

## **A low te(a)ch approach to digital humanities**

Liviu Pop

DigiHUBB, Babeş-Bolyai University, Cluj-Napoca, Romania,  
E-mail: liviu.o.pop@gmail.com

**Abstract:** Technological fatigue is spreading due to the increased arrays of technological branches and the rapid pace of development. Keeping up with the latest and newest might prove not only difficult and tiring, but also not necessarily recommended to beginners in digital humanities. A more “back to the basis” approach might be more useful for students that don’t have technical backgrounds.

**Keywords:** low tech; minimal computing; technological fatigue.

There are few domains that faced such a fulminant evolution as the computer sciences, the only comparable other domain being genetics and bio-engineering - it took 35 years from isolating the DNA as the genetic material (Avery et al.) to cloning the first sheep, Dolly (Kellan), in 1997. If the prehistory of modern computing can be traced back to the automatic looms of the 18<sup>th</sup> century from France (Basil Buchon and later Jaques Laquard), the true birth of what we call today computers was the middle of the 20<sup>th</sup> century. It is mind-blowing what happened in the last half century in the computer science and computer industry. The computers evolved in only 40 years from being highly specialized machines that were calculating missile projections (Colossus), or coding messages (Enigma) during the Second World War, to personal computers affordable in every house (Apple II). Not even the most daring of the first people in the computer industry, except perhaps Gordon Moore (Moore), could have imagined the exponential growth of the computing machines.

“The complexity for minimum component costs has increased at a rate of roughly a factor of two per year... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit

more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer.”

On the other side of the coin, there were assumptions that proved to be exceptionally wrong, like the anecdotic quote attributed to Bill Gates (Andrews, O’Toole) that we won’t need more than 640 KB on a computer. He was referring, of course, to computer running MS-DOS, but none the less, this proves that even the most acclaimed visionaries are suffering sometimes from a form of myopia.

This myopia is what sometimes stops us from grasping the immense and accelerated advanced of the digital technologies that took place in the last decades. In the same time, it might sometimes act as a barrier for having a deep grasp of how they work and where to start in understanding them. One of the most recent example is the state of JavaScript in 2016 and 2017. There were several high-profile articles complaining about the extremely baroque way the JavaScript initiatives are overlapping and depend upon each other. The JavaScript fatigue was coined in 2015 by Eric Clemmons, in an article with the same name (Clemmons), followed up by a more humorous and viral take on the same issue that was published in 2016 by Jose Aguinaga (Aguinaga). Some even tried to create a history and to go deep for the roots of the contemporary state of affairs, getting back to the browsers wars from the 1990s (French-Owen). There were other reactions, of course, defending the diversity and the large array of options that JavaScripts provides, such as the opinion of the JavaScript cheerleader Eric Elliot (Elliot) or the analogies made by Quincy Larson (Larson).

This state of increased complexity and intricate layers build on top and across each other can be found in other parts of the digital landscape. As tutors for future digital humanists, how can we help them better trough the challenge of having a deep and useful understanding of how the contemporary digital technologies work? How can we help them navigate this landscape? My short answer is not to look over their shoulders as they explore each map, but to give them a compass and to teach them how to read a map. Going back to the basics.

One of the options would be to wait for the computers to adapt to the humans, not the other way around. We already see this happening with the increased usage of mobile devices. Mobile devices, the phones and the tablets that became mainstream over the last decade, are powerful little computer always available for us to interact with in our pockets. This lowered the barrier for many digital non-natives in interacting with the computing devices, the multitouch screen being a real facilitator of quicker and easier interactions. But behind the scenes a team made up of programmers, designers, UI an UX designers programmed what will happen on the screen when a certain

action of the user takes place. Programming is still an arcane endeavour and we are still up to achieve the state where the “computer is used as a human brain assistant instead of using the human brain as a computer assistant.” (Rus)

But what can we do right now in the field of digital humanities in order to use the digital technologies for our benefit instead of being paralyzed by the myriad of options available? I would like to give a few examples of possible projects that might show an alternative approach. The first example comes from the Minimal Computing Working Group that “kickstarted itself into life with a workshop on July 8 at the DH2014 conference in Lausanne, Switzerland.” (Minimal Computing)

The group put together a few “thought pieces” around this subject and thus loosely define what minimal computing might be. It might be an “unfinished stairway” (Oroza) concept:

“Then, what is a stairway? How does one describe it? Could he build a structure in front of his doorway that looks nothing like a stairway but serves the same function? Maybe just objects stacked in such a way that one can climb and descend them? Or an object by Ettore Sottsass, a stack that includes all of Feijóo’s books, a Franz West sculpture, anything?

He decided on a conceptual shortcut: he built the stairway and waited to be fined. In this way, he gained time. The Law demanded that he cease building the stairway until the paperwork needed to divide the property was finalized.

Years went by. He used the unfinished stairway.

What’s a finished stairway?”

The concept of unfinished stairway is used by Gil in order to induce the need to short-circuit the whole process of making scientific research available to the public, getting back to using basic tools in order to start “producing our own scholarship ourselves” (Gil).

But the quintessential example of minimal computing, in my opinion, is The Minimal Computing Lab at the Centre for Textual Studies at De Montfort University from Leicester, England. In an attempt to help their students deeply understand how the computers work and considering that “to introduce students to how computers work and how to program them, modern computers – even stripped-down ones such as the Raspberry Pi – are much too complicated” (Egan) they went further down the line of computing history and they setup a lab with Altair 8800 and Amstrad PCW. Now here comes the real minimal computing paradigm: “These machines have no hard drives, no built-in operating systems, and no Read-Only-Memory (ROM). When such a machine is switched on it has empty memory and a processor doing.” (Egan)

Even more, learning how to program a very simple computer, by today’s standards using BASIC is a less terrifying approach for students that have a background in arts and humanities. The students already have the necessary skills for understanding the

BASIC syntax which is not very far away from finding a paragraph in a scholarly edition. Using numbers as a way of navigating a text structure is something familiar and can be further developed into thinking more abstract code structures. Even more, by facing the limitations of the very simple computers, the students are forced to think in modules and to use more complex algorithms from start. It might come as a surprise, but “the principles of Minimal Computing are an excellent introduction, for those who wish to pursue the topic, to the principles of Supercomputing.” (Egan)

Another member of the group, Sayers, compiled a manifesto in disguise, a list of minimal definitions. The list of minimums and maximums there are an ideological template for achieving the most with the simpler and the most basic of the available technologies. The list of commands tries to answer the fundamental questions of what do/not need or want. As any manifesto, it is an inspiring call to action along the sets of principles that arise from answering those questions. While some of the principles are related to strictly technical choices (“Minimal Maintenance - Reduce dependencies and the use of features to decrease the labour of updating, moderating, and stewarding a project over time”, “Minimal Vulnerabilities - Reduce attack vectors (e.g., cross-site scripting, SQL injections, and directory traversals) of projects to decrease likelihood of hacks and harassment”), other have an assumed social and environmental impact (“Maximum Ephemerality - Reduce an impulse to inscribe, measure, or visualize with technologies in order to increase the likelihood of experimentation and collective participation”, “Maximum Justice -

Reduce the use of technological, cultural, social, and economic barriers to increase entry, access, participation, and self-representation in computing and to also build systems/projects premised on social justice and difference, not white supremacy and settler colonialism.”) (Sayers)

This ethos of minimal computing is something that should be further explored in the way digital humanities are thought to students. Inspired by the minimal computing manifesto we can create projects in which the computing part is hidden, less visible and the focus is more on the creative part and the creative input. A project in which this approach might prove successful is solution for returning immaterial culture back to the villages where it was captured.

The ethnological archive from the Faculty of Letters, Babes-Bolyai University of Cluj-Napoca was conceived as a means to complete the Literary Folklore course (Sava). The documents are kept on paper, being written down, dictated or following direct field observations, or they represent magnetic tape transcripts. We strongly believe that is a moral duty to bring the documents from the Archive of the Folklore Circle back into the villages where they were gathered and we expect an increase in the preservation of the traditional immaterial culture in those villages by learning and re-learning the old customs, stories and songs. The documents have been digitized and now we are

currently having something akin a seed-bank, seeds of a culture that is fading. By using the digital technologies, we would like to replant those seeds back to where they belong (Pop).

One of the way to achieve this is to install small memory boxes with the digital versions of the documents. Those boxes should feel familiar to the villagers and it shouldn't look too "technological", in order to decrease the barrier for interactions. Even though the box will be an installation containing a Raspberry PI loaded with sound files that could be accessed via RFID tags. The most important aspects shouldn't be the computer or the interface in itself, but the content loaded on the computer. In order to achieve this, we will create a custom-made interfaces: a box that will start playing songs when a wooden coin containing RFID tags will be placed on top of it. The originality of the concept doesn't consist in creating a robot that plays music using RFID tags, which was already done and used as an inspiration source<sup>1</sup>, but on the emphasis put on hiding the computer and the auxiliary interfaces in order to obtain an installation that is less inhibiting. The minimal computing consists in using wide spread technology like RFID for obtaining an installation that does a simple action: starts playing music.

Minimal computing ethos might prove an alternative to the technological fatigue created by a continuous need to be up-to-date with the latest technology. Having this approach for the students that are starting to learn digital humanities might prove fertile.

### Works cited

- "About Minimal Computing." Minimal Computing. N.p., n.d. Web. 21 July 2017. <http://go-dh.github.io/mincomp/>.
- Aguinaga, Jose. "How it feels to learn JavaScript in 2016 – Hacker Noon." Hacker Noon. Hacker Noon, 03 Oct. 2016. Web. 21 July 2017. <https://medium.com/@ericclemmons/javascript-fatigue-48d4011b6fc4>.
- Andrews, Nancy. Windows: the official guide to Microsoft's operating environment. Harper & Row Publishers, Inc., 1986. Page 268, Microsoft Press, Redmond, Washington.
- Avery, Oswald T., Colin M. MacLeod, and Maclyn McCarty. "Studies on the chemical nature of the substance inducing transformation of Pneumococcal types." Journal of experimental medicine 79.2 (1944): 137-158. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2135445>
- Clemmons, Eric. "Javascript Fatigue – Eric Clemmons – Medium." Medium. Medium, 26 Dec. 2015. Web. 21 July 2017. <https://medium.com/@ericclemmons/javascript-fatigue-48d4011b6fc4>.

---

<sup>1</sup> User *roaldh*, Raspberry Pi Based RFID Music Robot, available at <http://www.instructables.com/id/Raspberry-Pi-based-RFID-Music-Robot/>.

- Egan, Gabriel. "Old Machines Running Old Languages." Minimal Computing. N.p., 03 Aug. 2015. Web. 21 July 2017. <http://go-dh.github.io/mincomp/thoughts/2015/08/03/old-machines/>.
- Elliott, Eric. "Why I'm Thankful for JS Fatigue. I know you're sick of those words, but this is different." Medium. JavaScript Scene, 24 Nov. 2016. Web. 21 July 2017. <https://medium.com/javascript-scene/why-im-thankful-for-js-fatigue-i-know-you-re-sick-of-those-words-but-this-is-different-296fae0c888f>.
- French-Owen, Calvin. "The Deep Roots of Javascript Fatigue." Segment. N.p., 15 Mar. 2016. Web. 21 July 2017. <https://segment.com/blog/the-deep-roots-of-js-fatigue/>.
- Gil, Alex. "The User, the Learner and the Machines We Make." Minimal Computing. N.p., 21 May 2015. Web. 21 July 2017. <http://go-dh.github.io/mincomp/thoughts/2015/05/21/user-vs-learner/>.
- Kellan, Ann. "A sheep cloning how-to, more or less." CNN. Cable News Network, 24 Feb. 1997. Web. 21 July 2017. <http://edition.cnn.com/TECH/9702/24/cloning.explainer/index.html>.
- Larson, Quincy. "JavaScript Fatigue Fatigue – freeCodeCamp." FreeCodeCamp. FreeCodeCamp, 06 Oct. 2016. Web. 21 July 2017. <https://medium.freecodecamp.org/javascript-fatigue-fatigue-66ffb619f6ce>.
- O'Toole, Garson. "Computer Memory: 640K Ought to be Enough for Anyone." Quote Investigator. N.p., 2012. Web. 21 July 2017. <http://quoteinvestigator.com/2011/09/08/640k-enough/>.
- Oroza, Ernesto, and Penélope De Bozzi. *Objets réinventés: La création populaire à Cuba*. Paris: Alternative, 2002. Print. p. 107
- Pop, Liviu. "Open Source Folklore Archives-Disseminating Small Memory Boxes", *Review of the National Center for Digitization*, 2015. 47-52. <http://elib.mi.sanu.ac.rs/files/journals/ncd/26/ncdn26p47-52.pdf>.
- Roaldh, and Instructables. "Raspberry Pi Based RFID Music Robot." Instructables.com. Instructables, 12 May 2016. Web. 21 July 2017. <http://www.instructables.com/id/Raspberry-Pi-based-RFID-Music-Robot/>.
- Rus, Teodor. "Milestones for Computing Future - (An answer to Vinton Cerf's call for Computing Milestones, Communications of the ACM 07/2015, 7)." N.p., n.d. Web. 21 July 2017. <http://homepage.cs.uiowa.edu/~rus/milestones.pdf>.
- Sava, Eleonora, and Liviu Pop. "The ethnological archive: Memory and technology." *Philobiblon: Transylvanian Journal of Multidisciplinary Research In Humanities*, 15393 407 (2010). [https://www.academia.edu/813668/The\\_Ethnological\\_Archive\\_Memory\\_and\\_Technology](https://www.academia.edu/813668/The_Ethnological_Archive_Memory_and_Technology).
- Sayers, Jentery. "Minimal Definitions (tl;dr version)." Minimal Computing. N.p., 03 Oct. 2016. Web. 21 July 2017. <http://go-dh.github.io/mincomp/thoughts/2016/10/03/tldr/>