Turing Complete User (2012)

"Any error may vitiate the entire output of the device.
For the recognition and correction of such malfunctions
intelligent human intervention will in general be necessary."
John von Neumann, 1945[1]

"If you can't blog, tweet! If you can't tweet, like!"
Kim Dotcom, 2012[2]

1 John von Neumann, First draft of a Report on the EDVAC. Moore School of Engineering, University of Pennsylvania (1945).

2 Kim Dotcom, "Mr President" (2012); https://www.youtube.com/watch?v=MokNvbiRqCM&t=218s, access: January 20, 2021.

Olia Lialina, Turing Complete User. Resisting Alienation in Human Computer Interaction, Heidelberg arthistoricum.net, 2021. (Interface Critique book) DOI: https://doi.org/10.11588/arthistoricum.972

Invisible and Very Busy

Computers are becoming invisible. They shrink and hide. They lurk under the skin and dissolve in the 'cloud'. We observe the process like an eclipse of the sun, partly scared, partly overwhelmed. We divide into camps and fight about the advantages and dangers of the Ubiquitous. But whatever side we take – we do acknowledge the significance of the moment.

With the disappearance of the computer, something else is silently becoming invisible as well — the User. Users are disappearing as both a phenomenon and a term, and this development is either unnoticed or accepted as progress – an evolutionary step.

The notion of the Invisible User is pushed by influential user interface designers, specifically by Don Norman, a guru of user-friendly design and long-time advocate of invisible computing. He can actually be called the father of 'invisible computing'. Those who study interaction design read his "Why interfaces don't work", published in 1990, in which he asked and answered his own question: "The real problem with the interface is that it is an interface". What's to be done? "We need to aid the task, not the inter face to the task. The computer of the future should be invisible!"[3]

It took almost two decades, but the future arrived around five years ago, when clicking mouse buttons ceased to be our main input method and touch and multi-touch technologies hinted at our new emancipation from hardware. The cosiness of iProducts, as well as breakthroughs in augmented reality (it got mobile), the rise of wearables, the maturing of all sorts of tracking (motion, face) and the advancement of projection technologies erased the visible border between input and output devices. These developments began to turn our interactions with computers into pre-computer actions or, as interface designers prefer to say, "natural" gestures and movements.

3 Don Norman, Why interfaces don't work, in: *The Art of Human–Computer Interface Design*, ed. Brenda Laurel (Reading, MA et al. 1990), p. 218.

Of course, computers are still distinguishable and locatable, but they are no longer something you sit in front of. The forecasts for invisibility are so optimistic that in 2012 Apple allowed themselves to rephrase Norman's predictive statement by putting it in the present tense and binding it to a particular piece of consumer electronics:

> *We believe that technology is at its very best when it is invisible, when you are conscious only of what you are doing, not the device you are doing it with […] iPad is the perfect expression of that idea, it's just this magical pane of glass that can become anything you want it to be. It's a more personal experience with technology than people have ever had.*[4]

In this last sentence, the word "experience" is not an accident, neither is the word "people".

Invisible computers, or more accurately the illusion of the computerless, is destroyed if we continue to talk about "user interfaces". This is why interface design started to rename itself "experience design", whose primary goal is to make users forget that computers and interfaces exist.

With experience design there is only you and your emotions to feel, goals to achieve, tasks to complete.

The field is abbreviated as UXD, where X is for eXperience and U is still for the Users. Wikipedia says Don Norman coined the term UX in 1995. However, in 2012 UX designers avoided using the "U" word in papers and conference announcements, in order not to remind themselves about all those clumsy buttons and input devices of the past. Users were for the interfaces. Experiences, they are for the PEOPLE![5]

In 2008 Don Norman simply ceased to address users as users. At an event sponsored by Adaptive Path, a user interface design company, Nor-

4 Apple Inc, Official Apple (New) iPad trailer (2012), http://www.youtube.com/watch?v=RQieoqCLWDo, access: January 20, 2021.

5 Another strong force behind ignoring the term "user" comes from adepts at gamification. They prefer to address users as gamers. But that's another topic.

man stated, "One of the horrible words we use is 'users'. I am on a crusade to get rid of the word 'users'. I would prefer to call them 'people'."[6]
After enjoying the effect of his words on the audience, he added with a charming smile, "We design for people, we don't design for users."
A noble goal, indeed, but only when perceived in the narrow context of interface design. Here, the use of the term "people" emphasises the need to follow the user-centred as opposed to an implementation-centred paradigm. The use of "people" in this context is a good way to remind software developers that the user is a human being and needs to be taken into account in design and validation processes.
But when you read it in a broader context, the denial of the word "user" in favour of "people" becomes dangerous. Being a user is the last reminder that there is, whether visible or not, a computer, a programmed system you use.
In 2011, new media theoretician Lev Manovich also became unhappy with the word "user". He writes on his blog "For example, how [sic] do we call a person who is interacting with digital media? User? No good."[7]
Well, I can agree that with all the great things we can do with new media – the various modes of initiation and participation, the multiple roles we can fill – it is a pity to narrow it down to "users", but this is what it is. Bloggers, artists, podcasters and even trolls are still users of systems they didn't program. So they – we – are all the users.
We need to take care of this word because addressing people and not users hides the existence of two classes of people – developers and users. And if we lose this distinction, users may lose their rights and the

6 For the video documentation of the talk see https://www.youtube.com/watch?v=WgJcUHC3qJ8, access: January 20, 2021. See also Norman's 2006 essay "Words matter" (2018): "Psychologists depersonalize the people they study by calling them 'subjects.' We depersonalize the people we study by calling them 'users.' Both terms are derogatory. They take us away from our primary mission: to help people. Power to the people, I say, to repurpose an old phrase. People. Human Beings. That's what our discipline is really about." https://jnd.org/words_matter_talk_about_people_not_customers_not_consumers_not_users/, access: January 20, 2021.

7 Lev Manovich, How do you call a person who is interacting with digital media? (2011); http://lab.softwarestudies.com/2011/07/how-do-you-call-person-who-is.html, access: January 20, 2021.

opportunity to protect them. These rights are to demand better software, the ability "to choose none of the above",[8] to delete your files, to get your files back, to fail epically and, back to the fundamental one, to see the computer.

In other words: the Invisible User is more of an issue than an Invisible Computer.

What can be done to protect the term, the notion and the existence of users? What counter-arguments can I find to stop Norman's crusade and dispel Manovich's scepticism? What do we know about a user, apart from the opinion that it is "no good" to be one?

We know that it was not always like this. Before Real Users (those who pay money to use the system) became "users", programmers and hackers proudly used this word to describe themselves. In their view, the user was the best role one could take in relation to their computer.[9]

Furthermore, it is wrong to think that first there were computers and developers and only later users entered the scene. In fact, it was the opposite. At the dawn of the personal computer the user was the centre of attention. The user did not develop in parallel with the computer, but prior to it. Think about Vannevar Bush's "As we may think" (1945), one of the most influential texts in computer culture. Bush spends more words describing the person who would use the Memex than the Memex itself. He described a scientist of the future, a superman. He, the user of the Memex – not the Memex itself – was heading the article.[10]

Twenty years later, Douglas Engelbart, inventor of the pioneering personal computer system NLS, as well as hypertext and the mouse, talked about

8 Borrowed from the subtitle "You may always choose none of the above" of the chapter Choice, in: Douglas Rushkoff, *Program or Be Programmed. Ten Commands for a Digital Age* (New York 2010), p. 46.

9 "The movie Tron (1982) marks the highest appreciation and most glorious definition of this term. […] The relationship of users and programs is depicted as a very close and personal one, almost religious in nature, with a caring and respecting creator and a responsible and dedicated progeny." Olia Lialina and Dragan Espenschied, Do you believe in users?, in: *Digital Folklore* (Stuttgart 2009).

10 Vannevar Bush, As we may think. A top U.S. scientist forsees a possible future world in which man-made machines will start to think. *Life Magazine* (September 19, 1945), pp. 112–124.

his research on the augmentation of human intellect as "bootstrapping" – meaning that human beings, and their brains and bodies, will evolve along with new technology. This is how French sociologist Thierry Bardini describes this approach in his book about Douglas Engelbart: "Engelbart wasn't interested in just building the personal computer. He was interested in building the person who could use the computer to manage increasing complexity efficiently."[11]

And let's not forget the title of J.C.R. Licklider's famous text, the one that outlined the principles for ARPA's command and control research on the real-time system, from which the interactive/personal computer developed – "Man–computer symbiosis" (1960).[12]

When the personal computer was getting ready to enter the market 15 years later, developers thought about who would be model users. At Xerox PARC, Alan Kay and Adele Goldberg introduced the idea of kids, artists, musicians and others as potential users for the new technology. Their paper "Personal dynamic media" from 1977[13] describes important hardware and software principles for the personal computer. But we read this text as revolutionary because it clearly establishes possible users, distinct from system developers, as essential to these dynamic technologies. Another Xerox employee, Tim Mott (aka "the father of user-centred design") brought in the idea of a secretary into the imagination of his colleagues. This image of the "lady with the Royal typewriter"[14] predetermined the designs of Xerox Star, Apple Lisa and further electronic offices.

11 Thierry Bardini, *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing* (Stanford 2000).

12 J.C.R. Licklider, Joseph Carl Robnett, Man–computer symbiosis. *IRE Transactions on Human Factors in Electronics* (1960) pp. 4–11; http://groups.csail.mit.edu/medg/people/psz/Licklider.html, access: January 20, 2021.

13 Alan Kay, Personal dynamic media, [1977], in: *The New Media Reader*, ed. Noah Wardrip-Fruin and Nick Montfort (Cambridge, MA 2003); http://www.newmediareader.com/book_samples/nmr-26-kay.pdf, access: January 20, 2021.

14 See Douglas K. Smith and Robert C. Alexander, *Fumbling the Future: How Xerox Invented, then Ignored, the First Personal Computer* (New York 1999), p. 110.

So, it is important to acknowledge that users existed prior to computers, that they were imagined and invented – users are a figment of the imagination. As a result of their fictive construction, they continued to be reimagined and reinvented through the 70s, 80s, 90s, and the new millennium. But however reasonable, or brave, or futuristic, or primitive these models of users were, there is a constant.

Let me refer to another guru of user-centred design, Alan Cooper. In 2007, when the "U" word was still allowed in interaction design circles, he and his colleagues shared their secret in About Face: The Essentials of Interaction Design:

> *As an interaction designer, it's best to imagine that users – especially beginners – are simultaneously very intelligent and very busy.*[15]

This is very kind advice (and one of the most reasonable books on interface design, btw) and can be translated roughly as "hey, front-end developers, don't assume that your users are more stupid than you, they are just busy". But it is more than this. What the second part of this quote gets to so importantly is that users are people who are very busy with something else.

Alan Cooper is not the one who invented this paradigm, nor was it even Don Norman with his concentration on the task rather than the tool. It originated in the 1970s. Listing the most important computer terms of that time, Ted Nelson mentions so-called "user-level systems" and states that these "User-level systems, [are] systems set up for people who are not thinking about computers but about the subject or activity the computer is supposed to help them with".[16] Some pages before he claims:[17]

15 Alan Cooper, Robert Reimann, David Cronin, *About Face 3: The Essentials of Interaction Design* (Indianapolis 2007), p. 45.

16 Ted Nelson, *Computer Lib/Dream Machines* (author's edition, 1987), p. 9.

17 Ibid., p. 3.

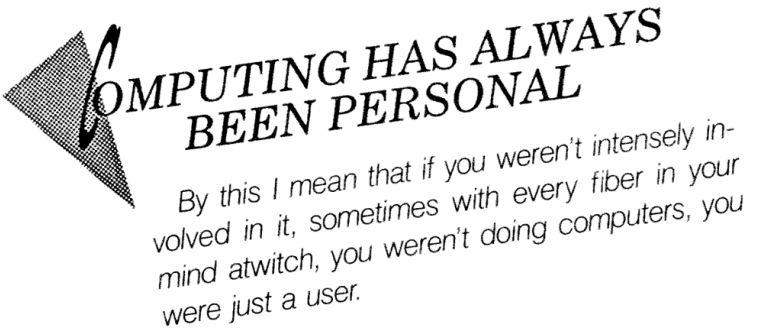


Fig. 1: Ted Nelson, Computer Lib/Dream Machines (author's edition, 1987), p. 3.

One should remember that Ted Nelson was always on the side of users and even "naive users", so his bitter "just a user" means a lot.

The alienation of users from their computers started in Xerox PARC with secretaries, as well as artists and musicians. And it never stopped. Users were seen and marketed as people whose real jobs, feelings, thoughts, interests, talents – everything that matters – lie outside of their interaction with personal computers.

For instance, in 2007, when Adobe, the software company whose products are dominating the so-called "creative industries", introduced version 3 of Creative Suite, they filmed graphic artists, video makers and others talking about the advantages of this new software package. Of particular interest was one video of a web designer (or an actress in the role of a web designer): she enthusiastically demonstrated what her new Dream Weaver could do, claiming that, in the end, "I have more time to do what I like most – being creative". The message from Adobe is clear. The less you think about source codes, scripts, links and the Web itself, the more creative you are as a web designer. What a lie. I liked to show this to fresh design students as an example of misunderstanding the core of the profession.

This video is not online anymore, but actual ads for Creative Suite 6 are not much different – they feature designers and design evangelists talking about unleashing, increasing and enriching creativity as a direct result

of fewer clicks to achieve this or that effect.[18] In the book *Program or Be Programmed*, Douglas Rushkoff describes similar phenomena:

> *[…] We see actual coding as some boring chore, a working-class skill like bricklaying, which may as well be outsourced to some poor nation while our kids play and even design video games. We look at developing the plots and characters for a game as the interesting part, and the programming as the rote task better offloaded to people somewhere else.*[19]

Rushkoff states that code writing is not seen as a creative activity, but the same applies to engagement with the computer in general. It is not seen as a creative task or as "mature thought".
In "As we may think", while describing an ideal instrument that would augment the scientist of the future, Vannevar Bush observes:

> *For mature thought there is no mechanical substitute. But creative thought and essentially repetitive thought are very different things. For the latter there are, and may be, powerful mechanical aids*[20]

In contrast to this, users, as imagined by computer scientists, software developers and usability experts are the ones whose task is to spend as little time as possible with the computer, without wasting a single thought on it. They require a specialised, isolated app for every "repetitive thought", and, most importantly, delegate drawing the border between creative and repetitive, mature and primitive, real and virtual, to app designers.
There are periods in history, moments in life (and many hours a day!) where this approach makes sense, when delegation and automation are

18 See for example the trailers for Adobe Creative Suite 6 (2012); https://www.adobe.com/creativecloud.html, access: March 10, 2021.

19 Rushkoff, *Program or Be Programmed*, p. 131.

20 Vannevar Bush, As we may think. *The Atlantic Monthly* 176 (1945), pp. 101–108; HTML version: https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/?single_page=true, access: March 10, 2021.

Fig. 2: "A Scientist of the Future" – title picture of Vannevar Bush's "As we make think" published in the September 10, 1945 issue of *Life Magazine*, and Russian travel blogger Sergey Dolya (photo by Mik Sazonov, 2012; https://sergeydolya.livejournal.com/510565.html , access: March 10, 2021). Collage by Olia Lialina.

required and enjoyed. But in times when every aspect of life is computerized, it is not possible to accept "busy with something else" as a norm. So let's look at another model of users that evolved outside and despite usability experts' imagination.

General Purpose, "Stupid" and Universal

In "Why interfaces don't work" Don Norman heavily criticises the world of visible computers, visible interfaces and users busy with all this. Near the end of the text, he suggests the source of the problem:

> *We are here in part, because this is probably the best we can do with today's technology and, in part, because of historical accident.*
> *The accident is that we have adapted a general-purpose technology to very specialized tasks while still using general tools.*[21]

In December 2011, science fiction writer and journalist Cory Doctorow gave a marvellous talk at the 28th Chaos Communication Congress in Berlin titled "The coming war on general computation".[22] He claimed that there is only one possibility for computers to truly become appliances, the tiny, invisible, comfortable one-purpose things Don Norman was preaching about: and that is to be loaded with spyware. He explains:

> *So today we have marketing departments who say things like "[…] Make me a computer that doesn't run every program, just a program that does this specialized task, like streaming audio, or routing packets, or playing Xbox games" […] But that's not what we do when we turn a computer into an appliance. We're not making a computer that runs only the "appliance" app; we're making a computer that can run every program, but which uses some combination of*

21 Norman, Why interfaces don't work, p. 218.

22 Transcript: https://joshuawise.com/28c3-transcript#the_coming_war_on_general_computation; video: https://www.youtube.com/watch?v=HUEvRyemKSg, access: March 21, 2021.

> *rootkits, spyware, and code-signing to prevent the user from knowing which processes are running, from installing her own software, and from terminating processes that she doesn't want. In other words, an appliance is not a stripped-down computer – it is a fully functional computer with spyware on it out of the box.*

By "fully functional computer", Doctorow means the general-purpose computer, or as US mathematician John von Neumann referred to it in his 1945 "First draft of a report on the EDVAC", the "all-purpose automatic digital computing system".[23] In this paper he outlined the principles of digital computer architecture (von Neumann Architecture), where hardware was separated from the software, and from this the so-called "stored program" concept was born. In the mid 40s, the revolutionary impact of it was that "by storing the instructions electronically, you could change the function of the computer without having to change the wiring."[24]

Today the rewiring aspect does not have to be emphasised, but the idea itself that a single computer can do everything is essential, and that it is the same general-purpose computer behind "everything" – from dumb terminals to super computers.

Doctorow's talk is a perfect entry point to get oneself acquainted with the subject. To go into the history of the war on general computation in more depth, you may consider reading Ted Nelson. He was the first to attract attention to the significance of the personal computer's all-purpose nature. In 1974, in his glorious fanzine *Computer Lib*, which aimed to explain computers to everybody, he writes in caps lock:

> *COMPUTERS HAVE NO NATURE AND NO CHARACTER*
>
> *Computers are, unlike any other piece of equipment, perfectly BLANK. And that is how we have projected on it so many different faces.*[25]

23 John von Neumann, Introduction to The first draft report on the EDVAC", 1945; http://web.mit.edu/STS.035/www/PDFs/edvac.pdf , access: March 10, 2021.

24 M. Mitchell Waldrop, *The Dream Machine* (San Francisco 2001), p. 62.

25 Nelson, *Computer Lib*, p. 37.

Some great texts written this century are *The Future of the Internet and How to Stop It* (New Haven and London 2008), by Jonathan Zittrain, and of course *The Future of Ideas* (New York 2001), by Lawrence Lessig. Both authors are more concerned with the architecture of the Internet than the computer itself, but both write about the end-to-end principle that lies at the Internet's core – meaning that there is no intelligence (control) built into the network. The network stays neutral or "stupid", simply delivering packets without asking what's inside. It is the same with the von Neumann computer – it just runs programs.

The works of Lessig, Zittrain and Doctorow do a great job of explaining why both computer and network architectures are neither historic accidents nor "what technology wants".[26] The stupid network and the general-purpose computer were conscious design decisions.

For Norman, further generations of hardware and software designers and their invisible users dealing with general-purpose technology are both accident and obstacle. For the rest of us, the rise and use of General-Purpose Technology is the core of New Media, Digital Culture and Information Society (if you believe that something like this exists). General-purpose computers and stupid networks are the core values of our computer-based time and the driving force behind all the wonderful and terrible things that happen to people who work and live with connected computers. These prescient design decisions have to be protected today, because technically it would be no big deal to make networks and computers "smart", i.e. controlled.

What does all this have to do with "users" versus "people", apart from the self-evident fact that only the users who are busy with computers at least a little bit – to the extent of watching Doctorow's video to the end – will fight for these values?

I would like to apply the concept of General-Purpose Technology to users by flipping the discourse around and redirecting attention from technol-

26 See Kevin Kelly, *What Technology Wants* (London 2010).

ogy to the user who was formed over three decades of adjusting general-purpose technology to their needs: *The General-Purpose User.*

General-Purpose Users can write an article in their email client, lay out their business card in Excel and shave in front of a web cam. They can also find a way to publish photos online without Flickr, tweet without Twitter, like without Facebook, make a black frame around pictures without Instagram, remove a black frame from an Instagram picture and even wake up at 7:00 without a "wake up at 7:00" app.

Maybe these users could more accurately be called "Universal Users" or "Turing Complete Users", as a reference to the Universal Machine, also known as the Universal Turing Machine – Alan Turing's conception of a computer that can solve any logical task, given enough time and memory. Turing's 1936 vision and design predated and most likely influenced von Neumann's "First draft" and all-purpose machine.[27]

But whatever name I choose, what I mean are users who have the ability to achieve their goals regardless of the primary purpose of an application or device. Such users will find a way to their aspiration without an app or utility programmed specifically for it. The universal user is not a super-user, not half a hacker. A universal user is not an exotic type of user.

There can be different examples and levels of autonomy that users can imagine for themselves, but the capacity to be universal is still in all of us. Sometimes it is a conscious choice not to delegate particular jobs to the computer, and sometimes it is just a habit. Most often it is no more than a click or two that uncovers your general-purpose architecture.

For instance, you can decide not to use Twitter at all and instead inform the world about your breakfast through your own website. You can use LiveJournal as if it is Twitter, you can use Twitter as Twitter, but instead of following people, visit their profiles as you would visit a home page.

You can have two Twitter accounts and log in to one in Firefox, and the other in Chrome. This is how I do it and it doesn't matter why I prefer to

27 Alan Turing, On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 2, 42 (1), pp. 230–265; received May 28, 1936.

manage it this way. Maybe I don't know that an app for managing multiple accounts exists, maybe I knew but didn't like it, or maybe I'm too lazy to install it. Whatever, I found a way. And you will do as well.

A universal user's mindset (it is a mindset, not a set of rules, not a vow) means to liaise with hardware and software. Behaviour that is antipodal to the "very busy" user. This kind of interaction makes the user visible, most importantly to themselves. And, if you wish to think about it in terms of interface design and UX, it is the ultimate experience.

Does this mean that to deliver this kind of user experience the software industry needs to produce imperfect software or hold itself back from improving existing tools? Of course not! Tools can be perfect.

Though the idea of perfect software could be revised, taking into account that it is used by the general-purpose user, valuing ambiguity and users' involvement.

And thankfully ambiguity is not that rare. There are online services where users are left alone to use or ignore features. For example, the developers of Twitter didn't take measures that prevent me from surfing from profile to profile of people I don't follow. The Dutch social network Hyves allows their users to mess around with background images so that they don't need any photo albums or instagrams to be happy. Blingee.com, whose primary goal is to let users add glitter to their photos, allows them to upload whatever stamps they want – not glittery, not even animated. It accepts everything and just delivers merged layers to the users.

I can also mention here an extreme example of a service that nourishes the user's universality – myknet.org – an Aboriginal social network in Canada. It is so "stupid" that users can repurpose their profiles every time they update them. Today it functions as a Twitter feed, yesterday it was a YouTube channel, and tomorrow it might be an online shop. Never mind that it looks very low-tech and like it was made 17 years ago, it works! In general, the WWW, outside of Facebook, is an environment open for interpretation.

Still, I have difficulties finding a site or an app that actually addresses the users and sees their presence as a part of the workflow. This maybe

sounds strange, because all Web 2.0 is about pushing people to contribute, and "emotional design" is supposed to be about establishing personal connections between people who made the app and people who bought it, but I mean something different. I mean a situation when the workflow of an application has gaps that can be filled by users, where smoothness and seamlessness are broken and some of the final links in the chain are left for the users to complete.

I'll leave you with an extreme example, an anonymous (probably student) project:

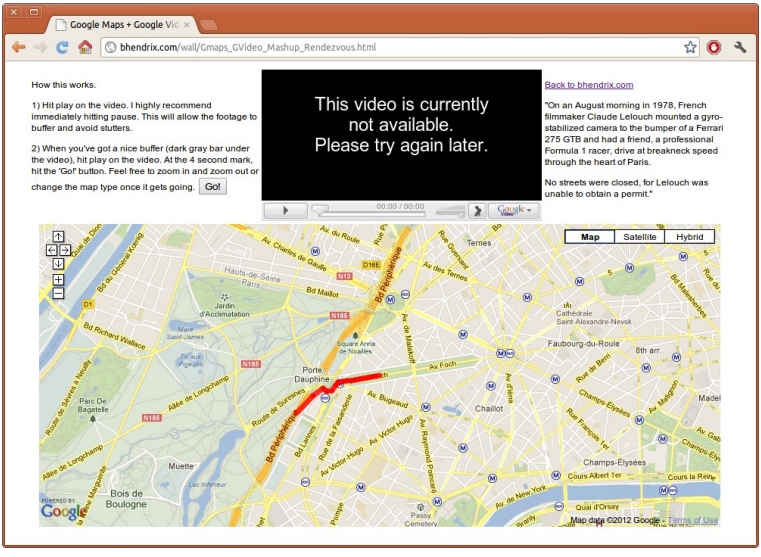


Fig. 3: "Google Maps + Google Video + Mashup – Claude Lelouch's Rendezvous"

It was made in 2006, at the very rise of Web 2.0[28], when the mash-up was a very popular cultural, mainstream artistic form. Artists were celebrating new convergences and a blurring of the borders between different pieces of software. Lelouch's Rendezvous is a mash-up that puts on the same

28 Web 2.0 was supposed to be a complete merging of people and technology but was again progressing alienation and keeping users and developers apart. People were driven from self-made home pages to social networks.

page the famous racing film of the same name and a map of Paris, so that you can follow the car in the film and see its position on the Google map at the same time. But the author failed (or perhaps didn't intend) to synchronise the video and the car's movement on the map.
As a result, the user is left with the instruction: "Hit play on the video. […] At the 4 second mark, hit the 'Go!' button."
The user is asked not only to press one but two buttons! It suggests that we can take care of it ourselves, that we can complete a task at the right moment. The author obviously counted on users' intelligence and had never heard that they were "very busy".
The fact that the original video file that was used in the mash-up was removed makes this project even more interesting. To enjoy it, you'll have to go to YouTube and look for another version of the film. I found one, which means you'll succeed as well.
There is nothing one user can do that another can't, given enough time and respect. Computer users are Turing Complete.

* * *

When Sherry Turkle, Douglas Rushkoff and other great minds state that we need to learn programming and understand our computers in order to not be programmed and that we should "demand transparency of other systems"[29], I couldn't agree more. If the approach to computer education in schools was to switch from managing particular apps to writing apps it would be wonderful. But apart from the fact that it is not realistic, I would say it is also not enough. I would argue it is wrong to say either you understand computers or you are the user.[30]

29 "Politics is a system, complex to be sure, all the same. If people understand something as complicated as a computer, they will demand greater understanding of other things." Respondent's statement, discussed in Sherry Turkle, *The Second Self: Computers and the Human Spirit* (Cambridge, MA 2004), p. 163.

30 "Instead of teaching programming, most schools with computer literacy curriculums teach programs […] The bigger problem is that their entire orientation to computing will be from [the] perspective of users" Rushkoff, *Program or Be Programmed*, p. 130.

An effort must be made to educate the users about themselves. There should be understanding of what it means to be a user of an 'all-purpose automatic digital computing system'.

General-purpose users are not a historic accident or a temporary anomaly. We are the product of the "worse is better" philosophy of UNIX, the end-to-end principle of the Internet, the "under construction" and later "beta" spirit of the Web. All these designs that demand attention, and ask for forgiveness and engagement, formed us as users, and we are always adjusting, improvising and at the same time taking control. We are the children of the misleading and clumsy desktop metaphor, we know how to open doors "with no knob".[31]

We general purpose users – not hackers and not people – who are challenging, consciously or subconsciously, what we can do and what computers can do, are the ultimate participants of the man–computer symbiosis. Not exactly the kind of symbiosis Licklider envisioned, but a true one.

31 "Direct-manipulation systems, like the Macintosh desktop, attempt to bridge the interface gulf by representing the world of the computer as a collection of objects that are directly analogous to objects in the real world. But the complex and abundant functionality of today's new applications – which parallels people's rising expectations about what they might accomplish with computers – threatens to push us over the edge of the metaphorical desktop. The power of the computer is locked behind a door with no knob." Brenda Laurel, *Computers as Theatre* (Amsterdam 1993), p. xviii.

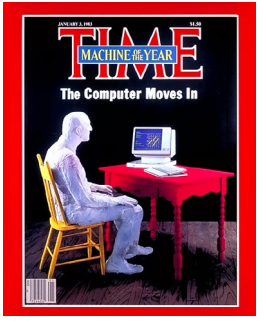
Appendix A: Subjects of Human–Computer Interaction (updated May 2021)

	UX	Web 2.0	Cloud Computing	Gamification	IoT	AI	NFT	HRI
computer	technology	social network	The Cloud	epic win	warehouse	voice	miners	humanoid
user interface	experience	submit button	upload button	epic win	milk	conversation	coins	empathy
users	people	you	download button	gamer	fridge	voice	minter	human

Appendix B: Users Imagined

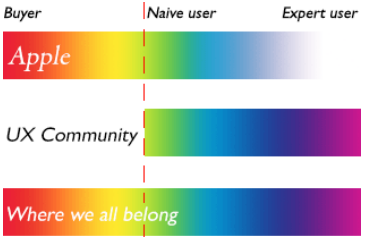
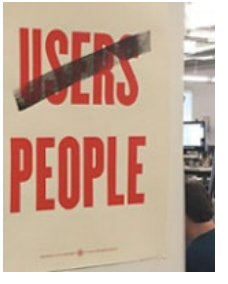
year	source	imagined user	statement
1945	Vannevar Bush, As we may think http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881, access April 10, 2021.	Scientist	"One can now picture a future investigator in his laboratory. His hands are free, and he is not anchored. As he moves about and observes, he photographs and comments."
1962	Douglas Engelbart, Augmenting human intellect http://www.dougengelbart.org/pubs/augment-3906.html, access April 10, 2021.	Knowledge Worker Intellectual Worker Programmer	"Consider the intellectual domain of a creative problem solver […]. These […] could very possibly contribute specialized processes and techniques to a general worker in the intellectual domain: formal logic—mathematics of many varieties, including statistics—decision theory—game theory—time and motion analysis—operations research—classification theory—documentation theory—cost accounting, for time, energy, or money—dynamic programming—computer programming."
"1970s"	J.C.R. Licklider, Some reflections on early history, in: *A History of Personal Workstations*, ed. Adele Goldberg (New York, NY: Association for Computing Machinery, 1988), p. 119.	Real Users	"People who are buying computers, especially personal computers, just aren't going to take a long time to learn something. They are going to insist on using it awfully quick."

1974	Ted Nelson, *Computer Lib/Dream Machines* (Tempus Books/Microsoft Press; revised edition, 1987), p. 9.	Naive User	"Person who doesn't know about computers but is going to use the system. Naive user systems are those set up to make things easy and clear for such people. We are all naive users at some time or other; it's nothing to be ashamed of. Though some computer people seem to think it is."
1975	Tim Mott, as quoted in: Douglas K. Smith, *Fumbling the Future – How Xerox Invented, then Ignored, the First Personal Computer* (Indiana: Iuniverse, 1999), p. 110.	Lady with the Royal Typewriter	"My model for this was a lady in her late fifties who had been publishing all her life and still used a Royal typewriter."
1977	Alan Kay, Personal Dynamic Media http://www.newmediareader.com/book_samples/nmr-26-kay.pdf, access April 10, 2021.	Children Artists Musicians	"Another interesting nugget was that children really needed as much or more computing power than adults were willing to settle for when using a timesharing system. […] The kids […] are used to finger-paints, water colors, color television, real musical instruments, and records."
1982	Steven Lisberger, TRON	Deity	– "You believe in the users?" – "Yes, sure. If I don't have a user, then who wrote me?"

1983	TIME Magazine		The "person of the year" is a machine: "Machine of the year: The computer moves in"
1993	Eric S. Raymond, September that never ended http://www.catb.org/~esr/jargon/html/S/September-that-never-end-ed.html, access April 10, 2021.	Clueless Newbies	"September that never ended: All time since September 1993. One of the seasonal rhythms of the Usenet used to be the annual September influx of clueless newbies who, lacking any sense of netiquette, made a general nuisance of themselves. This coincided with people starting college, getting their first internet accounts, and plunging in without bothering to learn what was acceptable."

1996	Eric S. Raymond, *The New Hacker's Dictionary* (Cambridge, MA: MIT Press, 1996).	hackers = Implementors lamers = Users	"hacker n. […] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary." p. 233 "lamer n. […] Synonym for luser, not used much by hackers but common among warez d00dz, crackers and phreakers. Oppose elite. Has the same connotations of self-conscious elitism that use of luser does among hackers." p. 275
2006	TIME Magazine	YOU	

2008	Don Norman, Talk at UX Week 2008 https://www.youtube.com/watch?v=WgJcUHC3qJ8	People	"I'd prefer to call them people."
2009	Sir Tim Berners-Lee, The next Web TED Talk https://www.ted.com/talks/tim_berners_lee_the_next_web, access April 10, 2021.	Them	"20 years ago […] I invented the World Wide Web."
2012	Jack Dorsey, executive chairman of Twitter: Let's reconsider our "users" https://jacks.tumblr.com/post/33785796042/lets-reconsider-our-users, access April 10, 2021.	Customer	"If I ever say the word 'user' again, immediately charge me $140."
2012	Janet Murray, interaction designer, educator, author of *Hamlet on the Holodeck*, in introduction to *Inventing the Medium. Principles of Interaction Design as a Cultural Practice* (Cambridge, MA: MIT Press, 2011).	Interactor	"[User] is another convenient and somewhat outdated term, like "interface" […] A user may be seeking to complete an immediate task; an interactor is engaged in a prolonged give and take with the machine." p.11

2013	Bruce Tognazzini, principal of Nielsen Norman Group, The third user https://asktog.com/atc/the-third-user/, access April 10, 2021.	Buyer	
2011	Facebook	People (public relations)	 "All of our dashboards, instead of saying 'daily average users,' say 'daily average people'." — Robinson Meyer, Facebook no longer likes the word "Users". *The Atlantic*, December 11, 2014

2014 (New)	Facebook	Target (internally)	The privacy activist group Europe vs Facebook analysed a data set disclosed by Facebook in 2012, finding out that "target" is the name of the item containing a user's ID number and the date of its generation. http://europe-v-facebook.org/, access April 10, 2021.