

BREJCHA, Jan. Ideologies in HCI: A Semiotic Perspective. In: MARCUS, A., ed. "Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience", Part I, HCII 2014, LNCS 8517, pp. 45–54. Springer, Switzerland, 2014. DOI: http://dx.doi.org/10.1007/978-3-319-07668-3_5. The original publication is available at www.springerlink.com.

Ideologies in HCI: A Semiotic Perspective

Jan Brejcha

Information Science and Librarianship, Charles University, Prague, Czech Republic

`jan@brejcha.name`

Abstract. The user-interface (UI) of interactive systems is the meeting point of people with interactive communication technology (ICT). As a human product, it forms a part of culture that determines us, often without our full awareness. The values and goals of the designers are implicitly encoded in the interface and the documentation but can be in conflict with the values of the user. This is when both the intentional and unintentional manipulation with the user starts because he or she is presented with inappropriate choices or even inappropriate goals. The aim of this article is to show how this manipulation works, in which regards it is unavoidable and how can we deal with it. Ideologies are a special means of manipulation and we can counter them by suitable education and analysis.

Keywords: User-interface, ideology, values, ethics, manipulation, persuasion, rhetorics, culture.

1 Introduction

The user-interface (UI) of interactive systems is the meeting point of people with interactive communication technology (ICT). As a human product, it forms a part of culture that determines us, often without our full awareness. The UI is constructed according to a set of values of the designer and other stakeholders in the production process. Their values and goals are implicitly encoded in the UI and the documentation, and can be in conflict with the values of the user. This means the UI directs the user interaction often according to the intent of the designer. This is when both the intentional and unintentional manipulation of the user starts because he or she is presented with choices, or even goals, that are incompatible for his or her intent. For the purpose of unmasking and decoding the inner workings of the UI, we have chosen a semiotics approach, with the emphasis on pragmatics, as defined by Charles Morris [23]. Semiotics is a study of semiosis, or sense-making, which has a syntactic (syntax), semantic (meaning) and pragmatic (purpose) dimension.

Pragmatics, "deals with the biotic aspects of semiosis, that is, with all the psychological, biological, and sociological phenomena which occur in the functioning of signs." [23][page 30]. This most complex dimension focuses on how we use or interpret vehicle/object relation, *i.e.*, what is the signs' purpose? The pragmatic dimension governs how signs are used or understood in their conventional and symbolic form.

Each and every UI is a result of diverse influences during the design process. Stakeholders in the process have their own goals and expectations that try to put into the final product. For example, the sales and marketing department could have strategic aims of short time-to-market, easy adoption of the product from the users and gimmicks to strengthen the brand and the product family. The programmers, on the other hand, might want to incorporate an advanced and clever technology, while the designers might want to create a simple and good-looking UI. All of these often conflicting values can have an input into the final product at the cost of the final user who expects the product to satisfy his or her needs and help achieve his or her goals. Often, such expectation falls short and the user is forced to become a "detective" trying to guess the motive of the designer in order to understand how to use the product in a sensible way. [8] In this light, the user should be aware as much as possible of the techniques used during the development process as well as the prevailing ideologies driving the UI production.

From our standpoint, the UI is an example of a complex language. Consequently, we can apply different UI language components such as: discrete elements, interaction sentences, narration, rhetorical tropes, and patterns. [3] Discrete elements are the smallest elements to have a meaning. The interaction sentence is a meaningful unit describing a task in the user's interaction. The narrative in UI is made both by the designers' meta-communication and the temporal and/or sequential aspects of perceiving UI elements. Rhetorical tropes are devices of persuasion and emphasis, such as metaphors. Patterns are typical configurations of UI language components in different settings.

By analyzing the individual statements exposed through the interaction with the software, we can follow an entire argumentation constructed with the help of the different UI elements. A simple way of doing this is transcribing the "interaction sentences" [3] that the user encounters while performing a certain task. The interaction sentences can be analyzed further in terms of what goals the designers have and what assumptions they have about his or her users. By exploring different parts of the system through the UI, we can extract the inherent values. We argue that when the UI follows the structure of natural language, it both behaves user-friendlier and conveys the designer's intent more effectively.

2 Ideology and HCI

For the purposes of this article, we understand ideology as:

"[A] logically coherent system of symbols which, within a more or less sophisticated conception of history, links the cognitive and evaluative perception of

one's social condition – especially its prospects for the future – to a program of collective action for the maintenance, alteration or transformation of society." [24][page 510]

What criteria should we then use to recognize ideologies and analyze them further? Again, according to Mullins, these components are: cognitive power, evaluative power, action-orientation and logical coherence. (*Ibid.*). By (2.1) cognitive power, he means the "cognition and retention of information" (*Ibid.*), when we identify and symbolize our recurrent experience. After having done this cognitive process we can simplify, order and abstract it for making choices between information, *e.g.* on different causal forces. The (2.2) evaluative power is then based on this understanding of information. Political ideology, "incorporates evaluations of what is conceived" and can anticipate "possible events and conditions." (*Ibid.*) The (2.3) action-orientation is based on the power of the ideology to "communicate conditions, evaluations, ideals, and purposes among members of groups (...) and thereby facilitates the mobilization and direction of energies and resources for common political undertakings." (*Ibid.*) Finally, the (2.4) logical coherence or consistency between various ideology components means, "the ideology must 'make sense' and not result in logical absurdities." (*Ibid.*)

As the word suggests, ideology is related to ideas. On this level, it is needed to focus on the relation between UI and image. As Mitchell put it,

"The concept of ideology is grounded, as the word suggests, in the notion of mental entities or 'ideas' that provide the materials for thought. Insofar as these ideas are understood as images - as pictorial, graphic signs imprinted or projected on the medium of consciousness - then ideology, the science of ideas, is really an iconology, a theory of imagery." [22][page 164]

Currently, in the context of ICT, ideology comes to us from a rather unexpected direction. As Galloway [16] points out, citing Althusser, ideology that was, "traditionally defined as an 'imaginary relationship to real conditions' (Althusser)" [16][page 953] has been superseded by simulation. He understands simulation as an "imaginary relationship to ideological conditions." In short, "ideology gets modeled in software." (*Ibid.*) Therefore, software serves as a prime example of current ideologies acting on us according to all the four criteria.

2.1 Cognitive Power

Software models ideology and is made visible through the way software works. This reflects the cognitive power of ideology. It does so by relating to the underlying hardware in a specific way:

"In a formal sense computers are understood as comprising software and hardware are ideology machines. They fulfill almost every formal definition of ideology we have (...). Software, or perhaps more precisely operating systems, offer us an imaginary relationship to our hardware: they do not represent transistors but rather desktops and recycling bins." [6][page 43]

Most importantly, as Chun continues, "software produces 'users'." [6][page 43] Software creates both a relation with hardware as well as with users. Hardware is

what the user encounters first, although the focus is then shifted to the software and the UI as a whole.

UI is regarded as an entrance into a simulated world, but UI is also forms a media layer between the “real” world and the user. "The doorway/window/threshold definition is so prevalent today that interfaces are often taken to be synonymous with media themselves." [16][page 936] An even more poignant definition relates the UI more tightly to the effect it has on the interacting users:

"The interface is this state of 'being on the boundary.' It is that moment where one significant material is understood as distinct from another significant material. In other words, an interface is not a thing; an interface is always an effect. It is always a process or a translation." [16][page 939]

The UI works thus not only on a semiotic level by differentiating symbols, but also on a psychological level when it creates relations and effects. For the UI to be effective and enjoyable, it is important to work “as a 'mirror' depicting the user's self-image, not only a 'window' looking into a world of content(...)." [21][page 53]

The differentiation work of the UI done between the user and users' self-image leads us to think about the UI in the terms of an active self-organizing entity. This notion is close to what Derrida [9] called "différance". Following Derrida's argumentation, the UI presents a different idea from the original one (or content) just by the way it is mediated. Thus, different media can go only as far as their structure permits. The medium of text can express other things than speech (*e.g.* Derrida's example of difference vs. *différance*, both of which are read the same); the medium of image can express other things than text, *etc.* The medium of the UI thus expresses its content differently.

The primacy of text for Derrida is something we can also observe very well in software. Software can go past the interacting subject, which is in contrast with the UI, which is bound in the subject/object relation [9] simply because it requires a user. And because the action is done through the UI, the UI privileges the content it presents. In this way, the UI not only tells us how to read a certain idea but can also pre-select for us which ideas we can possibly read. As Winograd and Flores state that: “Computers have an especially large scope, for they are machines that work with language. By using them, we join a discourse set up in the limits made by programmers.” [27][page 178]

Each UI presupposes a certain paradigm of use which is not always visualized. When built correctly, the UI lets us see only what has to be seen. The UI itself stands on a certain ideology. It defines relations which are to be made. The relations made by ideology are political in as much as they are social.

While the prevalent UI definition is connected with a gateway as a passage into another world, beyond the entrance this world is structured by another narrative. By analyzing the narrative, we can gain a better insight into the UI structure and the underlying ideology:

"In temporal terms, narrative is about what already happened while simulation is about what could happen. Because of its static essence, narrative has been used by our culture to make statements. (...) The potential of simulation is not as a conveyor of values, but as a way to explore the mechanics of dynamic systems." [14][page 86]

2.2 Evaluative Power

The user is presented with information “designed to program the spectators of techno-images to behave in a specific way and this in turn serves as a feedback to the programs calculating these techno-images.” [11] Here, techno-images are computer-generated images in Flusser's theory.

Therefore, for building new UIs, we ought to deconstruct the present ones and uncover their design/intent. Winograd and Flores also suggest this by stating that: “design is the interaction between understanding a creation... [We therefore] need to set up a theoretical framework not to watch how the devices operate, but what they cause.” [27][page 53] This is frequently the only way to understand new UIs in a situation when we do not have a suitable interpretation key - we do not know their code. It is, in a way, something like “reverse engineering” known from computer science.

2.3 Action Orientation

In order to use the UI, different *languages* are present in the form of action paradigms. “Action paradigms define a set of instructions that are available at any given moment. The paradigms offered by the system should match those the user needs so that she's not forced to perform an action she didn't intend.” [1][page 91] For example, take the interaction game for putting the computer to sleep in Microsoft® Windows® XP. Here the user has to first click on Start, then Shut Down, only then is he or she presented with the intended Sleep button. Thus, for putting the computer to sleep, we have to choose from UI language components that are in conflict with our intent. Even when something does not work as expected, we can gather interesting data out of it. When we interpret a connection between an UI sign and a proposed function, this mental connection is what forms our image of the system. “Systems work because they don't work. Non-functionality remains essential for functionality.” [16][page 931] Similar oppositions build the interaction space as “...the 'choices' operating systems offer limit the visible and the invisible, the imaginable and the unimaginable.” [6][page 43]

The action-orientation of ideology also works when the medialization (*i.e.* how the content is presented to the user) is not trustworthy. In such a case, however, the medialization works the other way round: it influences our intent according to what can be medialized. However, for a UI to be effective, it should be both trustworthy and familiar: “Designing for familiarity is crucial when trying to persuade people to behave in unfamiliar ways.” [26][page 99]

The user actuates the computer (or *apparatus*) to use it together with their technical imagination to create something but, paradoxically, one of the computer function is the user's intent. [12] This is so because the apparatus is predisposed only for some type of code and program cycles. As Bogost says: “Software establishes rules of execution, tasks and actions that can and cannot be performed.” [5][page 4] Therefore, for the designer's intention to be fulfilled, he or she can intend only what is doable.

Only using a specific apparatus for the chosen job can the designers' intent be fulfilled:

"The freedom of decision of pressing a button with one's fingertips turns out to be a programmed freedom. A choice of prescribed possibilities. I choose according to the regulations..." [13][page 93] Such freedom leads to the illusion of nearly unconfined freedom, however, our interactions are latently directed to a certain goal. This freedom leads us to take over the thinking of the designer.

The above-mentioned "programmed freedom" is closely connected with procedures as sequences of action. Again, with Bogost, "[p]rocedures are sometimes related to ideology; they can cloud our ability to see other ways of thinking (...)." [5][page 3] We can take the action-orientation element of ideology as a form of rhetoric.

Since Aristotle, rhetoric was used in different media to state arguments of the designer in order to make the audience believe in the presented reality. Persuasion as a technique has made its way into ICTs and has even been transformed into a tool. Fogg defines a persuasive technology tool as "an interactive product designed to change attitudes or behaviors or both by making a desired outcome easier to achieve." [15][page 32]

In the ICT environment, the persuasive tools are supported by the inner workings of software, as we have stated above. These workings, based on procedures, help to get predefined arguments to the users. Bogost calls it "procedural rhetoric". "Procedural rhetoric is a technique for making arguments with computational systems and for unpacking computational arguments others have created." [5][pages 2-3]

A specific characteristic of procedural rhetoric is that it does not build arguments using techno-images, but "through the authorship of rules of behavior, the construction of dynamic models." [5][page 29] Therefore, procedural rhetoric works in the space of medialization between design/intent and design/form. In such a manner, it is close to a "UI grammar" [3], where language plays the part of a rule system. In the system, the UI designer establishes grammar rules (syntax) for the combination of its elements. The manner in which UIs are built is governed by a set of rules given by the designer, *e.g.*, every UI produced can follow a different intrinsic UI grammar. The choice of elements is then subject to the pragmatics of the entire UI.

2.4 Logical Coherence

Mullins suggests that ideology should be coherent, *i.e.* syntagmatic rather than paradigmatic, since it needs to help create a seamless experience. From the perspective of internal connectedness, design fulfills the same function as art, technics and machines for they manipulate and try to master the original state of things, nature. [13] As Flusser [13][page 19] continues: "This is the design that is the basis of all culture: to deceive nature by means of technology, to replace what is natural with what is artificial and build a machine out of which there comes a god who is ourselves." What is important here is that the ideology perpetuates itself beyond the human reach.

Furthermore, "Programmers are not the important elements for the functioning of techno-images, but the structures of apparatuses they produce. Techno-images are imperativistic not because they are used by some ideologists to manipulate the society, but because they are a projection of such a pixel universe that pretends to present the world pixel by pixel." [11]

In the above quote, what is imperativistic is the constructed artificial world that forces us to take it for reality. What is imperialistic is the tendency of the producers (or even the producing automata of techno-images themselves) to colonize the semiotic space with signs (techno-images) referring to other techno-images and leaving out all the rest. Such tendency is supported by a number of ideologies embedded in the UI.

What are then the emerging ideologies present in the UI? Since its inception, the modernistic tradition of ideology orbits around five main concepts: emancipation (on a personal as well as social level), individuality (liberal ideology), time/space (fear of the stranger), work (with its emphasis on productivity) and community (nationalism, unity). We shift from "heavy" and "solid", hardware-focused modernity to a "light" and "liquid", software-based modernity. [2] In order to tackle this problem, a shift of analysis towards this liquid phase is needed. We argue that software - both on the personal (user) and social (society) level - should be regarded as a driving force, a catalyst, for a certain type of behavior. What happens when images are computer-generated, when they are "techno-images", as Flusser [12] coined them?

In the field of UI design, different instances of ideology are present. So far, one of the most prominent is the ideology of hypertext [4] - As Nielsen states, "[hypertext] makes individual users the masters of the content and lets them access and manipulate it in any way they please." [25] This user-empowering approach is contrasted by choice-obfuscation (*e.g.* when navigation links are not readily visible) or even user oppression (when user choice is limited or eliminated, *e.g.* in splash screens or ads). (*Ibid.*). Currently, the semantic space of UI ideology is somewhat centered around the terms "simple, fast, intuitive, social, minimal, choice, useful, fun", as a series of interviews with web designers suggest. [7]

Another important ideology is the ideology of ease. Dilger [10] presents the ideology of ease, which dissects users into the "computer illiterate" and "techies" and suggests that this "will ensure that the historical boundaries of gender, race and class are reproduced in computing practices for years to come." By ideologies, he means the "frameworks of thinking and calculation about the world - the 'ideas' that people use to figure out how the social world works, what their place is in it, and what they ought to do." (According to Dilger's reading of Hall, 1986). This is pretty much with Mullins' view, since the way the world works refers to cognitive and evaluative power, people's place in it and what they ought to do then refers to the action-orientation. Dilger states, that ease is gendered, which is to be seen in the connotation of an "easy" to use computer system as feminine. Ease has a different meaning in connection to work and leisure, during the former it has to be supported by the system, during the latter a certain difficulty could be desirable, *e.g.* in chess. At work, moreover, a task may not seem worthwhile if it does not seem easy. Pictures may furthermore seem easier to understand than text, which is supported by various media,

such as television or comics. The notion of speed is also connected to anything which would be labeled as easy including learning. Finally, the gain of ease is matched by a loss in choice, security, privacy, or health. (*Ibid.*)

Some of the HCI ideologies may even have a more pronounced impact on the user's behavior through the use of social cues and persuasion techniques. In the mobile context this has been carried out by Aaron Marcus and Associates, Inc., in a series of mobile applications. The applications may lead the users to act in a reduce their ecological footprint [20], reduce weight and improve dietary behavior [19], manage wealth after retirement [17], or share memories and family wisdom [18], among others.

3 Summary

In this paper, we have presented semiotics as an analytic method especially in its most complex dimension – pragmatics. Pragmatics stands in the design process at the beginning because it forms the strategy and purpose of the developed UI. In the sign, context pragmatics leads the meaning interpretation – what semantics will be assigned to which syntax elements. Not only is this a process of interpretation, but also the UI development strategy is subject to ideology in a large extent. Such ideology adapts its specific form in the UI. For the purposes of developing new UIs, but also for interacting with the UIs already in place, it is important to know the ways in which pragmatics, as an interpreting principle, is coded and mediated. We can then counter the ideologies by proper education and analysis.

A solution of how to address such a situation is therefore, on one hand, maximizing one's competence in terms of coding forms and medialization that has a big impact on the creation of UI. On the other hand, developing methods to analyze the influence of such UI on the society, the creation and modification of meaning and human relations that would be able to uncover the design behind every design. This article shows a possible approach based on UI language components analysis. This is in line also with Fogg's suggestion:

"One useful approach is to conduct a stakeholder analysis, to identify all those affected by a persuasive technology, and what each stakeholder in the technology stands to gain or lose. By conducting such an analysis, it is possible to identify ethical concerns in a systematic way." [15][page 233]

4 References

1. ANDERSEN, P. B.: A theory of computer semiotics. Cambridge: Cambridge University Press, 1997. ISBN 0521448689.
2. BAUMAN, Z. Liquid Modernity. Polity, 2000. ISBN 9780745624105.
3. BREJCHA, J, – MARCUS, A. Semiotics of interaction: Towards a UI Alphabet In: HCII 2013 Conference proceedings, 2013.
4. BUSH, V. "As We May Think." In The Atlantic Online. [accessed 8.10.2009]. <http://www.theatlantic.com/doc/194507/bush>. July, 1945. ISSN 1072-7825.

5. BOGOST, I. *Persuasive Games : The expressive power of videogames*. The MIT Press: Cambridge and London, 2007. ISBN 9780262514880.
6. CHUN, W. H. K. "On Software, or the Persistence of Visual Knowledge". *Grey Room*, pp. 26-51. 2005. ISSN 1526-3819.
7. CHANG, E. *Design 2.0: Minimalism, Transparency, and You*. [online]. [accessed 1.5.2009]. <http://www.emilychang.com/go/weblog/comments/design-20-minimalism-transparency-and-you/>.
8. DE SOUZA, C. S. *The semiotic engineering of human-computer interaction*. Cambridge (MA, USA): MIT, 2005. ISBN 0262042207.
9. DERRIDA, Jacques. *Writing and difference*. University of Chicago Press, 1978. ISBN 9780226143293.
10. DILGER, B. "The Ideology of Ease." In *Journal of Electronic Publishing*. 6 (September 2000). [accessed 3.4.2009]. <http://quod.lib.umich.edu/cgi/t/text/text-idx?c=jep;view=text;rgn=main;idno=3336451.0006.104>. ISSN 1080-2711.
11. FLUSSER, V. *Lob der Oberflächlichkeit*, Bollmann Verlag, Mannheim 1995
12. FLUSSER, V. *Into the Universe of Technical Images*, University of Minnesota Press, 2011. ISBN 978-0-8166-7020-8.
13. FLUSSER, V. *The Shape of Things: A Philosophy of Design*. Reaktion Books, 1999. ISBN 1861890559.
14. FRASCA, G. "Videogames of the oppressed: Critical thinking, education, tolerance, and other trivial issues." In *First Person: new media as story, performance, and game*. Edited by N. Wardrip-Fruin, pp. 85-94. MIT: 2004. ISBN 0262731754.
15. FOGG, B. J. *Persuasive Technology: Using computers to change what we think and do*. San Francisco, Morgan Kaufman Publishers: 2003. ISBN 9781558606432.
16. GALLOWAY, A. R. "The Unworkable Interface." In *New Literary History*, 39: 931-955. 2009. ISSN 0028-6087.
17. MARCUS, A. *The Money Machine: Helping Baby Boomers Retire*. *User Experience (2012a)* vol. 11 (2) pp. 24-27
18. MARCUS, A. *The Story Machine: a Mobile App to Change Family Story-Sharing Behavior*. *Proceedings of CHI 2012 (2012b)* pp. 1-4
19. MARCUS, A. *Health machine*. *Information Design Journal (2011)* vol. 19 (1) pp. 69-89
20. MARCUS, A. and JEAN, J. *Going green at home: the green machine*. *Information Design Journal (2009)* vol. 17 (3) pp. 235-245
21. MARCUS, A. "Metaphor design in user interfaces". In *SIGDOC Asterisk Journal of Computer Documentation*, vol. 22 (2). 1998. ISSN 0731-1001.
22. MITCHELL, W. J. T. *Iconology : Image, Text, Ideology*. University Of Chicago Press, July 1987. ISBN 0226532291.
23. MORRIS, C. W. *Foundations of the Theory of Signs*. Chicago: University of Chicago Press, 1970. ISBN 0226532321.
24. MULLINS, W. A. "On the Concept of Ideology in Political Science." *The American Political Science Review* 66 (1972): 498-510. ISSN 0003-0554.
25. NIELSEN, J. "Mastery, Mystery, and Misery: The Ideologies of Web Design." *UseIt.com* [online]. 2004. [accessed 3.4.2009]. <http://www.useit.com/alertbox/20040830.html>. ISSN 1548-5552.
26. WAI, C. – MORTENSEN, P. "Persuasive technologies should be boring." In *PERSUASIVE 2007*, LNCS 4744, edited by Y. de Kort et al., pp. 96-99. Berlin Heidelberg: Springer-Verlag, 2007. ISBN 978-3-540-77005-3.
27. WINOGRAD, T. – FLORES, F. *Understanding computers and cognition: A new foundation for design*. Boston: Addison-Wesley, 1987. ISBN 0201112973