A Bare Minimal Computer for Everyone

Lucas Chaloyard, Florent de Dinechin, Marie-Pierre Escudié, Lionel Morel
Univ Lyon, INSA Lyon, Inria, CITI, EA3720, 69621 Villeurbanne, France

I. The State of Computer Science

Since its beginning, computer science has relied on a number of self-fulfilling prophecies like Moore's law, which stipulates eg that "the number of transistors in a silicon chip doubles every two years". It brought us many great applications which have been beneficial to society like basic information processing, shared knowledge, inter-personal communication, and surely others. From our point of view, Moore's law is the result of two injunctions that feed of one another. Software people are trying to do the most out of the hardware resources given to them, eventually reaching their current limits and continually asking for more. Hardware designers on the other end will try to produce more powerful hardware to satisfy software demands and gain commercial leadership. As a correlation, this trend leads to a tendency to digitalize all human activities, and impose digital systems as a radical monopoly [8], [9].

Today, digital society's development raises many questions due to its hegemony, about the resources, both energy and materials, needed for this development [1], but it also has societal implications that can't be ignored [3], [4], [5]. Our current conception processes which are apparently more efficient than those of the past, aren't compensating the exponential growth of digital usage. Even worse, digital technologies act as accelerators for many other technologies and large scale rebound effects are probably acting as negative overcompensating forces. To put it bluntly, an exponential growth of digital solutions implies an exponential need of raw materials and energy. In a finite world, it is of course impossible for this to last forever. In the large, we believe digital society as it is idealized today isn't sustainable.

These concerns lead us to the working hypothesis that more digital technologies will not be the solution to society's issues. Thus, we need to rethink about how computers are used: What are the truly useful or desirable use cases of digital solutions? This implies that we need to have a specific definition of utility or usefulness, which takes into account the consequences of those use cases on society as a whole.

We think that it is possible to keep some of the benefits of digital systems while having an impact on the planet that can be sustained for decades or centuries from now. As computer scientists, we want to aggregate knowledge that will allow for the construction of such systems. To do so, we want a research based on a new paradigm, one that "aims of meeting the needs of all people within the means of the living planet" as central to the doughnut vision in economics, see figure 1 [14].

With this vision in mind, we should aim at two objectives.

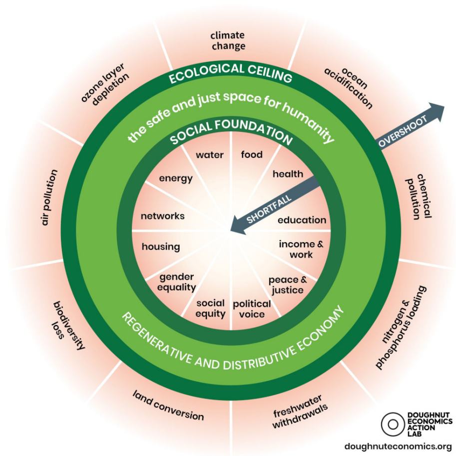

Figure 1. Doughnuts economic representation, credits: https://doughnuteconomics.org/about-doughnut-economics

One is to maintain social foundations, to make sure that every human being has access to life essential's and is able to live freely. The other objective is to stay below a certain ecological ceiling which is represented by our planet boundaries.

II. Computer Science in the Doughnut

In the large, our work fits into the trend of "computing within limits"[1] trying to envision and prepare "computers" to fit in the doughnut. One approach would be to reduce the impact of computer systems existing today and make them fit into the planet boundaries. From our point of view, this approach presents the issue of the precise definition of these limits. In order to be below the planet boundaries, we need relatively precise knowledge of their quantification, which is probably very hard to gain. Trying to fit computers into a strict upper limit leaves us with a really small error margin about the value of this limit.

Rather than trying to reduce the "ideal" computer so that it fits under the ceiling, we choose to try and identify properties, components and functions of a (kind of minimal) computer system for which we are confident its production, functioning and recycling would allow us to stay well below the planetary limits. As researchers in software, we will very roughly approximate the hardware properties of this "bare minimal computer" and try to design self-sufficient software for that hardware. Its the design of some of the software bricks necessary for this self-sufficient minimal computer to run that is at the heart of our current research.

[1] https://computingwithinlimits.org/

Apart from the ecological ceiling, the doughnut vision encourages us to also maintain or reach the social foundations. So we not only want a computer that can be manufactured and made to run within the geophysical limits of the planet, but we also want to build computers that are inclined to make people's basic needs more reachable. Social equity for example, can be worked on, by making the computer, as a tool, usable by and useful for everyone. One way to reach this goal, is to try to build a computer that would make no distinction between technical users and regular users, in order for everyone to have almost the same power on the tool.

Of course, the question of the place of digital systems within the doughnut really concerns all branches of computer science, of computer technologies, software and hardware, and all pieces in the traditional software stack. From the usage point of view, we've decided to focus our work on the personal computer. This dictates the kind of use cases we will consider as well as the kind of hardware we will build software for. But personal computers most easily fit the conviviality requirement we would like to pursue.

Deciding what are the truly useful, desirable or beneficial use cases can't be made by computer scientists alone. This kind of choices has cultural and political implications that should not be left to engineers or salesmen alone, as is the case today. For now, we have concentrated our thoughts on a few very generic use cases that seem truly beneficial to society, using our personal thoughts and ideas, but also research and discussion about the needs of humans [12], [6]. From that, when choosing the computer and design its software, we want to iteratively add the bare minimum needed for the development of those use cases. We call this hypothetical computer the *bare minimal computer for everyone*.

As low-level software researchers, we concentrate on the Operating System (OS for short) layer. Below this layer resides the computer's hardware that might vary from one platform to the other. The change of hardware should not annihilate the possibility to use the software. So we want to design software that minimizes dependency to the hardware, trying to enhance resiliency of the software towards the hardware. Above the OS, users need to comply to its interfaces. These will have influence on how the OS is used, and on what the objects built with it will look like. As a general goal, we need this interface to present some kind of conviviality in the sens of Ivan Illich in [8]: it should empower its user rather than imprison him.

In order to reach those goals, we'll follow a constructive approach (see figure 2). From a few use cases that we consider beneficial, we'll try to build the bare minimal computer capable of making these use cases possible. Despite the fact that we're only thinking about a few use cases, a computer remains a programmable tool, and we want it to stay that way, to let its users use this tool however they want. If a new use case seems appropriate, we'll think about the needed software extension, and if by adding it, we can maintain the good properties of the system, we might add it. Following this approach, we're hoping to build a computer useful and desirable for a frugal society [11], [7].

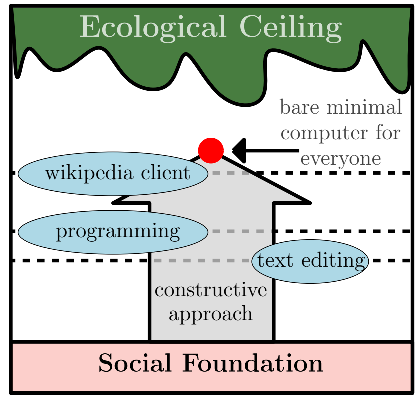

Figure 2. Constructive approach: building from the bottom until we reach our use cases

To begin with, we'll talk about some properties that we have identified as useful to reach resiliency and conviviality. Those properties have been inspired by the study of two minimal OSs CollapseOS[2] and DuskOS[3] built with self-sufficiency in mind, and that are capable of self-replication. Several of those properties are well-known in the fields of permaculture (and even permacomputing) and low-tech [2]. We quickly explore them now.

III. Properties of a Convivial and Resilient OS

A convivial tool is one that is controlled by its users and that allows them to shape the world as they wish. It must not decrease the autonomy of its user, and it has to augment her range of possibilities. As we see it, conviviality will lead to fairer tools, by making them usable by anyone.

Resiliency qualifies the capacity of an individual, a group or an object to resist to an external disturbance or change. Computer systems aren't usually resilient. One easy example would be the dependency of the whole digital technology to mineral resources. We might not be able to build computers as powerful as today for a long time, and the software designed for those powerful computers might become useless because of that.

We'll talk about portability, accessibility and minimality, which we consider as great properties to reach conviviality and resiliency.

Making an operating system more portable, ie less dependent to hardware, and by that we mean to minimize the effort needed to port a system to a new hardware platform, seems beneficial.

Portability should be coupled with a strong accessibility of the software. By accessibility we mean the ability of a system to be understood and appropriated easily by "any" kind of users (of course with an effort adapted to the initial knowledge of the user). It would allow the system to be ported

[2]http://collapseos.org/
[3]https://sr.ht/~vdupras/duskos/

by anyone, and to a lot of hardware platforms, making the system available for as many people as possible. Also, it makes possible for any user to control the system as she wishes.

So portability and accessibility seem a good way to reach resiliency and conviviality and also to make the tool more socially and environmentally fair.

In order to make the two previous properties more effective, we also think minimality could lead to two interesting properties. The first is that the system needs for memory are as low as possible, so the system becomes usable on a larger number of hardware platform, making portability greater. The second is that minimality will encourage simplicity, which makes a system easier to understand, and less dependable of highly complex tool chains, making it more accessible to everyone.

IV. Computer Sciences need Social and Human Sciences

Various fields from Social and Human Sciences should be put at work conjointly with Computer Science in order to build our minimal computer systems. Many tracks are already being explored. Here are a few of them that have caught our attention recently and that we believe should feed our work.

- The Limites Numériques[4] research project addresses the question of digital systems within planetary limits from a design research point of view. In particular, the work of Léa Mosesso et al [13] proposes an interview-based qualitative study of the influence of the gradual obsolescence of software on mobile devices. In this line of work, we would like to do a similar study about how people would feel about going back to simpler technologies.
- We have currently started a collaboration with the group at the Université de Montréal in charge of the Chemin de transition[5]. This prospective study, which is comparable to the work of ADEME in France on Les futurs en transition[6] proposes several prospective scenarios at the scale of a country targeting carbon neutrality within the next century. We are currently working on refining existing scenarios to define the place of computers in them: what computers would really be necessary? With what software?
- The ATD Quart Monde NGO's goal is to put an end to extreme poverty, build a fairer society which respect the fundamentals rights and give an equal dignity to everyone. Amongst many other subjects, ATD-QM works on how the poorest populations deal with the digital transformation of society. We plan to collaborate with AT Quart Monde's working group on "Digital Systems and Extreme Poverty"[7] in order to de-focus our thinking process from our own social categories (wealthy, western, male-centered) and understand the impact of technological choices from other stand-points.

[4]https://limitesnumeriques.fr/a-propos
[5]https://cheminsdetransition.org/
[6]https://transitions2050.ademe.fr/
[7]https://www.atd-quartmonde.fr/nos-actions/reseaux-wresinski/reseau-numerique-et-grande-pauvrete/

These projects will bring us many insights that we can't have as computer scientists, enriching our work to think and build systems that are truly beneficial.

There is still a lot of questions about computer science that need answers, and a lot of them need the participation of others disciplines to be answered. Those answers might be the missing pieces of the bare minimal computer for everyone that we call for.

V. Conclusion - The missing pieces

A trans-disciplinary research is needed to fully understand how computer science can be used to be beneficial to society. Those disciplines might help us answer new questions that emerge during our research.

- Why did computer science appear? What is the initial need it responds to?
- Are computers truly desirable?
- What is the current impact of computer science on society?
- How should research be done? And what should its end goal be?
- Will we be able as a society to keep computers as individual apparatus? Or should we keep only a few of them for very specific applications (scientific research, medical use, and so on...)?
- Should some computer science's applications be controlled or forbidden?

In a sense, all these questions encourage computer scientists to adopt an epistemological questionning of their discipline. Answering them will contribute to defining a concrete landing point for computers in a model of sustainable society, somewhere down to earth [10].

References

[1] ADEME. Les métaux: des ressources qui pourraient manquer ? https://multimedia.ademe.fr/infographies/infographie-terres-rares-ademe/.
[2] P. Bihouix. *L'Âge des low tech - Vers une civilisation techniquement soutenable*. Anthropocène Seuil, 2014.
[3] Y. Citton. *L'économie de l'attention*. La Découverte, May 2014.
[4] M. Dacos. Dominique Cardon, La démocratie Internet. Promesses et limites. *Lectures*, Oct. 2010.
[5] P. Flichy. Comment le numérique change le travail. *Idées économiques et sociales*, N° 194(4):16–23, Nov. 2018.
[6] W. Gianinazzi. Partir des besoins à partir d'André Gorz. *EcoRev'*, N°49(2):147, 2020.
[7] Heureuse Pour Une Frugalité. *Commune frugale: La révolution du ménagement*. Actes Sud, Mar. 2022.
[8] I. Illich. *Tools for Conviviality*. Marion Boyars Publishers, 1973.
[9] S. Kell. De-escalating software: conterproductivity versus conviviality, 2019.
[10] B. Latour. *Down To Earth. Politics in the New Climate Regime.* Polity, 2017.
[11] F. Lesueur, L. Morel, G. Salagnac, and N. Stouls. Proposition d'équipe «phénix» - informatique frugale et sécurité émancipatrice pour un numérique de la transition. https://phenix.citi-lab.fr/Docs/Propal.pdf, 2021.
[12] A. H. Maslow. A theory of human motivation. *Psychological Review*, 50(4):370–396, July 1943.
[13] L. Mosesso, N. Maudet, E. Nano, T. Thibault, and A. Tabard. Obsolescence Paths: living with aging devices. In *ICT4S 2023 - International Conference on Information and Communications Technology for Sustainability*, Rennes, France, June 2023.

[14] K. Raworth. *Doughnut Economics: Seven Ways to Think Like a 21st-Century Economist*. Random House, 2017.