, when operating in the compressed domain, the video data does not need to be decoded, transformed and then re-encoded (like in the pixel domain architectures). This leads to faster but limited (when it comes to the complexity of the operation) transcoding operations in the compressed domain.

In the following we will refer to three types of adaptation: temporal adaptation (which is done in the compressed domain), grayscale and size reduction (which are done in the decompressed or pixel domain). *Temporal adaptation* means dropping frames so that a lower average bitrate of the stream is achieved. *Size reduction* means down-sampling to a lower spatial resolution; the frame rate remains the same while the bitrate is reduced. *Grayscale reduction* drops the chrominance information (U and V in YUV format) obtaining a reduced bitrate (chrominance information makes up to 20% of a bitstream).

4. Dynamic Client-Driven Adaptation Support in Multimedia Streaming Protocols

When speaking about adaptation there are 3 things that have to be considered: who performs the operation, who decides when adaptation should be performed and the moment when the decision is taken.

In a client - server/proxy environment, adaptation can be performed both at the transmitter (i.e. server or proxy) and at the receiver side (i.e. client player). Generally, it is preferred that adaptation is performed at the transmitter's side, because of the following reasons: (a) multimedia adaptation like transcoding can be quite resource-consuming and usually the transmitter has greater computing power than the receiver and (b) an adapted multimedia stream usually has smaller demands for network bandwidth, so if the adaptation is performed at the transmitter, the network's bandwidth is used more efficiently.

The decision to adapt a multimedia stream is usually taken by the transmitter based on feedback from the client and on its current load. However, there are situations (see section 2) when it is necessary that the receiver takes the adaptation decision (even if the adaptation process itself is still carried out by the transmitter).

Regarding the moment when the decision to adapt is taken, this can either be (a) at the beginning of the streaming, during the session negotiation part (e.g. in the case of heterogenous clients, terminal capability negotiation, etc.), or (b) arbitrarily during the streaming session (e.g. session migration, changing user preferences, congested network links, subjective user decision, etc.).

Most of the standard streaming protocols provide little support for dynamic client-driven adaptation. SDP [1] includes partial support for terminal capabilities descriptions, while RTP/RTCP [2] supports sending feedback information (number of packets received/dropped, bandwidth received, etc.), from the client to the server, but provides no support for sending an adaptation decision from the client to the server. Extensions to RTSP [3] for stream switching allow the server to notify the client about stream switching. To our knowledge, none of these streaming protocols have strong support for allowing the client to communicate adaptation requests dynamically to the server. In the rest of this section, we briefly describe the Adaptation-aware Multimedia Streaming Protocol (AMSP), an experimental streaming protocol which provides support for dynamic client-driven adaptation.

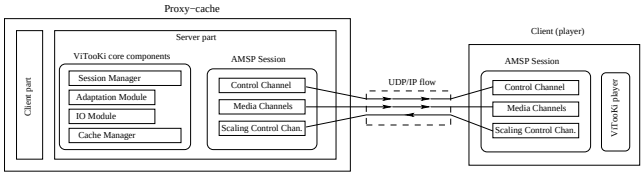


FIGURE 1. The streaming system's architecture

The Adaptation-aware Multimedia Streaming Protocol [4] is a streaming protocol similar to RTP. It conveys time sensitive information like multimedia data together with scaling information, so that multimedia streams can be adapted inside the network according to its rapid changing parameters. The scaling information can be used by common core routers to perform packet-level adaptation

of multimedia streams (i.e. drop less important packets) or by scaling proxies to perform complex media transformations inside the network (e.g. color reduction, temporal reduction, transcoding, etc.). The central concept of AMSP is the channel concept. Each channel is identified by an 8-bit field in the AMSP header called the ChannelID field. There are several types of channels AMSP supports: *control channel* (conveys configuration information global to the AMSP session), *media channels* (deliver multimedia data), *metadata channels* (transport metadata), *scaling control channels* (transport scaling information), *retransmission channels*, *feedback channels* and *auxiliary channels* (transport other types of information). A multimedia stream is mapped onto one or more media channels, and thus, its packets get the priority of the respective channel(s). Basically, an AMSP session is a multiplexation of several AMSP channels (at least the control channel). The Scaling Control Channel of an AMSP session can be used to convey scaling hints or adaptation requests to multimedia proxys or streaming servers to steer the adaptation of multimedia streams according to the network's load or the client's decision. One or more scaling hints are encapsulated in a scaling control channel packet which contains the AMSP header (including the channel id of the packet) and one or several scaling rules. Each scaling rule contains the scaling action (type of adaptation) to be performed (e.g. temporal reduction, size reduction, color reduction, requantization, etc.), the media channels on which this adaptation should be applied, quality and size reduction expected if this scaling rule is applied and payload specific to each rule.

We have evaluated the use of AMSP scaling hints and AMSP scaling control channel to enable dynamic client-driven adaptation decision support in multimedia proxy-caches. The architecture of our streaming environment is described in the following section.

5. The streaming environment used for evaluation

The streaming environment is built around the ViTooKi framework [5]. ViTooKi (The Video ToolKit) is an open source, high-level C++ multimedia library created with the goal of simplifying the implementation of multimedia applications. It offers encoding/decoding support for a number of video and audio formats, streaming via RTP/UDP, various adaptation techniques, meta-data support (MPEG-7 and MPEG-21), session management through RTSP and SDP and also cache management. It also includes some useful applications like a streaming server and a player.

The streaming environment used in our experiments is depicted in Figure 1. We used the AMSP library developed in [4] and integrated AMSP in ViTooKi as an IO streaming class so that we can stream multimedia data using AMSP. We also implemented the scaling control channel which was not implemented in the original AMSP library. The proxy architecture includes ViTooKi core components for managing multimedia sessions and for managing the cache, the adaptation

components for performing adaptation and the IO module for reading and decoding the multimedia content. All these ViTooKi components are not directly related to streaming. For streaming multimedia data the proxy uses an AMSP session which contains three kinds of channels: one control channel (necessary in an AMSP session for configuring the other channels), several media channels for sending multimedia data and a scaling control channel for receiving scaling hints (adaptation requests) from the client.

The experiments presented in the following section show the effects of using AMSP scaling hints on the state of the system (client, proxy and network) using the following metrics: client's perceived quality of multimedia data and the network bandwidth used. We make the following remark on the experiments: due to the fact that we want to show the effects of enabling dynamic client-driven adaptation decisions in multimedia proxy-caches on the network bandwidth used and on the client's perceived quality, we assumed multimedia data was always present in the proxy's cache so that the proxy doesn't have to get the data from the originating streaming server and moreover we ignored all the overhead related to the management of cache objects which is the same as when AMSP scaling hints were not used.

6. Experiments and evaluation

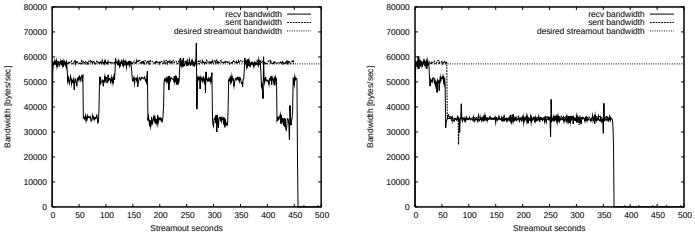
To evaluate the effects of enabling dynamic client-driven adaptation decisions through AMSP scaling hints in multimedia proxy-caches we streamed an MPEG-4 video from the proxy to the client (as depicted in Figure 1), through a traffic-shaped network line. We used a Linux traffic shaper on the network link between the proxy and the client and this traffic shaper changes the available bandwidth every 30 seconds using the following pattern: 75KB/s - 60KB/s - 36.7KB/s - 60KB/s - 75KB/s

So, in the first 30 seconds of streaming, the available bandwidth is 75KB/s, then it drops to 60KB/s, then after another 30 seconds it drops down to 36.7KB/s and at this point, after another 30 seconds, the available bandwidth climbs back to 75KB/s in two 30 seconds long steps. The above pattern is repeated indefinitely. The video stream used in our tests was the MPEG-4 reference stream "Big Show One + Two" with 13,000 frames and a frame rate of 25 fps in CIF resolution. The average bitrate of the stream is 400 kbps, with quantization levels of 28 for B-VOPs and 16 for I-VOPs and P-VOPs. The stream was encoded with the following frame pattern in each one second GOP: IBBBBPBBBBPBBBBPBBBBPBBBB. In order to avoid client buffer underruns, we considered the desired streamout rate of 457.7 kbps (57.2KB/s) which is a little higher than the average bitrate of the stream.

We performed three experiments and in each run we measure the effects on the network bandwidth used and on the client's perceived quality which we measure using PSNR. In the first run, the video is not adapted and it is streamed constantly

at the desired streamout rate of 57.2KB/s no matter what the available bandwidth is. In the second run, in the beginning, the proxy streams the video unadapted, at a constant streamout rate of 57.2KB/s, but around second 65, after the available bandwidth drops to 36.7KB/s, the client sends a color adaptation request through the AMSP scaling control channel and the proxy adapts the video at an average bitrate of 35KB/s using color to grayscale reduction. The third run is exactly like the second one, only that at second 65, temporal adaptation is used to adapt the video stream at 35KB/s.

By employing the aforementioned bandwidth fluctuations and by making the client request adaptation after the available bandwidth drops to 36.7KB/s, we think that our experiments emulate both use cases presented in section 2, i.e. when the client decides to request adaptation based on the degradation of the network conditions (i.e. AMSP scaling hints are used as a coarser-granularity replacement for feedback) and when the client decides to request adaptation based on its own subjective desires.



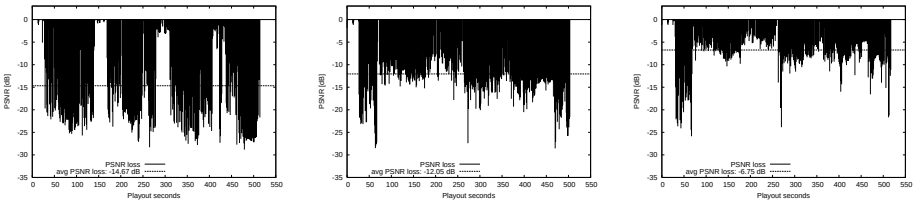
(a) for the unadapted video stream (b) for the color adapted video stream

FIGURE 2. Bandwidth evolution

For the first streaming scenario, when the video is streamed unadapted at a constant streamout rate of 57.2KB/s, Figure 2(a) shows the evolution of the proxy's sent bandwidth and the bandwidth received by the client. We see that although the proxy streams the video at the desired streamout rate of 57.2KB/s, the client receives data at a bandwidth that follows the fluctuations imposed by the traffic shaper. In this streaming scenario, according to Figure 3(a) the stream received by the client loses up to 27 dB PSNR due to lost I-, P- and B-VOPs. The average quality reduction is 14.67 dB.

For the second experiment, although the proxy starts streaming the video at the desired streamout rate (i.e., 57.2KB/s), after 65 seconds, due to a client adaptation request received through the AMSP scaling control channel, the video stream is adapted at a bitrate of 35KB/s using color to grayscale adaptation. The evolution of the sent and received bandwidth is shown in Figure 2(b). It can be seen, that

after second 65, the sent and received bandwidth equal 35KB/s. According to Figure 3(b) the quality degrades with an average of 12.05 dB PSNR due to lost VOPs. We note that, after color adaptation was applied, the quality reductions are less severe than the ones obtained in the first experiment. Please note that in the first 65 seconds, the PSNR loss is the same for all three experiments.



(a) for the unadapted video stream (b) for the color adapted video stream (c) for the temporal adapted video stream

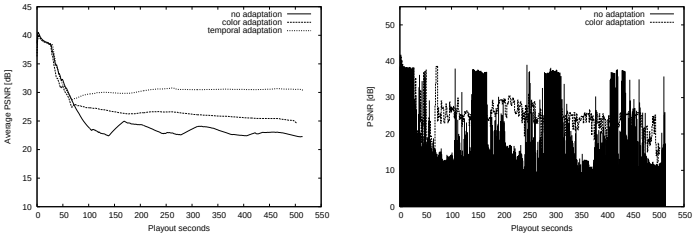
FIGURE 3. Quality loss

The bandwidth evolution for the third experiment, when temporal adaptation is applied on the stream after second 65, is the same as the one depicted in Figure 2(b). According to Figure 3(c) the average quality reduction is 6.75 dB which is less than the quality reductions observed in the first two experiments. This fact is shown more clearly in Figure 4(a) where the average PSNR values obtained for all three experiments are plotted as a function of playout seconds. We can see there, that in the first 65 seconds, the average PSNR values are the same for all three experiments, but after the client decides to request adaptation from the proxy, the temporal adaptation scenario achieves the greater PSNR average, followed by the color adaptation scenario and last, by the scenario when the video was not adapted at all.

We also note that, when the available bandwidth is high (i.e., 75KB/s), the unadapted stream achieves the greatest PSNR values, so the greatest quality. This is shown in Figure 4(b) where the PSNR values obtained for the unadapted stream and the ones obtained for the color adapted stream are compared.

7. Conclusion

We have evaluated the use of dynamic client-driven adaptation decision support in multimedia proxy-caches through the use of AMSP scaling control channel and AMSP scaling hints. Our experiments show that providing dynamic client-driven adaptation support has its benefits on the streaming environment and in some scenarios, it is the only viable solution.



(a) Average PSNR comparison for the three experiments

(b) Comparison of PSNR values for the unadapted and color adapted video stream

FIGURE 4. PSNR comparisons

REFERENCES

[1] M. Handley, V. Jacobson, SDP: Session Description Protocol, RFC 2327, April 1998.
[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 3550, July 2003.
[3] H. Schulzrinne, A. Rao, R. Lanphier, Real Time Streaming Protocol (RTSP), RFC 2326, April 1998.
[4] M. Ohlenroth, Network-Based Adaptation of Multimedia Contents, PhD. Dissertation, University of Klagenfurt, Austria, September 2003.
[5] The ViTooKi Framework, http://vitooki.sourceforge.net/

"BABES-BOLYAI" UNIVERSITY, CLUJ NAPOCA
E-mail address: {forest, claudiu, florin, bufny}@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 307–315

AUTOMATED PROOF OF GEOMETRY THEOREMS INVOLVING ORDER RELATION IN THE FRAME OF THE *THEOREMA* PROJECT

JUDIT ROBU[1]

Abstract. Collins' Cylindrical Algebraic Decomposition method (CAD) can be used to prove geometry theorems that involve order relation (that is, the algebraic form consists of polynomial equalities and inequalities). Unfortunately only very simple geometric statements can be proved this way, as the method is very time consuming. To overcome the slowness of Collins' CAD method for complicated polynomials we propose a method (section 4) that combines the area method for computing geometric quantities and the CAD method. We present an implementation of this method as part of the Geometry Prover in the frame of the *Theorema* project.

1. Introduction

In this paper we present a method for proving geometry theorems that involve order relation (that is, the algebraic form consists of polynomial equalities and inequalities). We deal with a class of statements in plane Euclidean geometry, that we call constructive geometry statements possibly involving inequalities. These statements are constructive statements in the sense presented in [5], but they may also contain further constraints for the constructed points. This way we can also deal with notions like a point on a segment, incircle or interior bisector.

Collins' Cylindrical Algebraic Decomposition method (we shall refer to it as CAD method) introduced in [6], improving earlier work of [9], can be used to prove geometry theorems that involve order relation as has been observed in [7], [1] and other papers. Unfortunately only very simple geometric statements can be proved this way, as the method is very time consuming.

2000 *Mathematics Subject Classification.* 68T15, 68W30.

Key words and phrases. geometry theorem proving, area method, cylindrical algebraic decomposition, *Theorema*.

Sponsored by Austrian FWF (Österreichischer Fonds zur Förderung der Wissenschaftlichen Forschung), Project 1302, in the frame of the SFB (Special Research Area) 013 "Scientific Computing".

©2007 Babeş-Bolyai University, Cluj-Napoca

To overcome the slowness of Collins' CAD method for complicated polynomials we propose a method that combines the area method presented in [5] and the CAD method. First we compute the expressions involved in inequalities using the area method. This way we obtain a new problem equivalent to the original one that is expressed only in terms of the free points (arbitrary points, introduced by a *point* construction, see Definition 1) of the original constructions. Applying the CAD method to this new problem we can obtain the result in reasonable time even for quite complicated problems.

The Geometry Prover is part of *Theorema*, a mathematical software system implemented in *Mathematica* and, hence, available on all computer platforms for which *Mathematica* is available. *Theorema* aims at providing one uniform logical and software technological frame for automated theorem proving in all areas of mathematics or, in other words and more generally, for formal mathematics, i.e. proving, solving, and simplifying mathematical formulae relative to mathematical knowledge bases, see [2], [3]. *Theorema* is being developed at the RISC Institute by the *Theorema* Group under the direction of Bruno Buchberger. For a presentation of *Theorema* compared to other existing provers see [10].

Theorema offers a user-friendly interface for problem input. It generates fully automatically the proofs that contain all the necessary explanations.

The Geometry Prover is based on the methods described in [11], [4], [7], [5]. The input for the geometry prover, i.e. the algebraic formulation of all the construction steps and of the property the final configuration should satisfy is generated automatically from the geometric description of the theorem.

In the next sections we shall

- define the object of our study: the class of constructive geometry statements possibly involving inequalities in plane Euclidean geometry;
- give a brief description of the area method and our notations relative to this method;
- present our method based on combining the area method and Collins' CAD method for proving the above class of statements;
- give an example.

2. Constructive Geometry Statements Possibly Involving Inequalities

In this paper we deal with a class of statements in plane Euclidean geometry. We consider three kinds of geometric objects: points, lines and circles. To make sure that the constructed objects are welldefined, we need to assume some nondegenerate conditions (denoted by ndg conditions in what follows). These conditions are automatically generated by the prover.

A straight line can be defined either by two distinct points, or a point and its direction. So it can be given in one of the following forms:

$line[A, B]$ is the line passing through points A and B.

$pline[C, A, B]$ is the line passing through point C, parallel to $line[A, B]$.

$tline[C, A, B]$ is the line passing through point C, perpendicular to $line[A, B]$. To make sure that all three kinds of lines are welldefined, we need to assume $A \neq B$.

$circle[O, A]$ is the circle with point O as its center and passing through point A. Again, $A \neq O$ has to be assumed.

Definition 1. *A construction is one of the following ways to introduce new points. For each construction we also give its ndg conditions.*

C1: $point[A_1, \ldots, A_n]$. *Take arbitrary points* $A_1, \ldots, A_n$ *in the plain. Each* A_i *is a free point.*

C2: $pon[A, ln]$. *Take a point* A *on line* ln. *The ndg condition of C2 is the ndg condition of line* ln.

C3: $pon[A, circle[O, U]]$. *Take a point* A *on* $circle[O, U]$. *The ndg condition is* $O \neq U$.

C4: $inter[A, ln1, ln2]$. *Let point* A *the intersection of line* $ln1$ *and line* $ln2$. *The ndg condition is* $ln1 \nparallel ln2$.

C5: $inter[A, ln, circle[O, P]]$. *Introduce point* A *as the intersection of line* ln *and* $circle[O, P]$. *The ndg condition is* $P \neq O$ *and line* ln *is not degenerate.*

C6: $inter[A, circle[O_1, P], circle[O_2, P]]$. *Point* A *is the other intersection point of* $circle[O_1, P]$ *and* $circle[O_2, P]$. *The ndg condition is* $P \neq O_1$ *and* $P \neq O_2$.

C7: $pratio[A, W, U, V, r]$. *Take a point* A *on the line passing through* W *and parallel to line* UV *such that* $\overline{WA} = r\overline{UV}$, *where* r *can be a rational number, a rational expression of some geometric quantities, or a variable. The ndg. condition is* $U \neq V$. $\overline{UV}$ *denotes the length of the oriented segment* UV.

C8: $tratio[A, U, V, r]$. *Take a point* A *on the line passing through* U *and perpendicular to line* UV *such that* $\overline{UA} = r\overline{UV}$, *where* r *can be a rational number, a rational expression of some geometric quantities, or a variable. The ndg. condition is* $U \neq V$.

The point A in each construction is said to be introduced by that construction.

Definition 2. *A constructive geometry statement possibly involving inequalities is a list* $S = (C, H, G)$, *where*

1. $C = \{C_1, C_2, \ldots, C_m\}$ *is a construction set. Each* C_i *is a construction such that the point introduced by it must be different from points introduced by* $C_j, j = 1, \ldots, i-1$ *and other points occurring in* C_i *must be introduced before;*
2. $H = \{H_1, H_2, \ldots, H_n\}$ *is a set of additional geometric properties whose algebraic representation may involve inequalities. All the points appearing in* H *have to be introduced by the constructions.*
3. $G = \{G_1, G_2, \ldots, G_k\}$, *the conclusion, is a set of geometric properties of the points introduced by the constructions (it may also contain inequalities).*

3. The Area Method

As a first step of our proof we use the area method, a coordinatefree technique for proving geometry theorems based on point elimination. The basic geometry invariants used are the signed area and Pythagoras differences of oriented triangles and ratios of oriented segments. The method can deal with geometric statements of constructive type, where each new point is introduced by one construction using only previously defined points. The conclusion can be any geometric property that can be expressed by the help of the defined geometric quantities involving only the constructed points. This method is also well-suited for computing geometric expressions that can be expressed using the same set of geometric quantities.

We use capital letters (or combination of capital letters and numbers) to denote points in the Euclidean plane.

We denote by $\bullet \mathrm{L}_{\{A,B\}}$ the length of the oriented segment from A to B and by $\bullet \mathrm{S}_{\{A,B,C\}}$ the signed area of the oriented triangle ABC. For an oriented quadrilateral $ABCD$, we define its area as $\bullet \mathrm{S}_{\{A,B,C,D\}} = \bullet \mathrm{S}_{\{A,B,C\}} + \bullet \mathrm{S}_{\{A,C,D\}}$.

In an oriented triangle ABC the Pythagorean difference $\bullet \mathrm{P}_{\{A,B,C\}}$ is defined as $\bullet \mathrm{P}_{\{A,B,C\}} = \bullet \mathrm{L}_{\{A,B\}}{}^2 + \bullet \mathrm{L}_{\{C,B\}}{}^2 - \bullet \mathrm{L}_{\{A,C\}}{}^2$.

We shall understand by geometric quantities the ratio of the length of two oriented segments on one line or on two parallel lines (denoted by $\bullet \mathrm{R}_{\{A,B,C,D\}}$), the signed area of an oriented triangle or a quadrilateral and the Pythagorean difference of an oriented triangle or a quadrilateral.

We use the elimination lemmas presented in [5].

4. The AreaCAD Method

Our goal is to prove constructive geometry statements possibly involving inequalities. As a first step, we reduce the original proof problem to an equivalent one that makes use only of the points that were introduced as free points. We compute the expressions that appear as additional hypothesis and conclusions eliminating the constructed points using the elimination steps of the area method [5]. Thus we obtain some expressions that depend on the free points and some rational constants denoted by r_i. These constants are introduced by the semibound points and appear when translating the original constructions into constructions accepted by the area prover.

Our original proof problem of finding nondegenerate conditions N such that

$$\begin{aligned}
&\underset{\substack{A_i \\ i=1,\ldots,p}}{\forall} \quad \underset{\substack{B_j \\ j=1,\ldots,m}}{\forall} \quad point[A_1,\ldots,A_p] \wedge \\
&\quad \wedge C_1[A_1,\ldots,A_p,B_1] \wedge \ldots \wedge C_m[A_1,\ldots,A_p,B_1,\ldots,B_m] \wedge \\
&\quad \wedge H_1[A_1,\ldots,A_p,B_1,\ldots,B_m] \wedge \ldots \wedge H_n[A_1,\ldots,A_p,B_1,\ldots,B_m] \wedge \\
&\quad \wedge N[A_1,\ldots,A_p,B_1,\ldots,B_m] \Rightarrow \\
&\quad \Rightarrow G_1[A_1,\ldots,A_p,B_1,\ldots,B_m] \wedge \ldots \wedge G_k[A_1,\ldots,A_p,B_1,\ldots,B_m]
\end{aligned}$$

is transformed to the equivalent problem of finding nondegenerate conditions N' such that

$$\underset{\substack{A_i \\ i=1,\ldots,p}}{\forall} \underset{\substack{r_j \\ j=1,\ldots,q}}{\forall} H'_1[A_1,\ldots,A_p,r_1,\ldots,r_q] \wedge \ldots \wedge H'_n[A_1,\ldots,A_p,r_1,\ldots,r_q] \wedge$$
$$\wedge N'[A_1,\ldots,A_p,r_1,\ldots,r_q] \Rightarrow$$
$$\Rightarrow G'_1[A_1,\ldots,A_p,r_1,\ldots,r_q] \wedge \ldots \wedge G'_k[A_1,\ldots,A_p,r_1,\ldots,r_q]$$

As a second step we have to prove this statement using the CAD method. For the CAD algorithm we have to transform the obtained problem into polynomial form. To obtain as simple expressions as possible we choose for the origin of the coordinate system the point with the highest number of occurrences in the expressions. The next point is taken as being on the $x$axis. We may take this point as having coordinates $\{1, 0\}$. This way the algebraic expressions become even simpler.

If the denominator of an obtained conclusion expression not being 0 does not result from the hypothesis this should be considered a non-degenerate condition and added to the hypothesis. At the end of the proof the user has to analyze whether the obtained condition is a non-degenerate condition or it introduces some essentially new hypothesis. If the simplified expressions contain square roots even powers of subexpressions should be extracted from the square roots adding the necessary conditions.

An implication is true if its conclusion is true or if the hypotheses are contradictory. In this second case we get no information on the logical value of the conclusion. Thus we check first the consistency of the hypothesis by an existential quantifier elimination, then the validity of the universally quantified expression is checked. For this purpose we use the built-in *Mathematica* functions.

5. Example

Let O be the circumcenter, I the incenter, G the centroid and H the orthocenter of a triangle. Then $OG \leq OI \leq OH$.

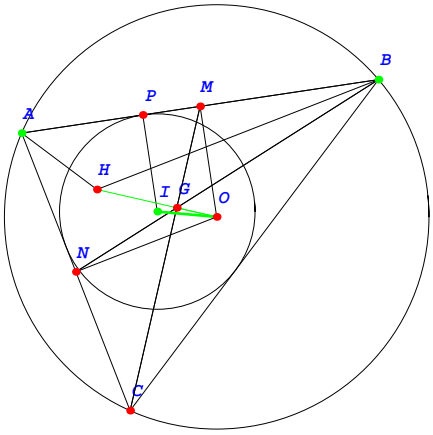
The input for the prover is a *Theorema* Proposition:

```
Proposition["Cad_113_27", any[I, A, B, C, P, H, M, N, O, G],
incircle[I, A, B, C, P] ∧ inter[H, tline[A, B, C], tline[B, A, C]] ∧
   median[C, M, A, B, C] ∧ median[B, N, A, B, C] ∧
   inter[O, tline[M, A, B], tline[N, A, C]] ∧ circle[O, A] ∧
   inter[G, line[C, M], line[B, N]]
   ⇒ inequation[seglengthsq[O, H] − seglengthsq[O, I] ≥ 0] ∧
      inequation[seglengthsq[O, I] − seglengthsq[O, G] ≥ 0]]
```

To display graphically the geometrical constraints among the involved points and lines we call function `Simplify`:

`Simplify[Proposition["Cad_113_27"], by → GraphicSimplifier]`

and obtain the output:



$\overline{HO}^2 \geq \overline{IO}^2$ for this configuration of the points
$\overline{IO}^2 \geq \overline{GO}^2$ for this configuration of the points

FIGURE 1. *Theorema* output

The geometry prover is invoked in the usual *Theorema* manner, specifying the AreaCAD prover, *Theorema* does the rest of the work:

- finds the convenient constructions recognized by the Area Prover;
- invoking the Area Prover computes the geometric expressions representing the constraints and conclusion;
- expresses the obtained new problem as a universally quantified boolean combination of polynomial equalities and inequalities, using the cartesian coordinates of the free points;
- invokes the *Mathematica* `ExistsRealQ` function to verify the consistency of the hypothsis, and then the `Resolve` function to find the trueness of the universally quantified formula;
- generates the notebook with all the explained details of the proof.

For the function call

```
Prove[Proposition["Cad_113_27"], by→GeometryProver, ProverOptions→
   {Method→"AreaCAD"}]
```

we obtain the following output from the prover:

=== Begin of Theorema notebook ===

We have to prove:

$$\underset{I,A,B,C,P,H,M,N,O,G}{\forall}(\text{incircle}[I,A,B,C,P]\wedge\text{inter}[H,\text{tline}[A,B,C],\text{tline}[B,A,C]]\wedge$$
$$\text{midpont}[M,A,B]\wedge\text{midpoint}[N,A,C]\wedge\text{inter}[O,\text{tline}[M,A,B],\text{tline}[N,A,C]]\wedge$$
$$\text{circle}[O,A]\wedge\text{inter}[G,\text{line}[C,M],\text{line}[B,N]]\Rightarrow$$
$$\text{inequation}[\text{seglengthsq}[O,H]-\text{seglengthsq}[O,I]\geq 0]\wedge$$
$$\text{inequation}[\text{seglengthsq}[O,I]-\text{seglengthsq}[O,G]\geq 0]$$

with no assumptions.

As the proposition contains inequalities, we have to use the CAD method. We shall use the area method first to obtain a simpler input for CAD.

First we have to transform the problem in internal form for the AreaCAD. We have to prove, that constructions

$\{A,B,I\}$ free points

$\alpha_A B\perp AI$ and $\alpha_A\in AI$ with ndg. $A\neq I$

$\alpha 1_A B\parallel B\alpha_A$ and $\frac{\overline{B\alpha 1_A}}{\overline{B\alpha_A}}=2$ with ndg. $B\neq\alpha_A$

$\alpha_B A\perp BI$ and $\alpha_B\in BI$ with ndg. $B\neq I$

$\alpha 1_B A\parallel A\alpha_B$ and $\frac{\overline{A\alpha 1_B}}{\overline{A\alpha_B}}=2$ with ndg. $A\neq\alpha_B$

$C=A\alpha 1_A\cap B\alpha 1_B$ with ndg. $A\neq\alpha 1_A, B\neq\alpha 1_B, A\alpha 1_A\nparallel B\alpha 1_B$

$PI\perp AB$ and $P\in AB$ with ndg. $A\neq B$

$\alpha_C I\perp AC$ and $\alpha_C\in AC$ with ndg. $A\neq C$

$\gamma_H B\perp AC$ and $\gamma_H\in AC$ with ndg. $A\neq C$

$\beta_H A\perp BC$ and $\beta_H\in BC$ with ndg. $B\neq C$

$H=A\beta_H\cap B\gamma_H$ with ndg. $\beta_H\neq A,\gamma_H\neq B,\beta_H A\nparallel\gamma_H B$

$MA\parallel AB$ and $\frac{\overline{AM}}{\overline{AB}}=\frac{1}{2}$ with ndg. $A\neq B$

$NA\parallel AC$ and $\frac{\overline{AN}}{\overline{AC}}=\frac{1}{2}$ with ndg. $A\neq C$

$N\gamma_O\perp NA$ and $\frac{\overline{N\gamma_O}}{\overline{NA}}=r_{23}$ with ndg. $A\neq N$

$\beta_O\perp MA$ and $\frac{\overline{M\beta_O}}{\overline{MA}}=r_{22}$ with ndg. $A\neq M$

$O=M\beta_O\cap N\gamma_O$ with ndg. $\beta_O\neq M,\gamma_O\neq N,\beta_O M\nparallel\gamma_O N$

$G=CM\cap BN$ with ndg. $C\neq M, B\neq N, CM\nparallel BN$

with additional constraints

$\bullet R_{\{A,P,P,B\}}>0$

$\bullet R_{\{A,\alpha_C,\alpha_C,C\}}>0$

imply that $\overline{HO}^2-\overline{IO}^2\geq 0$ and $-\overline{GO}^2+\overline{IO}^2\geq 0$

Step 1

Now we try to obtain a simpler form of the constraints and of the conclusion using the area method.

Explanations

========= details about the area method =========

Expression 1

We have to compute $\bullet R_{\{A,P,P,B\}}$

========= details of the computation steps =========

We obtain:

$$\bullet R_{\{A,P,P,B\}}=E1=\frac{\bullet L^2_{\{A,B\}}+\bullet L^2_{\{A,I\}}-\bullet L^2_{\{B,I\}}}{\bullet L^2_{\{A,B\}}-\bullet L^2_{\{A,I\}}+\bullet L^2_{\{B,I\}}}$$

Expression 2

We have to compute $\bullet R_{\{A,\alpha_C,\alpha_C,C\}}$

========= details of the computation steps =========

We obtain:

$$\bullet R_{\{A,\alpha_C,\alpha_C,C\}}=E2=\frac{-\bullet L^4_{\{A,B\}}+\bullet L^4_{\{A,I\}}+2\bullet L^2_{\{A,B\}}\bullet L^2_{\{B,I\}}-\bullet L^4_{\{B,I\}}}{\bullet L^4_{\{A,B\}}+\left(\bullet L^2_{\{A,I\}}-\bullet L^2_{\{B,I\}}\right)^2-2\bullet L^2_{\{A,B\}}\bullet L^2_{\{A,I\}}+\bullet L^2_{\{B,I\}}}$$

Expression 3

We have to compute $\overline{HO}^2-\overline{IO}^2$

========= details of the computation steps =========

We obtain:

$\overline{HO}^2 - \overline{IO}^2 = \bullet P_{\{H,O,H\}} - \bullet P_{\{I,O,I\}} = E3 =$

$-2(\bullet L^{12}_{\{A,B\}} - 4 \bullet L^{10}_{\{A,B\}} \bullet (L^2_{\{A,I\}} + \bullet L^2_{\{B,I\}}) + (\bullet L^2_{\{A,I\}} - \bullet L^2_{\{B,I\}})^4$
$(\bullet L^4_{\{A,I\}} + \bullet L^4_{\{B,I\}}) + 7 \bullet L^8_{\{A,B\}}(\bullet L^4_{\{A,I\}} + \bullet L^4_{\{B,I\}} + \bullet L^2_{\{A,I\}} \bullet L^2_{\{B,I\}}) -$
$\bullet L^2_{\{A,B\}}(\bullet L^2_{\{A,I\}} - \bullet L^2_{\{B,I\}})^2$
$(4 \bullet L^6_{\{A,I\}} + 4 \bullet L^6_{\{B,I\}} - \bullet L^2_{\{A,I\}} \bullet L^4_{\{B,I\}} - \bullet L^4_{\{A,I\}} \bullet L^2_{\{B,I\}}) -$
$\bullet L^6_{\{A,B\}}(8 \bullet L^6_{\{A,I\}} + 8 \bullet L^6_{\{B,I\}} + \bullet L^2_{\{A,I\}} \bullet L^4_{\{B,I\}} + \bullet L^4_{\{A,I\}} \bullet L^2_{\{B,I\}}) +$
$7 \bullet L^4_{\{A,B\}}(\bullet L^8_{\{A,I\}} + \bullet L^8_{\{B,I\}} - \bullet L^2_{\{A,I\}} \bullet L^6_{\{B,I\}} - \bullet L^6_{\{A,I\}} \bullet L^2_{\{B,I\}}))/$
$(\bullet L^2_{\{A,B\}}(- \bullet L^2_{\{A,B\}} + \bullet L^2_{\{A,I\}} + \bullet L^2_{\{B,I\}})^2$
$(\bullet L^4_{\{A,B\}} + (\bullet L^2_{\{A,I\}} - \bullet L^2_{\{B,I\}})^2 - 2 \bullet L^2_{\{A,B\}}(\bullet L^2_{\{A,I\}} + \bullet L^2_{\{B,I\}})))$

Expression 4

We have to compute $-\overline{GO}^2 + \overline{IO}^2$

========= details of the computation steps =========

We obtain:

$-\overline{GO}^2 + \overline{IO}^2 = \bullet P_{\{I,O,I\}} - \bullet P_{\{O,G,O\}} = E4 =$
$2(\bullet L^8_{\{A,B\}} - 2 \bullet L^6_{\{A,B\}}(\bullet L^2_{\{A,I\}} + \bullet L^2_{\{B,I\}}) + (\bullet L^2_{\{A,I\}} - \bullet L^2_{\{B,I\}})^2(\bullet L^4_{\{A,I\}} + \bullet L^4_{\{B,I\}}) +$
$\bullet L^4_{\{A,B\}}(2 \bullet L^4_{\{A,I\}} + 2 \bullet L^4_{\{B,I\}} - 7 \bullet L^2_{\{A,I\}} \bullet L^2_{\{B,I\}}) +$
$\bullet L^2_{\{A,B\}}(-2 \bullet L^6_{\{A,I\}} - 2 \bullet L^6_{\{B,I\}} + 11 \bullet L^2_{\{A,I\}} \bullet L^4_{\{B,I\}} + 11 \bullet L^4_{\{A,I\}} \bullet L^2_{\{B,I\}}))/$
$(9 \bullet L^2_{\{A,B\}}(- \bullet L^2_{\{A,B\}} + \bullet L^2_{\{A,I\}} + \bullet L^2_{\{B,I\}})^2)$

Step 2

Our initial problem becomes

$\underset{A,B,I}{\forall}(E1 > 0 \wedge E2 > 0 \Rightarrow E3 \geq 0 \wedge E4 \geq 0)$

Choosing a cartesian coordinate system with the x-axis $\{I, B\}$ and performing substitution: $\{x_I \to 0, y_I \to 0, x_B \to 1, y_B \to 0, x_A \to u_1, y_A \to u_2\}$ the problem becomes:

$\underset{u_1,u_2}{\forall}(\frac{-(-u_1+u_1^2+u_2^2)}{-1+u_1} > 0 \wedge \frac{u_1(u_1^2+u_2^2-u_1)}{u_2^2} > 0 \wedge u_1 \neq 0 \wedge u_2 \neq 0 \wedge -1 + u_1 \neq 0 \Rightarrow$
$\frac{(u_1^8-u_1^7+6u_1^6+u_1^4(1-2u_2^4)+u_2^4(-1+u_2^2)^2-2u_1^2u_2^2-u_1(u_2^4+u_2^6)-u_1^5(4+u_2^2)+u_1^3u_2^2(3+2u_2^2))}{(u_1^2u_2^2(1+u_1^2+u_2^2-2u_1))} \geq 0 \wedge$
$\frac{(-3u_1^5+4u_1^6+4u_2^4+u_1^2(4-6u_2^2+4u_2^4)+u_1^3(-3+2u_2^2)+u_1^4(-2+8u_2^2)+5u_1(u_2^2+u_2^4))}{(9u_1^2(1+u_1^2+u_2^2-2u_1))} \geq 0)$

Applying the CAD algorithm to this expression we obtain that the proposition is true.

=== End of Theorema notebook ===

6. Concluding Remarks

In this paper we presented a method for proving a class of plain Euclidean geometry theorems involving order relation. The algebraic methods for proving geometry theorems (Gröbner bases, characteristic sets) are very efficient, but they cannot deal with polynomial inequalities, that is geometry statemnts that involve order relation. On the other hand, Collins' CAD algorithm can deal with inequalities, but it is very slow for a system of polynomial equations and inequations with many variables (usually more than 15 for a nontrivial theorem). We propose a method that combines the point elimination used in the area method with Collins' CAD. We used our method to prove several quite difficult theorems, obtaining the result in reasonable time (at most some minutes).

References

[1] Buchberger, B., Collins, G.E., Kutzler, B.: Algebraic methods for geometric reasoning. *Ann. Rev. Comp. Sci.*, 3, page 85−119, 1988.

[2] Buchberger, B., Jebelean, T., Kriftner, F., Marin, M., Tomuta E., Vasaru, D.: An overview on the Theorema project. In: W. Kuechlin (ed.), *Proceedings of ISSAC'97 (International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii, July 2123, 1997)*. ACM Press 1997.

[3] Buchberger, B., Dupre, C., Jebelean, T., Kriftner, F., Nakagawa, K., Vasaru, D., Windsteiger, W.: The Theorema Project: A Progress Report, In: Kerber, M. and Kohlhase, M. (eds.): *Symbolic Computation and Automated Reasoning: The Calculemus-2000 Symposium (Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, August 6-7, 2000, St. Andrews, Scotland)*. A K Peters Ltd 2001.

[4] Chou, S.C.: *Mechanical geometry theorem proving*. Dordrecht Boston: Reidel 1988.

[5] Chou, S.-C., Gao, X.-S., Zhang, J.-Z.: Automated generation of readable proofs with geometric invariants I & II. *J. Automat. Reason.*, 17, page 325−370, 1996.

[6] Collins, G. E.:Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Springer's LNCS*, 33, page 134−165, 1975.

[7] Kutzler, B., Stifter, S.: *New approaches to computerized proofs of geometry theorems*. RISC 86−0.5 RISC-Linz, 1986.

[8] Robu, J.: *Geometry Theorem Proving in the Frame of the Theorema Project*, RISC-Linz Report Series No. 02−23, PhD thesis, 2002.

[9] Tarski, A.: *A decision method for elementary algebra and geometry*, Univ. of California Press, Berkeley Los Angeles, 2nd edition, 1951.

[10] Windsteiger, W., Buchberger, B., Rosenkranz, M.: Theorema. In: The Seventeen Provers of the World, Freek Wiedijk (ed.), *Springer's LNAI*, 3600, page 96−107, 2006.

[11] Wu, W.t.: Basic principles of mechanical theorem proving in elementary geometries. *J. Automat. Reason.* **2**, page 221−252 (1986).

(1) Babeş-Bolyai University, str. Kogalniceanu 1, 400084 Cluj-Napoca, Romania
E-mail address: robu@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 316–323

A HIERARCHICAL CLUSTERING ALGORITHM FOR SOFTWARE DESIGN IMPROVEMENT

ISTVAN GERGELY CZIBULA[(1)] AND GABRIELA ŞERBAN[(2)]

Abstract. *Refactoring* is a process that helps to maintain the internal software quality, during the whole software lifecycle. The aim of this paper is to present a new hierarchical clustering algorithm that can be used for improving software systems design. *Clustering* is used in order to recondition the class structure of a software system. The proposed approach can be useful for assisting software engineers in their daily works of refactoring software systems. We evaluate our approach using the open source case study JHotDraw ([13]), providing a comparison with previous approaches.

1. Introduction

The structure of a software system has a major impact on the maintanability of the system. That is why continuous restructurings of the code are needed, otherwise the system becomes difficult to understand and change, and therefore it is often costly to maintain.

In order to keep the software structure clean and easy to maintain, most modern software development methodologies (extreme programming and other agile methodologies) use refactoring to continuously improve the system structure.

In [7], Fowler defines refactoring as "the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure. It is a disciplined way to clean up code that minimizes the chances of introducing bugs". Refactoring is viewed as a way to improve the design of the code after it has been written. Software developers have to identify parts of code having a negative impact on the system's maintainability, and apply appropriate refactorings in order to remove the so called "bad-smells" ([11]).

In this paper we propose a new hierarchical clustering algorithm that would help developers to identify the appropriate refactorings. Our approach takes an existing software and reassembles it using hierarchical clustering, in order to obtain a better

2000 *Mathematics Subject Classification.* 68N99, 62H30.
Key words and phrases. Software Engineering, Refactoring, Hierarchical Clustering.

©2007 Babeş-Bolyai University, Cluj-Napoca

design, suggesting the needed refactorings. Applying the proposed refactorings remains the decision of the software engineer.

The rest of the paper is structured as follows. Section 2 presents the main aspect related to the problem of *clustering*. The clustering approach (*CARD*) for determining refactorings, that we have previously introduced in [1], is presented in Section 3. A new hierarchical clustering algorithm for identifying refactorings is introduced in Section 4. Section 5 provides an experimental evaluation of our approach. A comparison of the proposed approach with other similar approaches is given in Section 6. Conclusions and further work are given in Section 7.

2. Clustering

Unsupervised classification, or clustering, as it is more often referred as, is a data mining activity that aims to differentiate groups (classes or clusters) inside a given set of objects ([6]), being considered the most important *unsupervised learning* problem. The resulting subsets or groups, distinct and non-empty, are to be built so that the objects within each cluster are more closely related to one another than objects assigned to different clusters. Central to the clustering process is the notion of degree of similarity (or dissimilarity) between the objects.

Let $\mathcal{O} = \{O_1, O_2, \ldots, O_n\}$ be the set of objects to be clustered. The measure used for discriminating objects can be any *metric* or *semi-metric* function $d : \mathcal{O} \times \mathcal{O} \longrightarrow \Re$. The distance expresses the dissimilarity between objects.

In this paper we are focusing only on *hierarchical clustering*, that is why an overview of the hierarchical clustering methods is presented. Hierarchical clustering methods represent a major class of clustering techniques. There are two styles of hierarchical clustering algorithms. Given a set of n objects, the agglomerative (bottom-up) methods begin with n singletons (sets with one element), merging them until a single cluster is obtained. At each step, the most similar two clusters are chosen for merging. The divisive (top-down) methods start from one cluster containing all n objects and split it until n clusters are obtained.

The agglomerative clustering algorithms that were proposed in the literature differ in the way the two most similar clusters are calculated and the linkage-metric used (single, complete or average) ([10]).

3. Background. Refactorings Determination using Clustering

In this section we describe the clustering approach (*CARD*) introduced in [1] in order to find adequate refactorings to improve the structure of software systems. Our aim is, that based on the approach from [1], to introduce a new hierarchical clustering algorithm, that is why a brief description of *CARD* is given below.

In [1], a software system S is viewed as a set $S = \{s_1, s_2, ..., s_n\}$, where $s_i, 1 \leq i \leq n$ can be an application class, a method from a class or an attribute from a class. *CARD* consists of three steps:

- **Data collection.** The existing software system is analyzed in order to extract from it the relevant entities: classes, methods, attributes and the existent relationships between them.
- **Grouping.** The set of entities extracted at the previous step are re-grouped in clusters using a partitioning algorithm (*HARED* algorithm, in our approach). The goal of this step is to obtain an improved structure of the existing software system.
- **Refactorings extraction.** The newly obtained software structure is compared with the original structure in order to provide a list of refactorings which transform the original structure into an improved one.

A more detailed description of *CARD* is given in [1]. At the **Grouping** step of *CARD*, the software system S has to be re-grouped. This re-grouping is represented as a ***partition*** of S, $\mathcal{K} = \{K_1, K_2, ..., K_v\}$. In the following, we will refer to K_i as the i-th *cluster* of $\mathcal{K}$, and to an element s_i from S as an *entity*. A cluster K_i from the partition $\mathcal{K}$ represents an application class in the new structure of the software system.

4. A new Hierarchical Clustering Algorithm for Refactorings Determination - HARED

Based on the clustering approach *CARD* described in Section 3, we present in this section a new hierarchical clustering algorithm for refactoring determination (*HARED - Hierarchical Algorithm for Refactorings Determination*). This algorithm can be used in the **Grouping** step of *CARD*, in order to find an improved structure of the software system S.

HARED is based on the idea of hierarchical agglomerative clustering, but uses an heuristic for merging two clusters. We use *average link* as linkage metric, because we have obtained better results with this metric.

The heuristic used in *HARED* is that, at a given step, the most two similar clusters (the pair of clusters that have the smallest distance between them) are merged only if the distance between them is less or equal to a given threshold, $distMin$. This means that the entities from the two clusters are close enough in order to be placed in the same cluster (application class). This heuristic is particular to our approach and it will provide a good enough choice for merging two application classes.

In our clustering approach, the objects to be clustered are the entities from the software system S, i.e., $\mathcal{O} = \{s_1, s_2, \ldots, s_n\}$. Our focus is to group similar entities from S in order to obtain high cohesive groups (clusters).

We will adapt the generic cohesion measure introduced in [8] that is connected with the theory of similarity and dissimilarity. In our view, this cohesion measure is the most appropriate to our goal. We will consider the dissimilarity degree between

any two entities from the software system S. Consequently, we will consider the distance $d(s_i, s_j)$ between two entities s_i and s_j as expressed in Equation (1).

$$d(s_i, s_j) = \begin{cases} 1 - \frac{|p(s_i) \cap p(s_j)|}{|p(s_i) \cup p(s_j)|} & if \;\; p(s_i) \cap p(s_j) \neq \emptyset \\ \infty & otherwise \end{cases}, \tag{1}$$

where, for a given entity $e \in S$, $p(e)$ represents a set of relevant properties of e, defined as:

- If e is an attribute, then $p(e)$ consists of: the attribute itself, the application class where the attribute is defined, and all methods from S that access the attribute.
- If e is a method, then $p(e)$ consists of: the method itself, the application class where the method is defined, and all attributes from S accessed by the method.
- If e is a class, then $p(e)$ consists of: the application class itself, and all attributes and methods defined in the class.

Based on the definition of distance d given in Equation (1) it can be easily proved that d is a semi-metric function. We will consider the distance $dist(k, k')$ between two clusters $k \in \mathcal{K}$ and $k' \in \mathcal{K}$ as given in Equation (2).

$$dist(k, k') = \frac{1}{|k| \cdot |k'|} \cdot \sum_{e \in k, e' \in k'} d(e, e') \tag{2}$$

The main steps of *HARED* algorithm are:

- Each entity from the software system is put in its own cluster (singleton).
- The following steps are repeated until the partition of methods remains unchanged (no more clusters can be selected for merging):
  - select the two most similar clusters from the current partition, i.e, the pair of clusters that minimize the distance from Equation (2). Let us denote by $dmin$ the distance between the most similar clusters K_i and K_j;
  - if $dmin \leq distMin$ (the given threshold), then clusters K_i and K_j will be merged, otherwise the partition remains unchanged.

We give next *HARED* algorithm.

`Algorithm HARED is`
Input: - `the software system` $\mathcal{S} = \{s_1, \ldots, s_n\}, n \geq 2$,
- `the semi-metric` d `between entities,`
- $distMin > 0$ `the threshold for merging the clusters.`

Output: - `the partition` $\mathcal{K} = \{K_1, K_2, ..., K_p\}$, `the new structure of` $\mathcal{S}$.
Begin
`For i` $\leftarrow$ `1 to n do`

```
    Ki ← {si} //each entity is put in its own cluster
  endfor
  K ← {K1, ..., Kn} //the initial partition
  change ← true
  While change do //while K changes
    dmin ← dist(K1, K2) //the minimum distance between clusters
    For i* ← 1 to n-1 do //the most similar clusters are chosen
      For j* ← i* + 1 to n do
        d ← dist(Ki*, Kj*)
        If d < dmin then
          dmin ← d; i ← i*; j ← j*
        endif
      endfor
    endfor
    If dmin ≤ distMin then
      Knew ← Ki ∪ Kj; K ← (K \ {Ki, Kj}) ∪ {Knew}
    else
      change ← false //the partition remains unchanged
    endif
  endwhile
End.
```

In our approach we have chosen the value 1 for the threshold $distMin$, because distances greater than 1 are obtained only for unrelated entities (Equation (1)).

4.1. **Refactorings Extraction.** In this section we briefly discuss about the refactorings that *HARED* algorithm is able to identify.

Let us consider that S is the analyzed software system, and that $\mathcal{K} = \{K_1, K_2, \ldots, K_p\}$ is the partition provided by *HARED*, i.e., the new structure of S. The main refactorings identified by *HARED* algorithm are given below.

***Move Method* ([7]) refactoring**. It moves a method m of a class C to another class C' that uses the method most. The bad smell motivating this refactoring is that a method uses or is used by more features of another class than the class in which it is defined ([5]). This refactoring is identified by *HARED* by moving the method m in the cluster K_t corresponding to the class C'.

***Move Attribute* ([7]) refactoring**. It moves an attribute a of a class C to another class C' that uses the attribute most. The bad smell motivating this refactoring is that an attribute is used by another class more than the class in which it is defined ([5]). This refactoring is identified by *HARED* algorithm by moving the attribute a in the cluster K_t corresponding to the class C'.

***Inline Class* ([7]) refactoring**. It moves all members of a class C into another class C' and deletes the old class. The bad smell motivating this refactoring is

that a class is not doing very much ([5]). This refactoring is identified by *HARED* algorithm by decreasing the number of elements in the partition $\mathcal{K}$. Consequently, the number of application classes in the new structure of S is decreased, and classes C and C' with their corresponding entities (methods and attributes) will be merged in the same cluster K_t.

Extract Class **([7]) refactoring**. Creates a new class C and move some cohesive attributes and methods into the new class. The bad smell motivating this refactoring is that one class offers too much functionality that should be provided by at least two classes ([5]). This refactoring is identified by *HARED* algorithm by increasing the number of elements in the partition $\mathcal{K}$. Consequently, a new cluster appears, corresponding to a new application class in the new structure of S.

5. Experimental Evaluation

In order to validate our clustering approach, we consider as case study the open source software JHotDraw, version 5.1 ([13]). It is a Java GUI framework for technical and structured graphics, developed by Erich Gamma and Thomas Eggenschwiler, as a design exercise for using design patterns. It consists of **173** classes, **1375** methods and **475** attributes.

At the *Data collection* step of *CARD*, in order to extract from the system the input data for *HARED* algorithm, we use ASM 3.0 ([3]). ASM is a Java bytecode manipulation framework. We use this framework in order to extract the structure of the system (attributes, methods, classes and relationships between entities).

The reason for choosing JHotDraw as a case study is that it is well-known as a good example for the use of design patterns and as a good design. Our focus is to test the accuracy of *HARED* algorithm introduced in Section 4 on JHotDraw, i.e., how accurate are the results obtained after applying *HARED* algorithm in comparison to the current design of JHotDraw. As JHotDraw has a good class structure, the *Grouping* step of *CARD* should generate a nearly identical class structure. After applying *HARED* we have obtained the following results:

(i) The algorithm obtains a new class after the re-grouping step, meaning that an *Extract Class* refactoring is suggested. The methods which are placed in the new class are: **PertFigure.handles**, **GroupFigure.handles**, **TextFigure.handles**, **StandardDrawing.handles**.

(ii) There are two misplaced attributes, **ColorEntry.fColor** and **ColorEntry.fName** which are placed in **ColorMap** class. This means that two *Move Attribute* refactorings are suggested.

(iii) There are four misplaced methods, **UngroupCommand.execute**, **FigureTransferCommand.insertFigures**, **SendToBackCommand.execute**, and **BringToFrontCommand.execute** which are placed in **StandardDrawing** class.

In our view, the refactorings identified at (i) and (ii) can be justified.

- All the methods enumerated at (i) provide similar functionality ([13]), so, in our view, these methods can be extracted in a new class in order to avoid duplicated code, applying *Extract Class* refactoring.
- **ColorMap** and **ColorEntry** are two classes defined in the same source file. **ColorMap** is an utility class which manages the default colors used in the application. **ColorEntry** is a simple class used only by **ColorMap**, that is why, in our view, **fColor** and **fName** attributes can be placed in either of the two classes.

6. Related Work

There are various approaches in the literature in the field of *refactoring*. The only approach on the topic studied in this paper, that partially gives the results obtained on a relevant case study (like JHotDraw) is [2]. The authors use an evolutionary algorithm in order to obtain a list of refactorings using JHotDraw.

The advantages of *HARED* algorithm in comparison with the approach presented in [2] are illustrated bellow:

- In the technique from [2] there are **10** misplaced methods, while in our approach there are only **4** misplaced methods.
- Our technique is deterministic, in comparison with the approach from [2]. The evolutionary algorithm from [2] is executed **10** times, in order to judge how stable are the results.
- The overall running time for the technique from [2] is about **300** minutes (30 minutes for one run), while *HARED* algorithm provide the results in about **3.5** minutes (the execution was made on similar computers).
- As the results are provided in a reasonable time, our approach can be used by developers in their daily work for improving software systems.

We cannot make a complete comparison with other refactoring approaches, because, for most of them, the obtained results for relevant case studies are not available. Most approaches (like [4], [12]) give only short examples indicating the obtained refactorings. Other techniques address particular refactorings: the one in [4] focuses on automated support only for identifying ill-structured or low cohesive functions and the technique in [12] focuses on system decomposition into subsystems.

7. Conclusions and Future Work

We have presented in this paper, based on the approach from [1], a new hierarchical clustering algorithm (*HARED*) that can be used for improving systems design. We have demonstrated the potential of our approach by applying it to the open source case study JHotDraw and we have also presented the advantages of our approach in comparison with existing approaches.

Further work can be done in the following directions:

- To apply *HARED* for other relevant case studies.
- To use other approaches for clustering, such as search based clustering ([9]), or genetic clustering.
- To develop a tool (as a plugin for Eclipse) that is based on the approach presented in this paper.

References

[1] Czibula, I.G., Serban, G.: Improving Systems Design Using a Clustering Approach. International Journal of Computer Science and Network Security, VOL.6, No.12 (2006) 40–49
[2] Seng, O., Stammel, J., Burkhart, D.: Search-Based Determination of Refactorings for Improving the Class Structure of Object-Oriented Systems. Proceedings of GECCO'06 (2006) 1909–1916
[3] http://asm.objectweb.org/ (2006)
[4] Xu, X., Lung, C.H., Zaman, M., Srinivasan, A.: Program Restructuring Through Clustering Technique. In: 4th IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2004), USA (2004) 75–84
[5] Simon, F., Steinbruckner, F., Lewerentz, C.: Metrics based refactoring. In: Proc. European Conf. Software Maintenance and Reengineering. IEEE Computer Society Press (2001) 30-38
[6] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001)
[7] Fowler, M.: Improving the design of existing code. Addison-Wesley, New-York (1999)
[8] Simon, F., Loffler, S., Lewerentz, C.: Distance based cohesion measuring. In Proceedings of the 2nd European Software Measurement Conference (FESMA) 99, Technologisch Instituut Amsterdam (1999)
[9] Doval, D., Mancoridis, S., Mitchell, B.S.: Automatic clustering of software systems using a genetic algorithm. IEEE Proceedings of the 1999 Int. Conf. on Software Tools and Engineering Practice STEP'99 (1999)
[10] Jain, A., Murty, M.N., Flynn, P.: Data clustering: A review. ACM Computing Surveys **31** (1999) 264–323
[11] McCormick, H., Malveau, R.: Antipatterns: Refactoring Software, Architectures, and Projects in Crises. John Wiley and Sons (1998)
[12] Lung, C.H.: Software Architecture Recovery and Restructuring through Clustering Techniques. ISAW3, Orlando, SUA (1998) 101–104
[13] JHotDraw Project: http://sourceforge.net/projects/jhotdraw (1997)

(1) Department of Computer Science, Babeş-Bolyai University, 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: istvanc@cs.ubbcluj.ro

(2) Department of Computer Science, Babeş-Bolyai University, 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: gabis@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 324–331

METRICS-BASED SELECTION OF A COMPONENT ASSEMBLY

CAMELIA SERBAN[(1)] AND ANDREEA VESCAN[(1)]

Abstract. The work of integrating the components with each other and with the rest of the system is the most important part of the component-based development process. The interaction among components in an assembly is essential to the overall quality of the system. When integrating components into a system assembly, it would be useful to predict how the quality attributes for the whole system will be. In order to predict and to asses quality attributes, the usage of software metrics is a necessity.

Software metrics that follow the assembly-centric evaluation approach are used to select (from all obtained assemblies) the solution that best represents the system requirements.

1. Introduction

The goal of software composition is to find a good combination of components that leads to a software system that responds to client-specific requirements. Software composition promises components reuse and therefore productivity gains, because of shorter time-to-market and improved quality. Two steps need to be done when a system is to be constructed from a set of components: building all possible configurations and then analyzing each of these assemblies in order to obtain the final solution.

The long-term success of Component-Based Development (CBD) depends on the ability to predict the quality of the obtained systems. For this reason researchers and practitioners are keen on developing techniques for efficient component selection and composition [3]. Issues related to developing a composition theory include determining how to predict the properties of assemblies, how to measure properties of components, how to verify the measurements, and how to communicate the property values to component users.

The realm of software metrics includes proposals for both product and process assessment. In this paper, we are concerned with product metrics, with a focus on metrics for component selection and composition.

2000 *Mathematics Subject Classification.* 68N30, 68N19.

Key words and phrases. software metrics, component assembly, assemblies quality attributes prediction.

©2007 Babeş-Bolyai University, Cluj-Napoca

Problem statement. A set of specified components and the system requirements that we want to develop are given. In our previous work we had developed an algorithm [10] that obtains all the possible and correct system configurations from the given components. In order to decide which solution to choose (that best represents the system requirements) we use metrics to assess quality attributes that are of interest for assembly evaluation.

The remainder of this paper is organized as follow. After this introduction, section 2 presents our previous and current view on component and component composition. Next, section 3 presents a collection of assembly metrics that are relevant for measuring the quality attributes which we are interested in. An example and result analysis are given in Section 4. The paper finishes by drawing some conclusions and outlining further research activities.

2. Component Specification Elements and Composition

There are many similar but not identical definitions of components, although the basic idea seems to be the same. All definitions highlight the basic characteristics of a component: it is an independent software module; provides a functionality, but is not a complete system; can be accessed only through its interface and can be incorporated in a software system without regard to how it is implemented.

2.1. Component Specification Elements.

In this section our previous approach on component specification elements are presented. There are two kind of components: simple and compound [4]. A simple component can have many input data and many output data that represent the parameters of the functionality (only one) being implemented. A compound component is a group of connected components, in which the output of a component is used as input by another component from this group. Two particular simple components are the source [1] and the destination [2] component.

Another previous approach of component specification involved the following characteristics: component *id* and *interface*. The interface of the component should describe all the exported features of the component. The services provided by the component are seen as functions, so the interface specifies a list of function signatures. In this paper we separate the interface of a component in provided interface and required interface.

2.2. Components Composition Reasoning.

The goal of software composition is to find a good combination of components that leads to a software system that responds to client-specific requirements [1]. The components assembly process consists of building a set of all possible configurations with the given candidate

[1] A source component, i.e. a component without inports, is a component that generates data provided as outports in order to be processed by other components.

[2] A destination component, i.e. a component without outports, is a component that receives data from the system as its inports and usually displays it, but it does not produce any output.

components. A configuration is constructed by adding based on the data dependencies (provided and required services) a candidate component [10].

3. Some relevant suite of metrics

In order to assess, in an quantitative manner, some quality attributes that are considered important for our system, we need to define a set of metrics that measure these attributes. Before we define metrics, we need to know the type of information that is available about the entities we plan to asses (the software components and the assembly). The fact that components are black box and binary units of composition, whose internals cannot be viewed or accessed, only leaves us with externally observable elements of the component that allow to assess its quality. The quality attributes that we decided to measure are: reusability, functionality, understandability, maintainability and testability.

In this section we present a collection of software component metrics that are relevant in measuring the quality attributes stated above. All these metrics are context-dependent: a given component will have different metrics values depending on the particular architectural configuration in which it is placed.

3.1. Metrics Overview in CBD.

In this section we present metrics already introduced by other researchers. In [6] the metrics described above were formalized in OCL and a comparison was made.

In [7] Hoek et al. proposed metrics to asses service utilization in component assemblies. The metrics follow the assembly-centric evaluation approach.

Definition 1. ***Component service utilization metrics.*** [7] *The Provided Services Utilization (PSU) represents the ratio of services provided by the component which are actually used (Equation 1 - left side). The Required Services Utilization (RSU) is similar, but for required services (Equation 1 - right side).*

$$PSU_X = \frac{P_{actual}}{P_{total}} \qquad RSU_X = \frac{R_{actual}}{R_{total}} \tag{1}$$

where: P_{actual} = number of services provided by component X that are actually used by other components and P_{total} = number of services provided by component X; R_{actual} = number of services required by component X that are actually provided by the assembly and R_{total} = number of services required by component X.

Definition 2. ***Compound Service Utilization metrics.*** [7] *The Compound Provided Service Utilization (CPSU) represents the ratio of services provided by the components in the assembly which are actually used (Equation 2 -left side). The Compound Required Service Utilization (CRSU) is similar, but for services required by the components (Equation 2 - right side).*

$$CPSU_X = \frac{\sum_{i=1}^{n} P^i_{actual}}{\sum_{i=1}^{n} P^i_{total}} \qquad CRSU_X = \frac{\sum_{i=1}^{n} R^i_{actual}}{\sum_{i=1}^{n} R^i_{total}} \tag{2}$$

where P^i_{actual}= number of services provided by component i that are actually used by other components and P^i_{total} = number of services provided by component i; R^i_{actual} = number of services required by component i that are actually provided by the assembly and R^i_{total}= number of services required by component i.

In [8] Narasimhan and Hendradjaya proposed metrics to asses component interaction density (a measure of the complexity of relationships with other components).

Definition 3. ***Interaction density of a component.*** [8] *The Interaction Density of a Component (IDC) is defined as a ratio of actual interactions and potential ones (Equation 3). The Incoming and Outgoing Interaction Density of a Component (IIDC and OIDC, respectively) are similar, but considering only incoming interactions (Equation 4 - left side) or outgoing ones (Equation 4 - right side).*

$$IDC = \frac{\#I}{\#I_{max}} \tag{3}$$

where $\#I$ = Actual Interactions and $\#I_{max}$ = Maximum available interactions.

$$IIDC = \frac{\#I_{IN}}{\#I_{maxIN}} \qquad OIDC = \frac{\#I_{OUT}}{\#I_{maxOUT}} \tag{4}$$

where $\#I_{IN}$ = Actual incoming interactions and $\#I_{maxIN}$ = Maximum available incoming interactions and $\#I_{OUT}$ = Actual outgoing interactions and $\#I_{maxOUT}$ = Maximum available outgoing interactions.

Definition 4. ***Average Interaction Density of Software Components.*** [8] *The Average Interaction Density of Software Components (AIDC) represents the sum of IDC for each component divided by the number of components.*

$$AIDC = \frac{IDC_1 + IDC_2 + IDC_n}{\#components} \tag{5}$$

where IDC_i = IDC of component i and $\#components$ = number of components in the system.

In our previous work [9] a component assembly was view as a graph (transformed in a dependences tree). This approach enabled us to define new metrics for depth and breadth components hierarchy (measuring dependences calls between components).

3.2. **Proposed Metrics.** We propose the following two metrics for measuring coupling between components.

Definition 5. ***Component Coupling Grade.*** *The Component Coupling Grade (CCG) of a component X witch is dependent by a component Y, represents the number of services provided by Y that X uses. In what follows we will denote this value with CCG(X,Y).*

Definition 6. ***Component Coupling Total Grade.*** *The Component Coupling Total Grade (CCTG) of a component X which is dependent by a set of components C_1,C_2,...,C_n, represents the number of services provided by all these components that X uses.*

$$CCTG = CCG(X, C_1) + CCG(X, C_2) + ... + CCG(X, C_n). \tag{6}$$

3.3. **The influence of metrics values on quality attributes.** We stated before that our aim is to define metrics that are relevant in measuring the quality attributes which we are interested in. We need these informations for choosing the solution that best represents the system requirements. Table 1 presents the influence of metrics values on the quality attributes witch we consider important for the assembly evaluation. We use the following notations: m for metric low value, M for hight value of the metric, $+$ for positive influence and $-$ for negative influence. For example a low value of IDC influences positively the reusability of the component.

TABLE 1. The influence of metrics values on quality attributes

	Reusability	Functionality	Understandability	Maintainability	Testability
PSU	m/+	m/-	m/+	m/+	m/+
RSU	m/+	-	m/+	m/+	m/+
CPSU	m/+	m/-	m/+	m/+	m/+
CRSU	m/+	-	m/+	m/+	m/+
IDC	m/+	m/-	m/+	m/+	m/+
IIDC	m/+	-	m/+	m/+	m/+
OIDC	m/+	m/-	m/+	m/+	m/+
AIDC	m/+	-	m/+	m/+	m/+
CCG	M/-	M/+	M/-	M/-	M/-
CCTG	M/-	M/+	M/-	M/-	M/-

A threshold is a limit (high or low) placed on a specific metric. All the above metric values scale between 1 and 0, except the CCTG and CCG. We set the value of the threshold at 0.5. In our future work we will apply precise methods in choosing the threshold value.

4. Example and Result Analysis

In this section we present an example to illustrate the above metrics and our approach for the best solution selection based on metrics.

The system designer, during the requirements analysis phase, grouped the input and output data of the system into three required interfaces and two provided interfaces. The first step of how configurations can be built consists of selecting from a repository the set of components that may potentially participate in the final system. In this example, nine components have been found as candidates. We add two more components to complete the final system: a *Read* (R) and a $Write$ (W) component. The algorithm [5, 10] provided several solutions. For the purpose of this paper we only present two of them and discuss the different metrics values for each system-solution and their influences on the quality attributes.

The first solution is represented in figure 1 - right side. From the set of selected candidate-components this solution contains only six of them (without taking into consideration the R and W components).

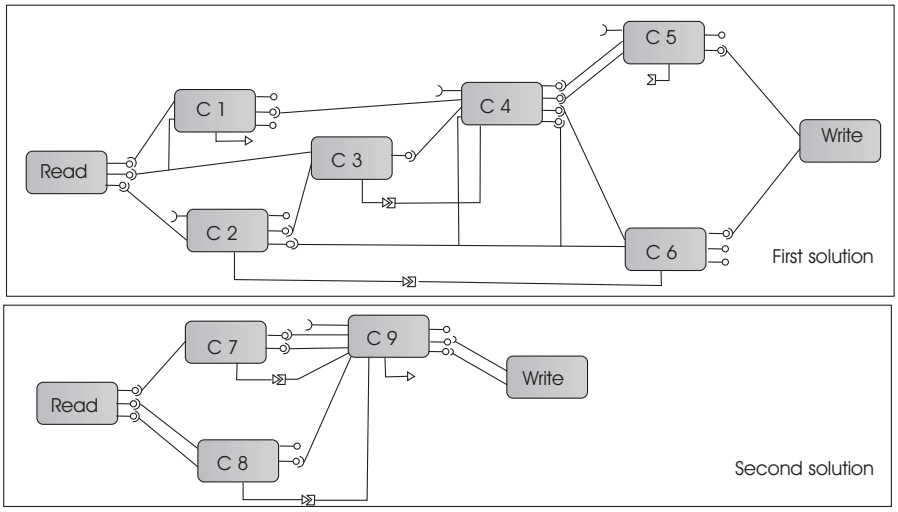


Figure 1. System first and second solutions

The values of the metrics for each component in the final system are presented in table 2 and the assembly value metrics in table 3. The tables show that the solution described above has the values for the metrics around the medium value, for all quality attributes. For example, the majority of the components have a very high functionality in the system (PSU, CPSU, OIDC and IDC values are very close to 1) and at the same time they can offer new functionalities for the future improvement by adding new provided services (influences the maintainability attribute).

	PSU	RSU	IDC	IIDC	OIDC	CCTG
C_1	0.33	1	0.5	1	0.25	2
C_2	0.66	0.50	0.66	0.5	0.75	1
C_3	1	1	1	1	1	2
C_4	1	0.75	0.88	0.80	1	3
C_5	0.50	0.66	0.50	0.50	0.50	2
C_6	0.33	1	0.71	1	0.33	3
C_R	1	–	1	–	1	–
C_W	–	1	1	1	–	2

TABLE 2. System first solution metrics values

CPSU	CRSU	AIDC
0.68	0.82	0.77

TABLE 3. Assembly metrics values

Regarding the coupling metrics we can remark that there is a maximum limit that is not very high and we can say that the maintainability and reusability are not strongly influenced. The assembly values metrics suggest that the solution is not considered to be the "best" for every quality attribute, but a medium "best" solution for the overall system. The value of the $AIDC$ metric is close to 1 but we must take also into consideration the $CCTG$ metric to decide which solution best represents our future needs (if we would like to improve and add new functionality or if we just want to have a good functionality for the system).

The second solution chosen to be represented is depicted in figure 1 - left side. The solution contains only three internal components form the set of candidate-components. The values of the metrics for each component in the final system are presented in table 4 and the assembly value metrics in table 5

	PSU	RSU	IDC	IIDC	OIDC	CCTG
C_7	1	1	1	1	1	1
C_8	0.50	1	0.80	1	0.66	2
C_9	0.66	0.75	0.70	0.83	0.50	3
C_R	1	–	1	–	1	–
C_W	–	1	1	1	–	2

TABLE 4. System second solution metrics values

CPSU	CRSU	AIDC
0.80	0.88	0.90

TABLE 5. Assembly metrics values

The metrics values that influence the functionality attribute are close to 1 revealing a good functionality of each component inside the system, but the other metrics values influence negatively the other quality attributes. The 0.50 chosen threshold is exceeded for all the computed metrics. In table 1 we can see that a high value influences negatively almost all the quality attributes discussed. The

values of $CCTG$ metric are relatively high considering that there are few components in the solution. The $CCTG$ value for the ninth component is considered to be high yielding a very hard understandability, testability and maintainability.

5. Conclusions and Future work

Software metrics provide a quantitative means to control the quality of software. After building all possible configurations (component assemblies) from a given set of specified components, the designer has to decide which solution to use further. We discussed and proposed in this paper some quality attributes to consider when analyze the quality of an assembly. Software metrics are used to select the solution, among all obtained configurations, that best represents the system requirements.

We set the value of the metrics threshold at 0.5. In our future work we will apply precise methods in choosing the threshold value. There are different methods like statistic-based and even genetic-based algorithms.

References

[1] Crnkovic, I., *Component-based software engineering - new challenges in software development*, Software Focus, John Wiley & Sons, 2001

[2] Crnkovic, I., Larsson, M., *Building Reliable Component-Based Software Systems*, Artech House publisher, 2002

[3] Crnkovic, I., Schmidt, H., Stafford, J. A., Wallnau, K., *The 6^{th} ICSE Workshop on Component-Based Software Engineering: Automated Reasoning and Prediction*, ACM SIGSOFT Software Engineering Notes, vol 29, nr 3, pp.1-7, 2004

[4] Fanea, A., Motogna, S., *A Formal Model For Component Composition*, Proceedings of the Symposium Cluj-Napoca Academic Days, pp. 160 - 167, 2004

[5] Fanea, A., Motogna, S., Dioşan, L., *Automata-Based Component Composition Analysis*, Studia Universitas "Babes-Bolyai", Seria Informatica, vol. L (1), pp. 13 - 20, 2006

[6] Goulo,M. A., Abreu, F. B., *Composition Assessment Metrics for CBSE*, 31st Euromicro Conference, Component-Based Software Engineering Track, Porto, Portugal, IEEE Computer Society, pp. 96 - 103, 2005

[7] Hoek, A. v. d., Dincel, E., and Medvidovic, N., *Using Service Utilization Metrics to Assess and Improve Product Line Architectures*, 9th IEEE International Software Metrics Symposium (Metrics'2003), Sydney, Australia, 2003

[8] Narasimhan, V. L. and Hendradjaya, B., *A New Suite of Metrics for the Integration of Software Components*, The First International Workshop on Object Systems and Software Architectures (WOSSA'2004), Australia, 2004

[9] Serban, C., Vescan, A., *Metrics for Component-Based System Development*, Creative Mathematics and Informatics, Vol. 16,pp. 143-150, 2007

[10] Vescan, A., Motogna, S., *Syntactic automata-based component composition*, The 32nd EUROMICRO Software Engineering and Advanced Applications (SEAA), Work in Progress, 2006

(1) Computer Science Department, Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania

E-mail address: {camelia, avescan}@cs.ubbcluj.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 332–340

ARCHITECTING AND SPECIFYING A SOFTWARE COMPONENT USING UML

DRAGOŞ PETRAŞCU[(1)] AND VLADIELA PETRAŞCU[(2)]

Abstract. The current paper experiments a component-based approach for the *LCD Wallet Travelling Clock* case study. It proposes a component architecture and tries to formally specify its building blocks using UML and OCL. Following this architecture and specifications, a JavaBeans implementation has also been developed.

1. Introduction

Software components can be thought of as *units of composition with contractually specified interfaces and explicit context dependencies only* [6]. Ideally, they should be black boxes, enabling third parties to reuse them without knowing the details of their inner structure [4]. The interfaces they provide should be the only access points. Therefore, specifying precisely these interfaces becomes of utmost importance for both clients (which rely solely on that specification when accessing a component's services) and implementors (which are provided with an abstract definition of a component's internal structure).

At least three levels can be identified when specifying software components: syntactic, semantic, and nonfunctional. Without disregarding its significance, the last of these is not covered by our current paper. As for the syntactic specification, it conforms to the following general model: A component implements a set of named interfaces (also known as *provided* or *incoming interfaces*), while making use of the services offered by another set of named interfaces (the *required* or *outgoing interfaces*). An interface consists of a set of named operations, each of which has a number of named parameters (in, out, or inout). Types are associated to all parameters. But even though it is the basis for client code type checking and component interoperability verifications, this kind of specification is missing semantic information. Nothing can be said about the effects of invoking one of the services exposed by an interface, except what might be guessed from the name given to

2000 *Mathematics Subject Classification.* 68N30, 68Q60, 68N19.
Key words and phrases. component specification, UML, JavaBeans.

©2007 Babeş-Bolyai University, Cluj-Napoca

that operation and the names and types of its parameters [4]. Nevertheless, this is the only type of specification used with dedicated component-based approaches as COM, CORBA or JavaBeans (COM and CORBA use different dialects of the IDL for syntactic specification, while JavaBeans uses the Java programming language).

So as to overcome the above mentioned defficiencies, several techniques for components semantic specification were provided in the literature. Most of them propose a *design by contract* approach in order to formally describe the semantics of the services offered by a software component. We have followed the general guidelines given in [2], which introduces a process inspired, on its turn, by Catalysis [5].

2. Component specification using UML

The approach proposed by [2] enhances the general syntactic model introduced earlier with new concepts. The core one is that of a *contract*. Two types of contracts can be distinguished when specifying software components: usage contracts and realization contracts.

A *usage contract* is a run-time contract between an interface offered by a component object and its clients. Each such contract is represented by an *interface specification* that consists of the following:

- all the services that compose the interface with their signatures and associated behavior;
- the interface's information (state) model and any constraints (invariants) on that model.

The information model associated to an interface is an abstraction of that part of a component's state that affects or may be affected by the execution of operations in the interface. It does not impose any implementation restrictions. It is merely an abstraction that helps in specifying operations behavior. Each such operation is considered as a fine-grained contract in its own right. Behavior is described in terms of pre/post-condition pairs. A precondition is an assertion that must be true before the operation is invoked. It is the client's responsibility to ensure that it holds prior to making the call. It is a predicate expressed in terms of the input parameters and the state model. The postcondition is guaranteed by the component's implementor, after the execution finishes, provided that the precondition was met. It is also a predicate, involving both input and output parameters, as well as the state just before the invocation and just after.

A usage contract is represented by means of an Interface Specification Diagram, with associated constraints. An Interface Specification Diagram is a usual UML Class Diagram, just enriched with some specific stereotypes. It contains the interface to specify and its information model. The interface (including the signatures of its provided services) is figured as a classs having the `<<interface`

type>> stereotype. The information model is represented by a collection of associated types (classes), at least one of them having a composition relationship with the interface. The types that are part of the information model, excepting the built-in ones, are stereotyped as <<info type>>s. All the constraints (operations pre/post-conditions and information model invariants) are formalized using OCL.

For illustration purpose, we consider an ISpellCheck interface [4], implemented by a simple SpellChecker component. ISpellCheck offers a single method, isCorrect, that checks whether a certain word is correctly spelled or not. The interface specification makes use of a basic information model, consisting of a set of strings. There are no invariants associated to this model. The corresponding usage contract is presented in figure 1.

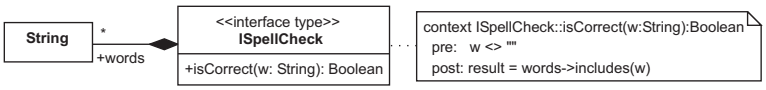


FIGURE 1. ISpellCheck - Interface Specification Diagram

While a usage contract is a run-time contract with the client, a *realization contract* is a design-time contract with the component implementor. The realization contract is represented by the entire *component specification*, that, apart from the usage contracts, includes information regarding the following:

- all the interfaces provided and required by the future component;
- inter-interface constraints (relationships between the information models of the different interfaces);
- interaction scenarios with other components, required in order to implement the provided services.

The required and provided interfaces are illustrated by means of a Component Specification Diagram. This is another variation of the UML Class Diagram, an example being given in figure 2a. Componenent Specification Diagrams form the building blocks of Arhitecture Specification Diagrams for component-based systems.

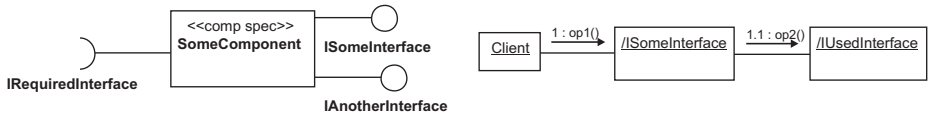


FIGURE 2. a)Component Specification Diagram b)Collaboration Diagram

When a component implements several interfaces and/or requires the services offered by other interfaces, certain relationships may exist between their associated

information models. These inter-interface relationships can be expressed using OCL constraints. Suppose that the above mentioned `SpellChecker` component implements another interface, `ICustomSpellCheck`, having a state model that also consists of a collection `words` of strings, and offering services that allow a potential client to add (remove) words in (from) the collection. Then, the fact that the two interfaces work on the same state model can be expressed in OCL as follows: `ISpellCheck::words = ICustomSpellCheck::words`.

Required interactions with other components, in order to implement the provided services, are best described using UML Collaboration Diagrams, as the one in figure 2b.

3. Case study: LCD Wallet Travelling Clock

3.1. **General requirements.** We have followed the above mentioned component specification guidelines in a case study. The object of our case study was the LCD Wallet Travelling Clock (already introduced in [3]), which we have tried to approach (architect, specify and implement) from a component perspective.

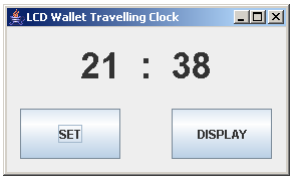


Figure 3. LCD Wallet Travelling Clock

Next, we give a brief description of the clock's requirements. There are two kinds of events that influence its behavior. Firstly, there is an internal *tick* event (generated from inside the clock by an inner ticker) whose occurence causes the time to advance by one second. The same event controls the showing or hiding of the vertical dots separating the two main display sections of the screen. Secondly, there may be external events, generated by the user depressing one of the two buttons offered by the clock (the *display* button and the *set* one). By starting in the default state in which the clock is showing the current time (under the format `hour:minute`) and repeteadly depressing the display button, the display states are went through: date display (under the format `month_day`), seconds display (under the format `_:seconds`, the underscore character indicating an empty display zone), then again time display and so on. Analogously, by starting in the default display state and repeteadly depressing the set button, the setting states are visited: month setting, day setting, hour setting, minute setting, then again time display and so on. While in a setting state, the depressing of the display button causes the value of the component to be set (month, day, hour or minute) to increment by one unit.

3.2. **Component architecture and specification.** Our aim was to give a JavaBeans component implementation for the LCD Wallet Travelling Clock. By analysing the general behavioral requirements described earlier, we have decided to factor the functionality offered by the clock in two provided interfaces: one that allows client code to send `pressDisplayButton` and `pressSetButton` requests, named `IClockKeyboard`, and the other, `IReadOnlyClockDisplay`, used to obtain the values to be shown on a screen-like part of a user interface. In order to accomplish this, `IReadOnlyClockDisplay` exposes the following three operations: `getLeftSide`, `getMiddle`, and `getRightSide`. Besides, the clock component requires the services provided by a `PropertyChangeListener`, in order to notify the user interface about changes occured in the displayed values. The graphical component has the ability to register/unregister itself as a clock listener, by means of `[add|remove]PropertyChangeListener` services, also offered by the `IReadOnlyClockDisplay` interface. All the mentioned interfaces, as well as the dependencies they cause among the clock component and the `ClockFrame` (playing the role of a visual interface) are represented on the architecture specification diagram in figure 4.

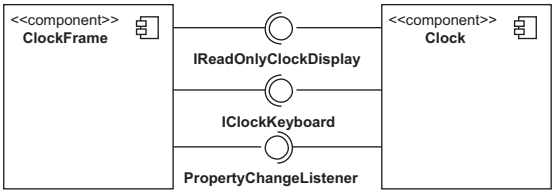


FIGURE 4. External view of *Clock* component

By now, we have offered an external (user) view of our clock component: the interfaces it provides and requires and the way it interracts with its environment. Following, figure 5 shows its internal structure through a component-connector-port architecture. Both diagrams employ the new UML 2.0 component concepts and associated graphical notations.

Basically, the clock behavior is ensured by a `ClockController` component object. As shown on the architectural diagram in figure 5, the `ClockController` offers the same two interfaces provided by the clock (`IReadOnlyClockDisplay` and `IClockKeyboard`); all the messages that the latter receives, requiring services from one of its interfaces, are further delegated to the controller (this is indicated by the delegation connectors that link the two ports on the left with the corresponding provided interfaces). In order to grant this functionality, the `ClockController` component requires the services of three other interfaces, namely `IClockTicker`, `IClockMemory`, and `IClockDisplay`. These are implemented by the `ClockTicker`, `ClockMemory`, and `ClockDisplay` components, respectively. All three are observed

components; they have the ability to notify a potential listener (the controller in this case) when certain events (ticks) or state changes occur. Therefore, the interfaces they provide should offer `[add|remove]XListener` type services, requiring, at the same type, services provided by `XListener` type interfaces. As can be seen, the `PropertyChangeListener` interface requied by the clock component is actually required by its `ClockDisplay` (another delegation connector that links, this time, a required interface to a port).

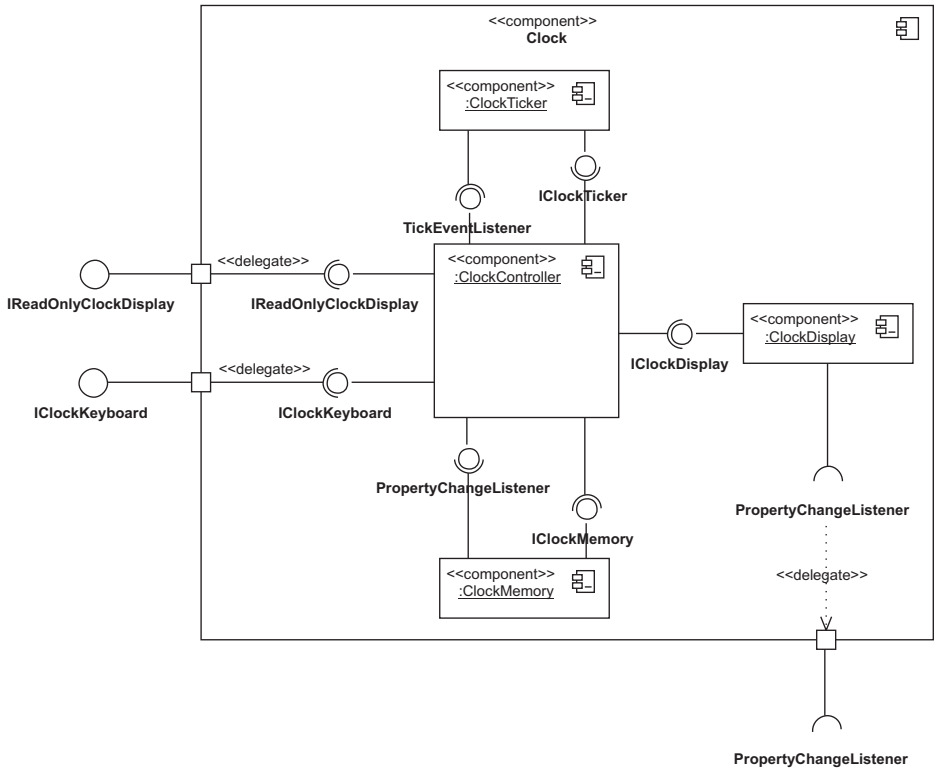


FIGURE 5. Internal architectural view of *Clock* component

In order to accomodate the space constraints of the paper, we will not enter the specification details of all interfaces. We will only insist on `IClockMemory`, by illustrating its information model, as well as some of the operations' specifications.

Figure 6 shows the Interface Specification Diagram for `IClockMemory`. It depicts the interface itself, together with the types that make up its information model. As shown by the diagram, the `IClockMemory` interface is represented as a class having the `<<interface type>>` stereotype, that lists all its services inside

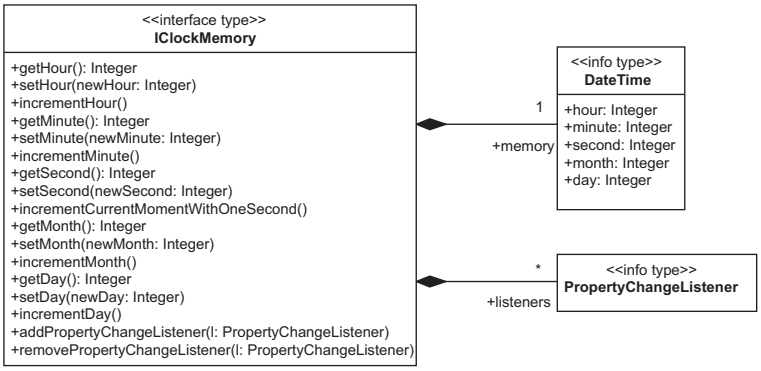


FIGURE 6. `IClockMemory` - Interface Specification Diagram

the operations compartment. Some of these services enable the handling (setting, getting or incrementing) of values stored in the clock memory (i.e. hour, minute, second, day, and month), while the others allow the management (adding or removing) of listeners registered with it. In order to ease the precise specification of these two types of services, we have associated to the `IClockMemory` interface an information model consisting of two classes, namely `DateTime` and `PropertyChangeListener`. Both have the `<<info type>>` stereotype (since they represent informational types) and are linked to the interface via composition relationships.

To conclude with the usage contract corresponding to `IClockMemory`, listing 1 shows some of the OCL pre/post-condition pairs that specify the operations' semantics. The entire OCL specification has been validated with OCLE 2.0 [1].

3.3. **Implementation issues.** Up to now we have concentrated on creating a component architecture and specifications. We have tried to make them as independent as possible from technological issues. The *provisioning* and *assembly* [2] phase is concerned with providing realizations for previously defined component specifications (either by finding existing implementations or developing them from scratch), as well as binding them as architected.

We have implemented our *Clock* component all from scratch using the JavaBeans standard. Each component specification has its counterpart in a JavaBean class (its realization). The bean implements all the provided interfaces appearing on the component specification diagram. As for the required interfaces, they were managed by storing their references inside the component object. This dependency allows calling their services whenever needed. The real component objects supporting these services were plugged inside the client component by means of specialized "plugging interfaces" that we have created. We regard these so called

Listing 1. IClockMemory - OCL constraints. ocl

```
context IClockMemory::getMonth():Integer
  post: result = self.memory.month

context IClockMemory::setMonth(newMonth:Integer)
  pre:  newMonth >= MIN_MONTH and newMonth <= MAX_MONTH
  post: self.memory.month = newMonth

context IClockMemory::incrementMonth()
  post: self.memory.month = self.memory.month@pre mod NO_OF_MONTH  + 1 and
        if self.memory.day@pre > noOfDays(self.memory.month)
        then self.memory.day = MIN_DAY
        else true
        endif

context IClockMemory::incrementCurrentMomentWithOneSecond()
  post: if self.memory.second@pre < MAX_SECOND
        then self.memory.second = self.memory.second@pre + 1
        else self.memory.second = MIN_SECOND and
          if self.memory.minute@pre < MAX_MINUTE
          then self.memory.minute = self.memory.minute@pre + 1
          else self.memory.minute = MIN_MINUTE and
            if self.memory.hour@pre < MAX_HOUR
            then self.memory.hour = self.memory.hour@pre + 1
            else self.memory.hour = MIN_HOUR and
              if self.memory.day@pre < noOfDays(self.memory.month@pre)
              then self.memory.day = self.memory.day@pre + 1
              else self.memory.day = MIN_DAY and
                if self.memory.month@pre < MAX_MONTH
                then self.memory.month = self.memory.month@pre + 1
                else self.memory.month = MIN_MONTH
                endif
              endif
            endif
          endif
        endif

context IClockMemory::addPropertyChangeListener(l:PropertyChangeListener)
  post: listeners = listeners@pre->including(l)

context IClockMemory::removePropertyChangeListener(l:PropertyChangeListener)
  post: listeners = listeners@pre->excluding(l)
```

"plugging interfaces" as playing the role of assembly connectors, linking a component's required interface to a corresponding provided one.

Several patterns have been applied during the design process. Among them, we may mention the *Observer* pattern, used for managing the raising of events by the `ClockTicker` as well as the state changes occured in the `ClockMemory` or `ClockDisplay`, the *State* pattern, employed for handling the `ClockController`'s dynamic behavior, or the *Factory Method* pattern, used in building the `Clock` itself

from components. Readers interested in all these details are strongly encouraged to contact the authors.

4. Conclusions and future work

The current paper belongs to a series of works trying to apply different formal specification techniques for the *LCD Wallet Travelling Clock* case study. This time, we have experimented a component-based approach. We have proposed a component architecture and we have tried to formally specify its building blocks using UML and OCL. Following this architecture and specifications, a JavaBeans implementation has also been developed. As a future research direction, we intend to study the opportunities given by formal specifications in the field of test automation. Precisely, we will try to derive JUnit test cases based on previously described OCL constraints.

References

[1] OCLE Homepage. `http://lci.cs.ubbcluj.ro/ocle/index.htm`.

[2] John Cheesman and John Daniels. *UML Components: A Simple Process for Specifying Component-Based Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.

[3] Vladiela Ciobotariu-Boer and Dragoş Petraşcu. X-Machines Modeling. A Case Study. In Militon Frenţiu, editor, *Proceedings of the Symposium "Colocviul Academic Clujean de Informatică"*, pages 75–80. Faculty of Mathematics and Computer Science, "Babeş-Bolyai" University of Cluj-Napoca, România, June 2005.

[4] Ivica Crnkovic and Magnus Larsson (editors). *Building Reliable Component-Based Software Systems*. Artech House, Inc., Norwood, MA, USA, 2002.

[5] Desmond F. D'Souza and Alan Cameron Wills. *Objects, Components, and Frameworks with UML: the Catalysis Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[6] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[1] Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Str. Mihail Kogălniceanu nr. 1, RO-400084 Cluj-Napoca

E-mail address: `petrascu@cs.ubbcluj.ro`

[2] Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Str. Mihail Kogălniceanu nr. 1, RO-400084 Cluj-Napoca

E-mail address: `vladi@cs.ubbcluj.ro`

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 341–345

A TSPACES BASED FRAMEWORK FOR PARALLEL-DISTRIBUTED APPLICATIONS

E. SCHEIBER

Abstract. A framework enabling the deployment and the execution of a parallel-distribute application is presented. The application is build on an imposed template based on the master-slave model. The application as well as the infrastructure of the framework use TSpaces servers. The coordination of the activities of the application is of data driven type.

1. Introduction

The purpose of this note is to present a framework enabling the deployment and the execution of a parallel-distribute application, whose development is based on the template presented in [4].

The development programming language is Java, used in many projects, frameworks and products for parallel-distribute computing and for high performance computing. The Java technology is improving continuously. Java was designed to meet the real world requirement of creating interactive, networked programs. Java supports multithreaded programming, too.

The model of the template is that of *master-worker* and the coordination is of *data driven* type based on a shared dataspace [3]. The idea of the developed approach is that of the Piranha project [3], and in order to carry out the proposed tasks we integrate as much as possible software components that are free of charge at least for a non commercial application.

The framework and the application template allows to solve problems which may be solved using MPI [5, 2, 1] in a network of workstations.

2. On the template of the parallel-distribute application

The instance of the *Dispatcher* class corresponds to the master, while an instance of the *Worker* class corresponds to a worker. The *Dispatcher* and the

2000 *Mathematics Subject Classification.* 68N99, 68N19.
Key words and phrases. parallel-distribute framework, Java, TSpaces.

©2007 Babeş-Bolyai University, Cluj-Napoca

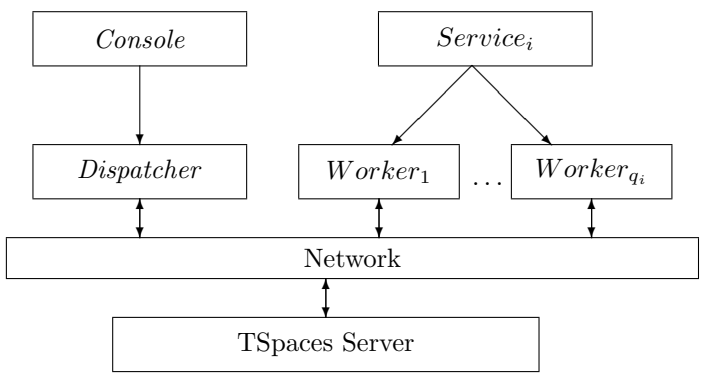
Worker classes are threads. The `run` methods of these Java threads contain the specific activities to solve the given problem.

For this version of the framwork we use *TSpaces 2.1.2,* from I.B.M., [6], as the shared data space for coordination and as a warehouse of the messages. *TSpaces* is a Java implementations of the Linda computational model.

Between the dispatcher thread and the worker threads there are asynchronous message exchanges. These messages are kept by a TSpaces server - the application associated tuple space. The data to be exchanged between the dispatcher and the workers are wrapped into objects. To distinguish between different kind of data, a tag field may be introduced. A message consumer (dispatcher or worker) waits until all the required messages are available. In this way the synchronization problems are solved as well as the coordination of the activities.

The dispatcher is launched to run by a *Console* program. On a workstation it is possible to start several worker threads. These threads are instantiated and started by a *Service* program. The planed number of the worker threads are equaly distributed to the involved workstations. This kind of organization of an application is similar to that used in the *JCluster* software [1].

The *Service* program is a servlet and we use the *apache-tomcat* as a servlet container Web server. The *Console* program which starts the dispatcher makes the requests to the *Service* servlets, too. The *Console* and *Service* classes are independent from an application.



The i index denotes a workstation on which q_i worker threads are run.

3. The framework

We suppose that several computers in a network will perform the required computation. On each workstation, the *apache-tomcat* Web server must be active.

To the network of workstations it is associated a second TSpaces server to keep the names of the involved computers-denoted as the tuple space of the network.

A scheme of the deployment of the framework with the required programs is given in Fig. 1.

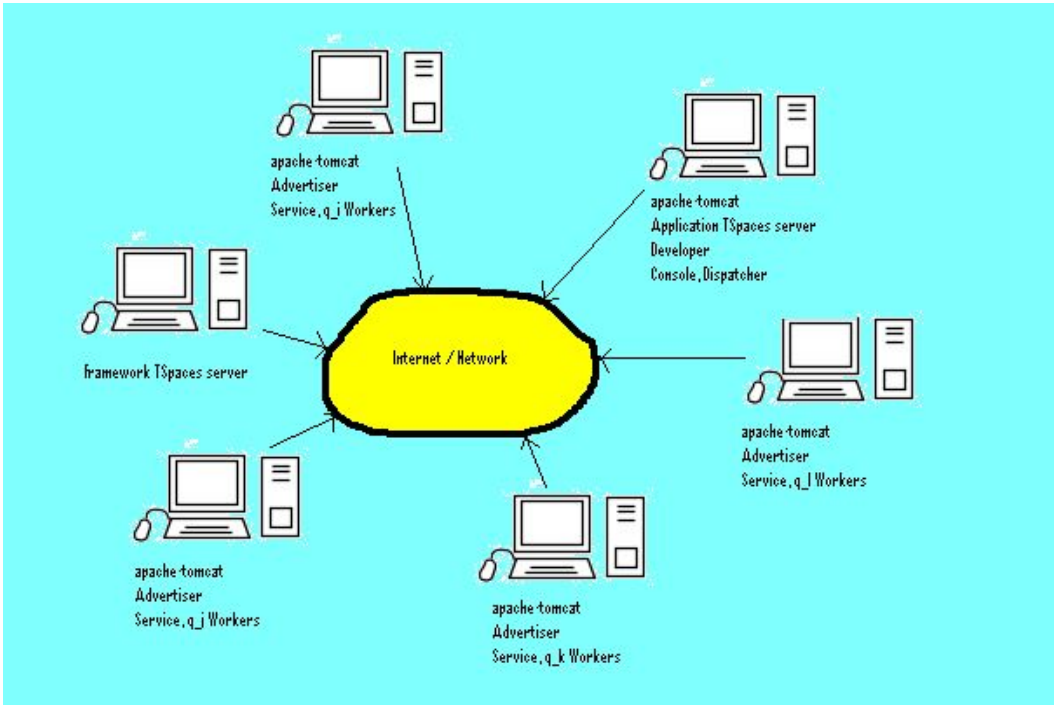


FIGURE 1. The running programs within the framework.

Two tools defines the framework:

- The *Advertiser* tool serves to state the network of workstations. Using this tool a computer is linked to the network of workstations to perform a parallel-distribute computing.

  The graphical interface of the *Advertiser* tool allows to
  - declare the computer as a member of the network of workstations - i.e. a tuple with the name of the computer is written into the tuple space of the network;
  - remove the computer from the network - i.e. remove the above define tuple from the tuple space of the network;
  - show the list of the computers in the network of workstations.

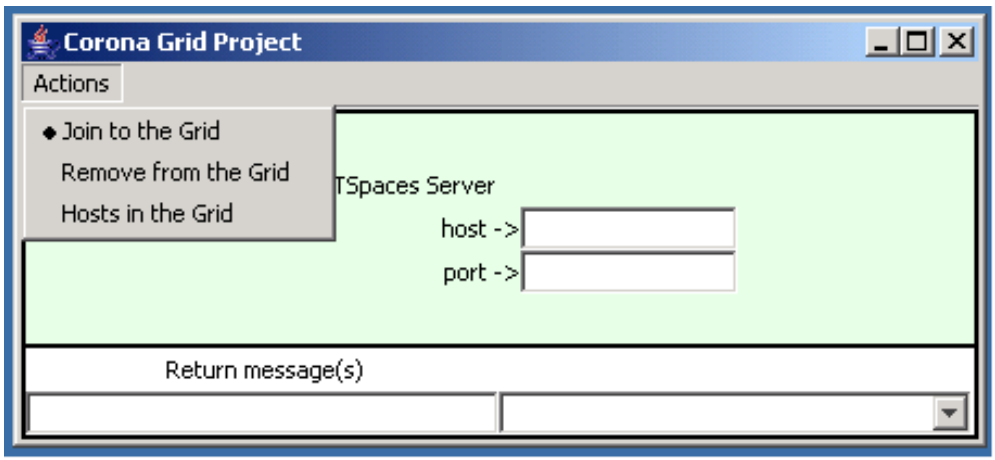


FIGURE 2. The window of the *Advertiser* tool.

- The *Developer* tool allows to
  - compile and arhive the service part of the application;
  - deploy the service part to the workstations of the network. The deployment is done with the *apache - tomcat - deployer;*
  - compile the console part of the application.
  - launch the console to run. In the *Console* class the request are programmed using the *commons - httpclient* software;
  - undeploy the service part of the application.

  These targets are executed throught *apache-ant.*

  A number of parameters are required: the name, the host and the port of the application tuple space, the number of the worker threads, the path to the *apache-ant* and the username and password for the `manager` application of *apache - tomcat.*

  The list of the computers name in the network is required, too. This list may be generated with the *Advertiser* tool, contained in the *Developer* tool, too.

The scenario to run a parallel-distribute application using the framework involves:

(1) To set up the network of workstations:
   (a) On launches the tuple space associated to the network;
   (b) On each workstation starts the *apache-tomcat* Web server and, with the Advertiser tool, declares the availability of that computer joining to the network of workstations.

(2) Using the Developer tool, on the host workstation install, deploy, launch the tuple space associated to the application and execute the parallel-distribute computation.

Conclusions. A framework to deploy and execute parallel-distribute applications is developed. The framework can be ported to any Java enabling platform. Before running an application, workstations can be added or removed. As a drawback of the framework, if a workstation dies then a running application never finishes. We intend to work on this problem. The security constraints are that of the *apache-tomcat* Web server.

The current version of the framework may be downloaded from the author's Web page *http://cs.unitbv.ro/site/pagpers/scheiber* The arhive contains several examples.

References

[1] Z. BAOYIN, "Jcluster A Java Parallel Environment," Distributed with the software, version 1.0.5, 2005.

[2] R. BISSELING, *Parallel Scientific Computation. A structured approach using BSP and MPI,* Oxford Univ. Press, 2004.

[3] G. A. PAPADOPOULOS, F. ARBAB, Coordination models and languages. CWI Report, SEN-R9834, 1998.

[4] E. SCHEIBER, Template for a parallel-distribute Application Based on a messaging Service. Proceedings of ICCCC 2006, Băile Felix, Oradea, Romania, 410-416.

[5] M. SNIR, D. OTTO, S. HUSS-LEDERMAN, D. WALKER, J. DONGARRA, *MPI: The Complete Reference.* MIT Press, Cambridge, MA, 1996.

[6] * * * , "TSpaces-User's Guide & Programmer's Guide," Distributed with the software, Version 2.1.2, 2000.

Transilvania University of Braşov, Romania
E-mail address: scheiber@unitbv.ro

KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 346–352

APPLYING TRANSITION DIAGRAM SYSTEMS IN DEVELOPMENT OF INFORMATION SYSTEMS DYNAMIC PROJECTS

NICOLAE MAGARIU

Abstract. The model of complex software development based on the applying of transition diagrams systems is considered in this paper. This model provides the minimization of elaboration time, improving the quality, and raising the technological level of the software.

1. The problem

The necessity of intensification of human society economical development on the one hand, and advanced performances of modern computers on the other hand, force automation of the most complex Information Handling Processes (IHP) in the field of economics. This automation can be realized by means of Complex Information System (CIS) [1]. CIS elaboration needs big intellectual efforts of a group of developers in the course of several years. In the context of CIS elaboration a special attention is given to CIS design phase, especially to CIS dynamic aspects design: components and subcomponents behavior specification, correlation of system's static and physic aspects with its dynamic aspects, etc. One of the well known means, which is applied to specify dynamic aspects of simple programs, is the transition diagram [2, 3]. An attempt to specify dynamic aspects of a CIS with the help of single transition diagram causes serious difficulties. Applying a set of transition diagrams to specify CIS dynamic aspects can be a solution of the posed problem. In this paper a model of applying Systems of Transition Diagrams (STD) in CIS dynamic projects elaboration is proposed.

2. Preliminary

STD had been used for the first time in 1963 by American scientist Melvin E. Conway when he had elaborated a separable compiler [4]. Conway had specified

2000 *Mathematics Subject Classification.* 68N30, 68N99.

Key words and phrases. Dynamic project, Systems of Transition Diagrams, Complex Information System, diagrammer.

©2007 Babeş-Bolyai University, Cluj-Napoca

the syntax of COBOL language - a Symbolic Programming Language (SPL) - utilizing STD, and had proposed an original algorithm of compilation which he had named Diagrammer.

A diagram defines the syntactical structure of an SPL construction and has a name, which identifies this construction. The diagram consists of nodes and edges, where nodes represent states and edges define transitions from one state to another. Edges can bee marked by terminal symbols of the SPL or by name of a diagram. The method of interlinking between diagrams characterizes a STD. The STD includes a main diagram - "program". Every edge can be associated with a program unit, which is executed when the corresponding edge is passed. The diagrammer analyzes a program (an input line) and identifies the program construction using one of the diagrams from STD. The diagrammer starts the input line analysis from the initial state (node) of the main diagram and recognizes program constructions in nondeterministic way - with returns in input line. Conway determines the possibility of STD and diagramer application for other languages compilation.

The American specialist, David Bruce Lomet, had studied the Conway diagrammer and had proved diagrammer's equivalence to a restricted Deterministic PushDown Acceptor (DPDA) called a nested DPDA [5]. He had established that the class of nested DPDA's is capable of accepting all deterministic context-free languages.

The author has been studied Conway's diagrammer independently of D. B. Lomet. A new version of diagrammer that functions in deterministic way was elaborated by him [6]. It was applied in the elaboration of APL interpreter.

The syntax of APL language constructions is very simple [7]. User variables are dynamic and they can obtain as a value a numerical array or an array of char type with arbitrary number of dimensions. APL operations can be nullary, unary or binary. Operands of the operations can be some expressions, evaluated to arrays, with arbitrary number of dimensions. Semantics of the APL operations is very consistent and can be described in C language using tens or hundreds of instructions. Almost all operations in APL have equal priority (with small exceptions). User Defined Functions (UDF) prototypes have the structure similar to the APL basic operators structure. The UDF can be made up from expression instructions or branching instructions. Jumps can be made inside UDF only. A UDF can't be defined inside another UDF. All UDFs necessary for solving a problem or a class of problems are stored in a Work Space. An expression can contain invocations of UDF. To solve a problem by means of APL system, one needs to compose the set of UDF and the APL expression. The execution of this expression leads to execution of UDFs, which are invoked directly or indirectly from this expression.

So, the SDT of the APL expression defines the behavior of a UDF system. The APL expression represents a control message to realize functionality, which

is necessary for a user. The diagrammer controls the order of execution of UDF (program units). In such way, the author came to the idea about using the SDT in designing of CIS [8]. In this case the SDT represents a dynamical project and an input line represents a sequence of control symbols for the diagrammer. Diagrammer takes control symbols from input line and executes corresponding program units of the project.

3. Diagrammer functions

The STD uses two disjoint vocabularies: Σ – terminal symbols vocabulary (control symbols), N – nonterminal symbols vocabulary (diagrams names).

The author proposed the set of formal notation for STD and defined some diagram structure limitations. These limitations provide the deterministic diagrammer work. The following notations were introduced:
Σ – terminal symbols of vocabulary;
N – nonterminal symbols of vocabulary;
IL – input line;
d_i – the name of i-th diagram of the STD ($i \geq 0$), $d_i \in N$;
Σ_{ij} – the set of terminal symbols, which mark edges going from j-th node of i-th diagram;
N_{ij} – the set of diagram names, which mark edges going from j-th node of i-th diagram;
$F(d_i)$ – the set of terminal symbols which are the first symbols of the sentences that can be read on passing d_i diagram ($i = 0..n-1$).

The construction method of $F(d_i)$ set:

$$F(d_i) = \Sigma_{i0} \cup (F(d_l)),\ for\ all\ d_l \in N_{i0}$$

The limitations for STD diagrams:

1) $N \neq \emptyset$;
2) Σ and N vocabularies don't contain inaccessible symbols;
3) STD diagrams don't contain unmarked edges;
4) there can't be two edges, outgoing from the same node and marked with the same symbol;
5) for every d_l diagram ($d_l \in N_{ij}$) the intersection Σ_{ij} and $F(d_l)$ is empty;
6) for 2 diagrams d_i and d_l, ($d_i, d_l \in N_{ij}$) the intersection $F(d_i)$ and $F(d_l)$ is empty set.

The sets $F(d_i)$ for concrete STD can be calculated before diagrammer starts execution.

If STD diagrams satisfy enumerated limitations, then in every state the diagrammer can choose only one possible transition. So, the diagrammer functions in deterministic way.

The determinate diagrammer executes the following actions:

A1. Work state of the diagrammer is determined: $i = 0; j = 0;$.

A2. The contents of the input line IL is determined; $k = 1$; //k – the index of a symbol from the IL.

A3. If $IL_k \in \Sigma_{ij}$, then: {
- Program unit associated with the edge, marked with IL_k is executed;
- Transition through the edge, marked with IL_k is realized and the value of j variable is modified;
- $k = k + 1$;
- Go to $A5$;}

A4. If $IL_k \in F(d_l)$ for d_l which belongs to N_{ij}, then: {
- The current state (the number of the current diagram and the current node number of this diagram) of the diagrammer is memorized in stack;
- It is fixed as current state the state corresponding to initial one of d_l diagram and respectively modifies values of i and j variables.}

If N_{ij} doesn't contain a dl diagram for which $IL_k \in F(d_l)$, then an error is detected in input line and the diagrammer stops.

A5. If the current node is the final node of the main diagram then the diagrammer stops.

If the current node is the final node of a diagram that differs from the main diagram then: {
- The information on the top of the stack is deleted;
- The state from the top of the stack is determined as the current state;
- The transition through the edges, marked with the name of the passed diagram, is made;}

Go to $A3$; //Or ***accepts an event*** after which goes to $A3$.

4. The applying of STD and diagrammer in the development of information system's dynamic project

When a STD represents a dynamic project of a software product (SP), we can naturally assume, that input line is always correct and so it is always accepted by diagrammer.

If the dynamic project is represented by a single transition diagram and this program realizes a single functionality, which doesn't need an intense dialog with the user, then we can consider the applying of the diagrammer less efficient.

If the IHP execution requires an intensive interaction with the user or other actors, then applying the diagrammer is efficient enough. In this case the diagrammer can work in conditions of accepting messages.

Depending on the way of interaction between user and CIS, elaborated on basis of diagramer, there can be used three modes of driving the diagrammer's work:

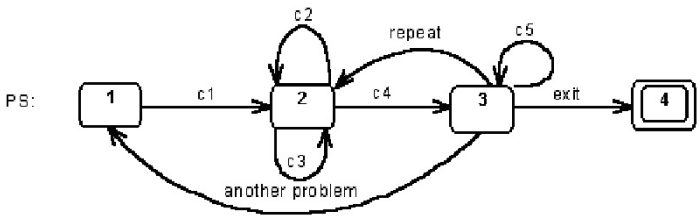


FIGURE 1. Main diagram of the SDT

(1) Any functionality of a CIS is defined preliminarily by a constant sequence of the control symbols. This constant is placed in input line of the diagrammer after selecting the corresponding option from a menu.
(2) Control symbols are generated as a result of the events produced by users or some other actors. The diagrammer waits the appearance of an event from the given set of events.
(3) Diagrammer's work is partially driven by constants - sequences of the control symbols, and partially by events.

Diagrammers functioning can be adapted in correspondence with the exploiting requirements of a CIS.

The CIS's dynamic project representation by means of STD correlates very well with top-down design of software systems. Top-down design of the complex program systems recommends initially elaborating a general structure of a system - its components (subsystems), and after that it recommends detailed modeling of these components. In this case a dynamic project can be represented by a STD, which contains a main diagram named "CIS". This diagram specifies the behavior of program system's components. The STD also must contain at least one diagram for every component.

Let us consider an example of a generalized project of complex software system, which consists of components $c1$, $c2$, $c3$, $c4$, $c5$. Let these components realize the following functionalities, necessary to user: $c1$ – initializations, adaptations, $c2$ – calculations according to methodology M1, $c3$ – calculations according to methodology M2, $c4$ – comparative analysis of the calculation results, $c5$ – report generation. Suppose that $c1$, $c2$ and $c3$ components need an intensive interaction with the user. Suppose that the behavior of the components is specified by the main diagram "CIS", shown in Figure 1. Symbols "another problem", "repeat" and "exit" are terminal symbols (events).

Suppose that the diagram, which corresponds to $c1$ component, has structure, shown in Figure 2. In this diagram $e0$, $e1$, $e2$, ..., eN are the terminal symbols (events). The names of the program units associated with edges of the diagram are not shown on diagrams. On appearing of the event the diagrammer starts

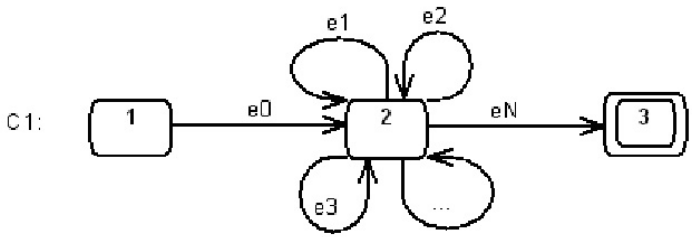


FIGURE 2. Transition diagram for component c1

the execution of the program unit, associated with the corresponding edge of this diagram. The transition is done after execution of program unit.

The transition diagrams for other components ($c2$, $c3$, $c4$, $c5$) of the CIS are constructed similarly.

It is easy to observe that the modifying of the software system needs the STD modifying only, but the diagrammer remains the same.

5. Conclusions

We can mention that the application of STD in software system development offers some important advantages:

1. Application of STD and determinate diagrammer represents a more systematized method of modeling complex software system's behavior and construction than known analogical methods. It permits the effective distribution of work among developers and permits fast assembling of software system.
2. The diagrammer represents an invariant part of a software system. Therefore it can be constructed once and reused in developing of other CIS. This fact provides the possibility to expand the quality of software systems and reduces time of their construction.
3. The structure of a software system can be modified and extended fast and easy.
4. The automation of the STD creation reduces time of CIS construction.
5. Deterministic STDs and diagrammer represent an efficient mechanism for elaborating event-driven software systems.

To construct quickly the compound software system, the instrumental framework may be constructed using the stated results in this paper.

CIS design by means of STD is based on functional refinement of the project. To improve effectiveness of this type of modeling it needs to be completed with an adequate systematic method of data design.

6. References

1. A. Stepan , Gh. Petrov , V. Iordan , *Fundamentele proiectării şi realizării sistemelor informatice*, Ed. MIRTON, Timişoara, 1995, 282 p.

2. Ciubotaru C.S., Magariu N.A., *Organizaţyia lexicescogo analiza* (Metodicescaia razrabotca). /RIO CGU, Chişinău, 1984, 28 s. (In Russian)

3. *http://yourdon.com/strucanalysis/wiki/index.php?title=Chapter_13 #STATE- TRANSITION_DIAGRAM_NOTATION*

4. Melvin E. Conway, *Design of a separable transition-diagram compiler.* //Communications of the ACM, Volume 6, Issue 7 (July 1963), pp. 396–408.

5. David Bruce Lomet, *A Formalization of Transition Diagram Systems.* //Journal of the ACM (JACM) V. 20 , Issue 2 (April 1973), pp. 235–257.

6. Magariu N.A. *Ispolizovanie diagram perehoda pri realizaţii dialogovoi sistemi programmirovania.* //Matematiceschie issledovania AN MSSR, vîpusc 107. Teoria i practica programmirovania. Chişinău, Ştiinţa, 1989, ss. 100–110.(In Russian)

7. Magariu N.A., *Iazîc prorammirovania APL.* /Radio i sviazi, Moscva, 1983, 87 s. (In Russian)

8. Magariu N., *Utilizarea diagramelor de tranziţie la construirea sistemelor de prelucrare a informaţiei*, Materialele conferinţei republicane "Informatică şi tehnică de calcul", Chişinău, 1993, pp.61–62.

State University of Moldova, Chişinău
E-mail address: magariu@usm.md