Designing the User

GALO CANIZARES
The Ohio State University

As our planet grows increasingly reliant on platforms and software for every conceivable task, a new subjectivity has emerged: the user. In general, users are figures at the tail end of computed activities and are in active negotiation with anonymous creators (programmers) about how tasks should be done. In the design professions, users combine their disciplinary knowledge with optimized workflows to produce solutions to design problems. But this ideal relationship between users and software is a constructed myth in itself. If the abundance of online help forums are any indication, users are far more complex beings than software engineers ever expected. In this sense, the user is a fickle subject that warrants further study, especially in design professions where the role of designer is becoming increasingly synonymous with user.

This paper puts forth a set of criteria for analyzing users. Putting aside the techno-positivism pervading traditional user-centered design, the questions asked here prioritize a narrative approach focusing on conflicts, absurd scenarios, and weird collaborations between software and users. Designing the user thus could be seen as an exercise in creating problems as much as solving them.

USER DESIGN

The world today runs on software. And architecture, as the discipline charged with designing the built portion of the world, is no exception. The field relies on a variety of applications, digital tools, and information management systems vaguely summarized as *design software*. Along with this technical shift, however, we can also identify a larger techno-political reconfiguration in how we perceive ourselves as designers, inhabitants, citizens, managers, and actors. In the world of software, we are all *users*.

The user is a new subjectivity borne out of our everyday interactions with software systems. In general, users are figures at the tail end of computed activities, those that execute predetermined functions in a certain order to achieve results. In the design professions, users are in active negotiation with anonymous creators (programmers) about how tasks should be done. They combine their disciplinary knowledge with optimized workflows to produce solutions to design problems. But users are far more complex than software engineers ever expected.

The user is foremost a designed subject. There are no default users, only the perception of users with default qualities. Users are designed by computer programmers, tech CEOs, software engineers, but also by firm managers, complex algorithms, interaction designers, architects, and lifestyle gurus (to name a few). At times, you are a user—when you use software; other times, you might be designing a user—when you imagine someone using your designed space/tool/interface. As Benjamin Bratton has written in his book, *The Stack: On Software and Sovereignty,* “the User is not a type of creature but a category of agents; it is a position within a system without which it has no resolve or essential identity.”1 In other words, the user does not constitute a whole being, but is nevertheless fundamentally tied to who we are as independent actors. By using Facebook or AutoCAD, we are users, and while that is only part of our selves, this virtual-ontological signifier is quite consequential in the age of software (think: do you identify as a Mac or PC user? Chrome or Firefox?).

Aside from software, users in architecture are significant because they take on a variety of roles. Typically, they are clients or the projected inhabitants of our buildings and spaces. They appear in our plans, sections, elevations, and perspectival imagery as the population we’re designing for. And yet, with the rise of information systems and data-centric applications, the definition of an architectural user is becoming as muddled and intricate as the software user. Architects are now users loyal to certain software suites such as Autodesk Revit or Adobe Photoshop, algorithmically approximated taste curators on sites like Instagram and Pinterest, lists of preferences with keyboard shortcuts and macros stored in our personal or work computers, and in some cases race and gender quotas that meet diversity goals in the industry. We should therefore ask, how are these perceptions changing the way we regard design and ourselves as designers?

FIVE ARCHITECTURAL USERS

Given that users are designed subjects, it follows that stereotypes would become commonplace in certain fields. This is perhaps most evident in computing, where signifiers like “nerd” or “geek” suggest an imagined user-type, often reinforced by pop culture imagery. While stereotypes are indeed harmful in fabricating unjust attitudes towards specific individuals and groups, the formation of such a type in

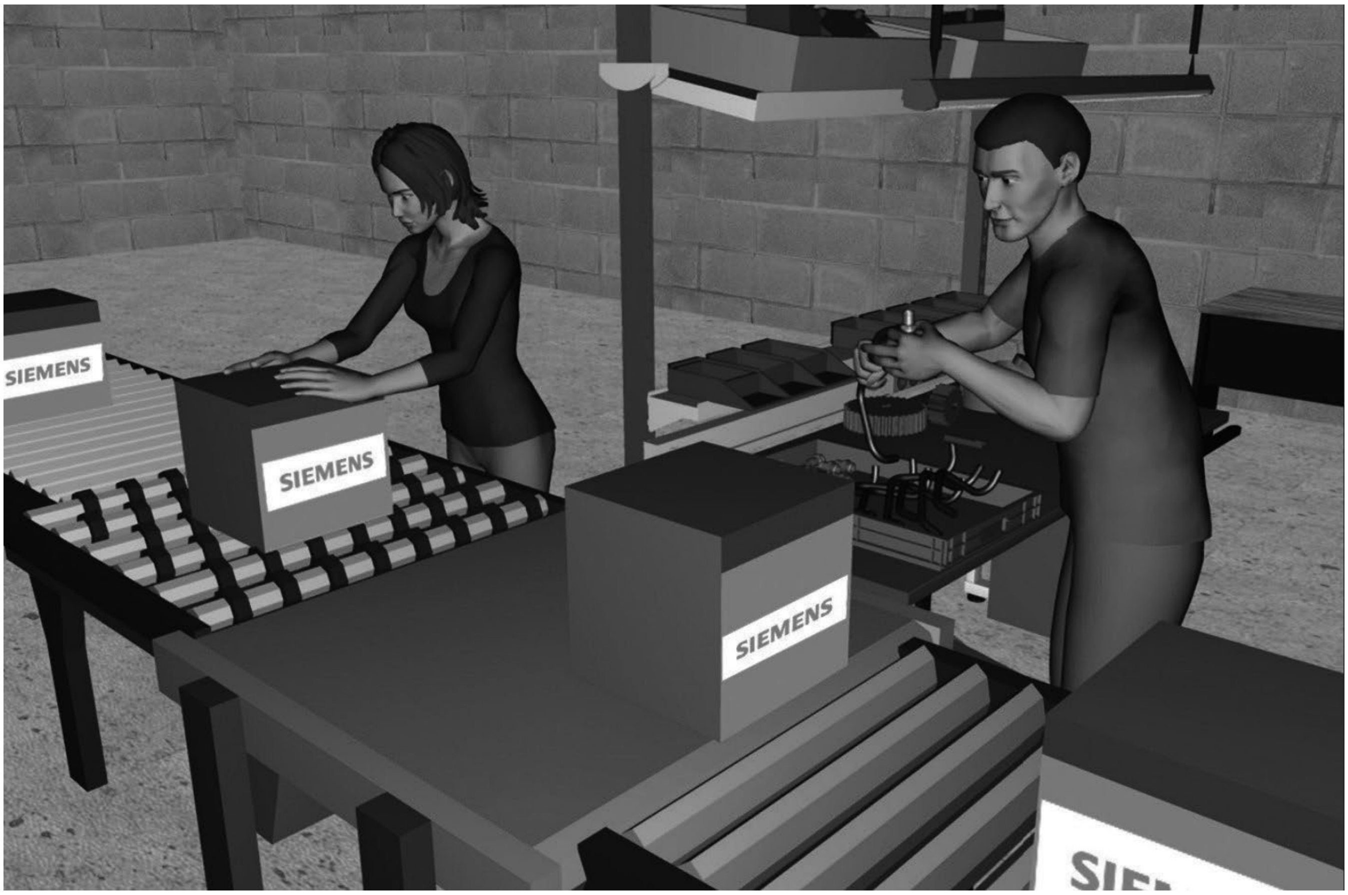


Figure 1. Tecnomatix Jack simulation software, Siemens Corp., 2013.

the first place is precisely what I describe as *design of the user*. In their essay "Do You Believe in Users?" Olia Lialina and Dragan Espenschied follow a changing perception of users in the late 20th Century. Beginning with Vannevar Bush—who imagined that those who use computers would be primarily scientists—and ending with Silicon Valley companies like Apple—who view users primarily as naive consumers, looking for convenience and not concerned with the intricacies of computation—Lialina and Espenschied's text lays the foundations for a discourse on users that spans digital culture, economics, pop culture, and sociology.[2] For us in the field of architecture, it is helpful to begin with their assessment of user subjectivities and explore how they manifest themselves in our own discipline.

Architectural users could range from clients, inhabitants, citizens, consumers, hackers, makers, and designers. Each of these suggests a different sphere of operating and a wide variety of attitudes towards information systems. A citizen, for example, is an incredibly complex user-type consisting of an individual member of a sovereign state, whose data is traceable by that state, and who is legally bound the laws of that state. Architects as citizens operate quite differently based on their citizenship. North Korean architects have a significantly different set of protocols for designing than North American architects. Citizenship is thus as much a component of a user's identity as the tools they use.

But because this discussion on users can get murky fast, In this paper I will only address five architecture-specific user-types: the Client-User, the Designer-User, the Superuser, the End-User, and the Data-User.[3] Each of my identified user-types depends on a specific point of view, reinforcing the claim that users are not objective characters but highly malleable subjectivities connected to labor, technology, philosophy, and specific systems of knowledge. In presenting these five stereotypes, the goal is to reflect on how technology shapes subjects and perceptions of those involved in design. It should be noted that this is not an exercise that excludes bias, but rather one that puts biases directly on the table so they might be discussed at length.

CLIENT-USER (AS IMAGINED BY A DESIGNER)

The Client-User is the subject of many 20th Century anthropometric studies. These range from ergonomic and physiological to behavioral and psychological. At its core, the Client-User is a measurable figure, an average taken from a larger sample. In short, it is the subject that will use whatever it is we design.

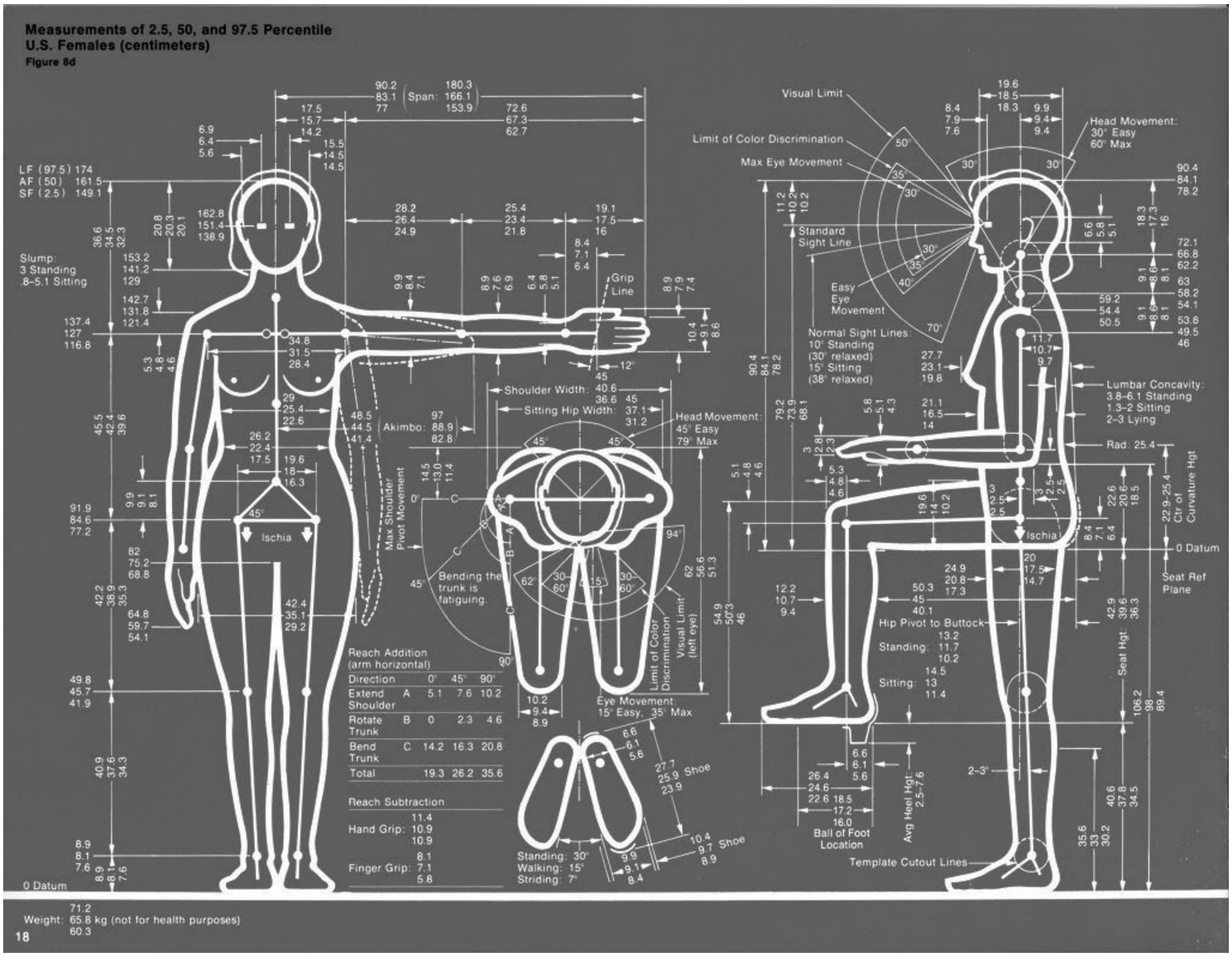
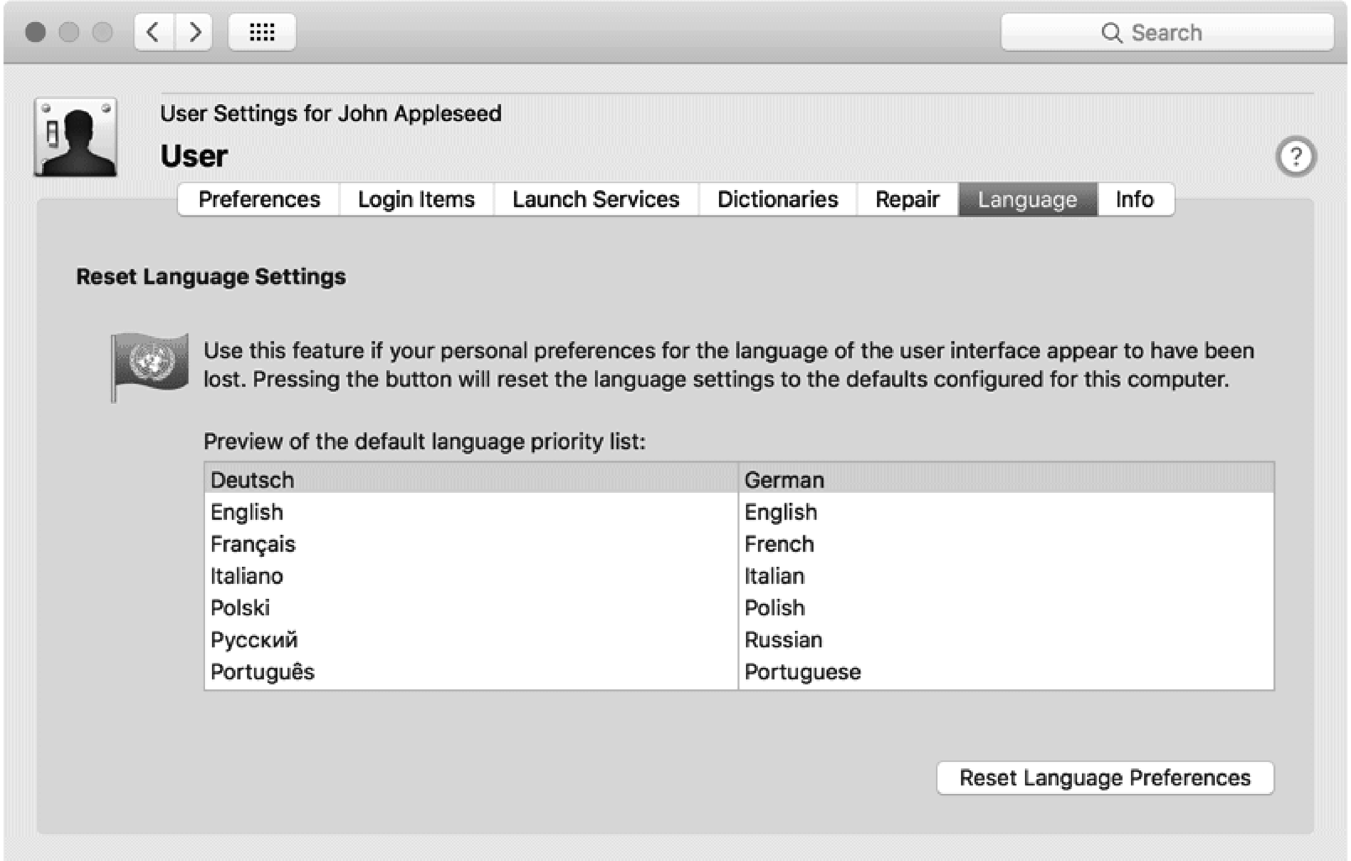

Measurements of 2.5, 50, and 97.5 Percentile
U.S. Females (centimeters)
Figure 8d
Span: 180.3 166.1 153.9
LF (97.5) 174
AF (50) 161.5
SF (2.5) 149.1
Slump: 3 Standing .8–5.1 Sitting
Grip Line
Shoulder Width: 40.6
Sitting Hip Width: 37.1
Akimbo: 97 88.9 82.8
Head Movement: 45° Easy 79° Max
Max Shoulder Pivot Movement
Bending the trunk is fatiguing.
Ischia
Reach Addition (arm horizontal)
Direction 0° 45° 90°
Extend Shoulder A 5.1 7.6 10.2
Rotate Trunk B 0 2.3 4.6
Bend Trunk C 14.2 16.3 20.8
Total 19.3 26.2 35.6
Reach Subtraction
Hand Grip: 11.4 10.9 10.9
Finger Grip: 8.1 7.1 5.8
Limit of Color Discrimination
Visual Limit (left eye)
Eye Movement: 15° Easy, 35° Max
Shoe
Standing: 30° Walking: 15° Striding: 7°
Visual Limit
Limit of Color Discrimination
Max Eye Movement
Standard Sight Line
Easy Eye Movement
Normal Sight Lines: 10° Standing (30° relaxed) 15° Sitting (38° relaxed)
Head Movement: 30° Easy 60° Max
Lumbar Concavity: 3.8–6.1 Standing 1.3–2 Sitting 2–3 Lying
Rad: 25.4
Ctr of Curvature Hgt
0 Datum
Seat Ref Plane
Seat Hgt
Hip Pivot to Buttock
Standing: 13.2 11.7 10.2
Sitting: 14.5 13 11.4
Ball of Foot Location
Avg Heel Hgt: 2.5–7.6
Template Cutout Lines
0 Datum
Weight: 71.2 65.8 60.3 kg (not for health purposes)
18

Figure 2: Henry Dreyfuss and Alvin R. Tilley, The Measure of Man wall chart, 1966.

In the 1950s, Henry Dreyfuss and Alvin R. Tilley conducted a series of studies to arrive at general principles and metrics for human-centered design. They summarized their findings in their book, *Designing for People*—a manual for successful ergonomic, anthropocentric industrial design. In it, Dreyfuss explains that designers must fit the machine to the human, not the other way around. *Designing for People* was not only praised for its humanist vision, but also for the graphic catalog of human data it included. Largely derived from the military (for men) and the fashion industry (for women), the drawings presented a range of percentiles of body height, arm length, postures, and other human measurements.

But the idea of a quantifiable standard or average is fraught with dubious assumptions. Dreyfuss and Tilley's biases are clearly present in their analysis. By using primarily military data which skewed heavily Caucasian and male, they excluded a large percentage of the population; their reliance on abled body types elevated the Client-User to a standard, bipedal, fit individual; and with their labeling of the "typical American users of products and spaces"[4] as "Joe" and "Josephine," it is difficult to see these average users as anything other than white individuals.

While the Client-User is who we generally design for, the field of anthropometry inevitably skews our perception of humans. Much more data has been compiled since Dreyfuss and Tilley's study; data that is now embedded into software systems like Tecnomatix Jack and other digital human model (DHM) software.[5] These tools not only catalog the various proportions and limits of human movement, but they also present specific human imagery back to us as we form an ideal image of who our user might be. Joe and Josephine have become Tecnomatix's Jack and Jill, default figures used as stand-ins for the entire human race. What we perceive as the Client-User is therefore an amalgam of recorded human data—largely derived from military studies like the ANSUR II[6]—and our own biases and idealized subjects.



User Settings for John Appleseed

User

Reset Language Settings

Use this feature if your personal preferences for the language of the user interface appear to have been lost. Pressing the button will reset the language settings to the defaults configured for this computer.

Preview of the default language priority list:

Deutsch	German
English	English
Français	French
Italiano	Italian
Polski	Polish
Русский	Russian
Português	Portuguese

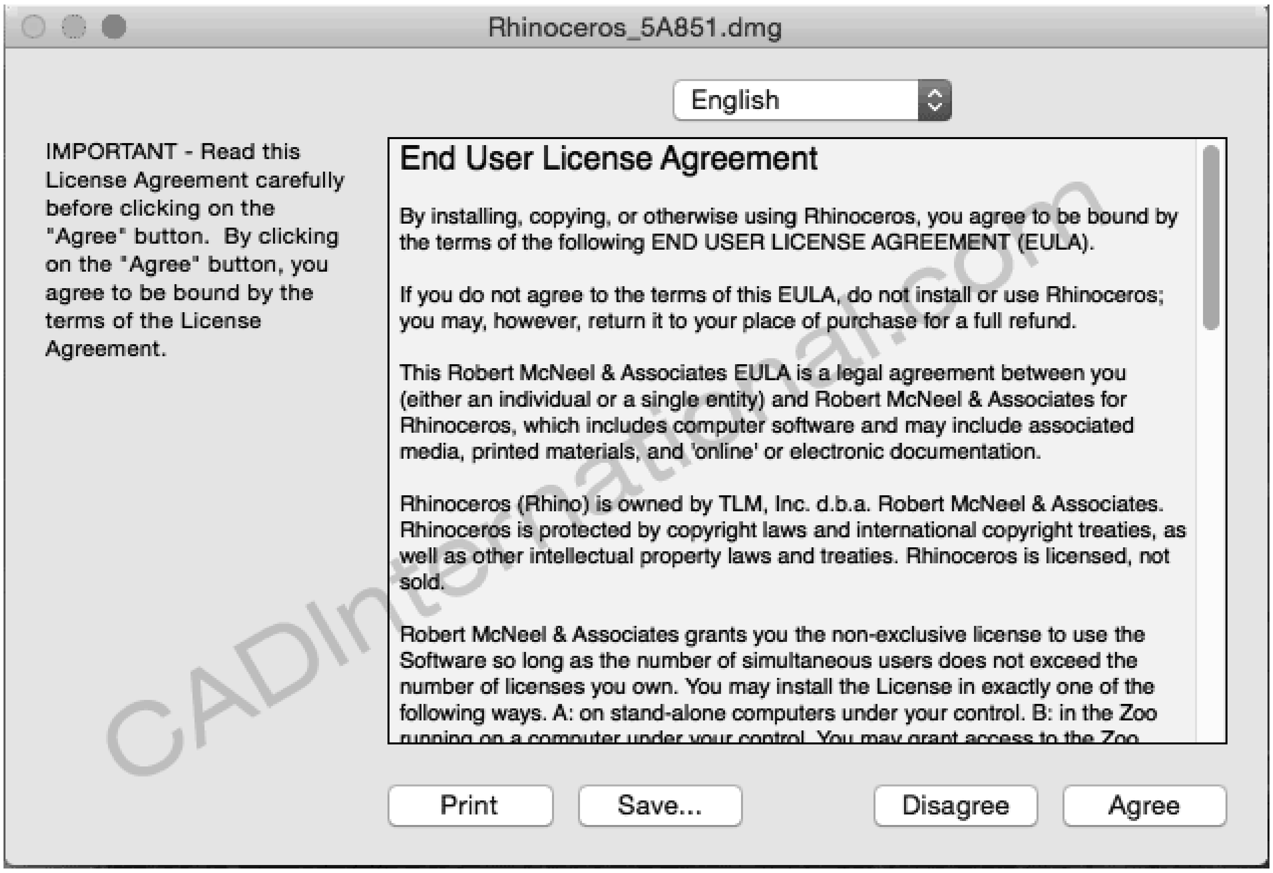
Figure 3: Typical user preferences, MacOS.

DESIGNER-USER (AS IMAGINED BY A PROGRAMMER)

"In the minds of many persons an industrial designer is a brisk, suave character, brimming with confidence, who bustles around factories and stores, streamlining stoves and refrigerators that aren't going anywhere, reshaping doorknobs, and squinting at this year's automobiles and arbitrarily deciding that next year's fenders should be two and three eighths inches longer."[7] This is the "caricature" that Henry Dreyfuss used to introduce *Designing for People*. Here we can identify the counterpart to the Client-User: the Designer-User.

The Designer-User is the subject who designs for clients, and in doing so uses certain tools. Today they are most often imagined by a programmer or software engineer. Somewhat like the Client-User, they are the subject of a problem-solving exercise, which would eventually result in a tool, application, procedure or other technical solution. Unlike the Client-User, however, the metrics used to arrive at a solution are less organic, and much more interpretive. They require a discussion about disciplinary conventions and specific types of knowledge that may be quite abstract. In the case of architectural software, for instance, the Designer-User is a subject that requires a tool for translating imagined designs into graphic figures such as plans, diagrams, or 3D models. Thus, the software engineer must imagine a user that is quite intelligent, but also preoccupied with efficiency and clarity.

The Designer-User is what Lialina and Espenschied call a "naive user"—a subject concerned much more with the activity at hand than with the tool aiding them in enacting that task. "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."[8] This last statement brings us back to the design of this user. The perception from a software engineer's point of view might be that they are designing a system for a subject too busy or uninterested to dive into their technology. The solution, thus, would have to be simple enough to be readily learned, but also powerful enough to produce desired results—in our case, architecture. Significant examples of the design of this subject include Ivan Sutherland's dissertation *Sketchpad: A Man-Machine Graphical Communication System*, which effectively codified the rules for screen-based drafting[9] and Steve Jobs's famous interview where he described his ideal users as, "people [who] really don't have to understand how computers work."[10]



IMPORTANT - Read this License Agreement carefully before clicking on the "Agree" button. By clicking on the "Agree" button, you agree to be bound by the terms of the License Agreement.

End User License Agreement

By installing, copying, or otherwise using Rhinoceros, you agree to be bound by the terms of the following END USER LICENSE AGREEMENT (EULA).

If you do not agree to the terms of this EULA, do not install or use Rhinoceros; you may, however, return it to your place of purchase for a full refund.

This Robert McNeel & Associates EULA is a legal agreement between you (either an individual or a single entity) and Robert McNeel & Associates for Rhinoceros, which includes computer software and may include associated media, printed materials, and 'online' or electronic documentation.

Rhinoceros (Rhino) is owned by TLM, Inc. d.b.a. Robert McNeel & Associates. Rhinoceros is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. Rhinoceros is licensed, not sold.

Robert McNeel & Associates grants you the non-exclusive license to use the Software so long as the number of simultaneous users does not exceed the number of licenses you own. You may install the License in exactly one of the following ways. A: on stand-alone computers under your control. B: in the Zoo running on a computer under your control. You may grant access to the Zoo

CADInternational.com

Figure 4: Rhinoceros 3D End User License Agreement.

SUPERUSER (AS IMAGINED BY AN EMPLOYER)

The Superuser is a term borrowed—by Randy Deutsch—from the language of software management. In computing, ***superuser*** is the highest level of administrative access to the computer's system. Often interchangeable with "admin" or "root," the designation signals the ultimate authority. Deutsch reappropriated this term in his book, *Superusers: Design Technology Specialists and the Future of Practice*. In it, he describes the Superuser as, "a heroic character," one who is "able to achieve magic with the technology we have and create the tools we need."[11]

For Deutsch, this subject is an asset to any contemporary architecture firm. In contrast to Designer-Users, Superusers are deeply knowledgeable about computing and the latest technology. They are the intellectual scientist-users imagined by Vannevar Bush and Douglas Engelbart in the mid-20th Century, those knowledge workers fluent in the language of programming.[12] Today we might call this subject "whiz-kid," "hacker," or "maker." The Superuser is a direct product of the development of open-source tools, rapid growth of fabrication technologies, and widespread access to amateur knowledge (via websites like YouTube and Vimeo). In *Superusers*, Deutsch makes the claim that these individuals can not only learn tools quickly, but are also able to combine it with other tools and, most importantly, teach these tools to others. As a result of this malleability and inquisitive nature, they provide great value to any company. They are "today's version of the generalist architect."[13]

END-USER (AS IMAGINED BY A LAWYER)

End-User is a term given to one who enters into a contract with software developers in exchange for use of their products. An End User Licensing Agreement (EULA) is this legal contract. Typically, the agreement limits the user's rights—they often cannot, for example, modify the program or share their copy of the application with anyone else. Moreover, the contract often makes clear that the developer holds very little liability in case of product misuse, loss of data, or unexpected results.

The End-User is foremost an unpredictable legal entity. As Curtis Roth has noted, "End users are characterized by their likelihood to behave irrationally"[14] Therefore we can say that this subject exists primarily in a legal zone designed to protect the software manufacturer from the subject's precarity. By entering into these agreements, End-Users do not only receive limited rights to certain actions, but they also allow the products to interject themselves into the user's virtual life. EULAs can give software engineers the right to update their software at any point. They can manage their product remotely and collect and process data—in the form of images, location data, preferences, cookies, and error messages—that users produce so that they may improve their product.

In October, 2019, Adobe, the developer of Photoshop, Illustrator, etc, cancelled every Creative Cloud account that was registered in Venezuela. This was a direct result of sanctions that the United States had put on trade with Venezuela; no US company was allowed to continue to do business in the country. But because Creative Cloud software was subscription-based—users "rented" the software as opposed to owning it—this meant that no one could design or produce materials as they were accustomed to. Architects, Journalists, Graphic Designers all scrambled to work.

This episode illustrates perfectly the politics inherent in the End-User and software developer relationship. That any company can immediately cut-off access to their applications may come as a shock to users traditionally used to purchasing software in standalone versions, but subscription-based software is becoming increasingly the standard in design fields. While this shift does make software more accessible to those unable to pay the whole cost of a standalone application, it comes with the caveat of being a limited End-User. And there are currently no legislative rights for renters of software.

DATA-USER (AS IMAGINED BY AN ALGORITHM)

"The "user" [is] a contemporary mediated image of the self, one that is often reduced to narrow and utilitarian frames, but also open to a diverse variety of possible human and non-human agencies. The user position can both over-individuate that agent's sense of self and also radically multiply it. For example, data generated by Users and producing traces and shadows of their worldly transactions, initially creates a high-resolution portrait of a single user (for example as seen in the Quantified Self movement) but as overlapping external data streams are introduced, the coherency the user's subjectivity is dissolved by the overdetermination by external relations and forces. Any durable politics of the User must understand this dynamic of platform sovereignty."[15]

What Bratton describes in the above passage is the blurriness of the Data-User's subjectivity. The Data-User is often not an individual, but a collection of preferences, profiles, clicks, and other mediated behaviors recorded by algorithms. They are a product of algorithmic citizenship, a mirror image of an individual reflected back as personalized advertisements and online suggestions.

Most importantly, the Data-User is not designed by other individuals, as with the previous user-types. This subject is designed by algorithms and forecasting models running on cloud servers, meaning that the Data-User is much more than an estimation of who you are at the moment; it is "a profile of you to -come."[16] We have most likely all witnessed this in some form or another: books or people are recommended to us by Google or Facebook, our search history is played back to us in the form of targeted ads, and our autocorrect suggest words we use quite frequently. Most useful for advertising algorithms is not the self that we currently are, but the self we might become. The Data-User is you + your clicking behaviors + your phone + your smart-watch + your Firefox history + your dating profile + your credit score, "constructed as much through [y]our curated tastes as they are through obscure algorithms reprocessing [y]our rights of citizenship from a Nevada desert."[17]

TOWARD ABSURD USERS

If the above seems frighteningly dystopian, it's because we are not used to thinking of our ***selves*** as being designed by someone else, let alone an algorithm. But the goal here is not to scare, rather my intent is to shed light on the intricacies of users in order to find ways to move forward. We can no longer ignore the subjects created through hardware and software design, nor can we let information systems take charge of defining who we are as individuals. At the same time, contemporary life requires our participation in the constant exchange of data and interactions with interfaces. Can there be some middle ground? What would a series of design problems that use both the existing frameworks of user design—averaging, interpreting, simulating—and the absurdity of having billions of individuated preference profiles floating in the cloud, each waiting to predict your every move look like?

At the end of his chapter on the User Layer of *The Stack*, Bratton proposes a design prompt for three new "posthuman" users: the ***animal*** user, the ***AI*** user, and the ***machine*** user.[18] He states, "As more and more unlike figures come to occupy the ***User*** position, smashing up against one another and plugging into one another, they contort that position into different shapes, sizes, and durations."[19] Each of these nonhuman subjects operates much like our architectural users above, as stereotypes linked to particular emerging techno-social phenomena. The animal user is, for example, a companion species interface (like an animal cyborg); the AI user is Siri, your headless voice assistant with a personality; and the machine user is your driverless car. Looking at nonhuman users, Bratton states, allows us to reflect on what exactly constitutes a user. Is it intelligence? Is it a personality? Is it a specific behavior? More importantly, this allows us to push past the anthropocentrism that has yielded our current user confusion.

Bratton's design prompt suggests a kind of absurdist approach to user design. One that does not seek out averages and pseudo-objective truths, but instead operates similar to Jorge Luis Borges's Chinese Encyclopedia from "The Analytical Language of John Wilkins" in which taxonomies of animals are broken up into hyper-specific, almost nonsensical categories.[20]

As soon as we lose the "human-centered" of human-centered design, we can make room for other user-types, be they animal, robot, hybrid, or otherwise. But playing with the unexpected or unpredictable, as in an absurdist exercise, might be a good start and would ensure that designers still have agency in the complex world of software and algorithms.

ENDNOTES

1. Benjamin Bratton, "User Layer" in *The Stack, On Software and Sovereignty* (Cambridge: MIT Press, 2015) 251.
2. Olia Lialina and Dragan Espenschied. "Do You Believe in Users?" in *Mass Effect: Art and the Internet in the Twenty-First Century* (Cambridge: MIT Press, 2015)
3. For an expanded philosophical discussion on users see Bratton, "User Layer," in *The Stack*.
4. Ellen Lupton, *Beautiful Users: Designing for People* (New York: Princeton Architectural Press, 2014) 15.
5. "Human-centered design, simulation and ergonomics" in *Siemens Digital Industries Software*. Accessed November 11, 2019. https://www.plm.automation.siemens.com/global/en/products/tecnomatix/human-modeling-simulation.html
6. "ANSUR II," accessed November 11, 2019. https://www.openlab.psu.edu/ansur2/
7. Henry Dreyfuss, *Designing for People* (New York: Allworth Press, 1955) 14.
8. Olia Lialina and Dragan Espenschied. "Do You Believe in Users?" 10.
9. Ivan Sutherland, "Sketchpad, A Man-Machine Graphical Communication System" (PhD Dissertation, Department of Electrical Engineering, MIT, 1963)
10. David Sheff, "Playboy Interview: Steven Jobs," *Playboy*, February 1985.
11. Randy Deutsch, *Superusers: Design Technology Specialists and the Future of Practice* (New York: Routledge, 2019)
12. Lialina and Espenschied, "Do You Believe in Users?" 9.
13. Deutsch, *Superusers*, .
14. Curtis Roth, "Software Epigenetics and Architectures of Life" in *e-flux Architecture*. Accessed November 11, 2019. https://www.e-flux.com/architecture//248079/software-epigenetics-and-architectures-of-life/
15. Benjamin Bratton, "User Layer," 251.
16. Curtis Roth. "Drawing the Algorithmic Self." *Strelka Magazine*. Accessed November 11, 2019. https://strelkamag.com/en/article/curtis-roth-algorithmic-self
17. Ibid.
18. Bratton, "User Layer," 284.
19. Ibid., 285.
20. Jorge Luis Borges, "The Analytical Language of John Wilkins" in *Other Inquisitions (1937–1952)*, translated by Ruth L. C. Simms. (Austin, TX: University of Texas Press, 1975)