

BREJCHA, Jan and MARCUS, Aaron. Semiotics of Interaction: Towards a UI Alphabet. In: KUROSU, M., ed. *Human-Computer Interaction, Part I*, HCII 2013, LNCS 8004, pp. 13--21. Springer, Heidelberg, 2013. DOI: [http://dx.doi.org/10.1007/978-3-642-39232-0\\_2](http://dx.doi.org/10.1007/978-3-642-39232-0_2). The original publication is available at [www.springerlink.com](http://www.springerlink.com).

## Semiotics of Interaction: Towards a UI Alphabet

Jan Brejcha<sup>1</sup>, Aaron Marcus<sup>2</sup>

<sup>1</sup> Information Science and Librarianship, Charles University, Prague, Czech Republic

`jan@brejcha.name`

<sup>2</sup> Aaron Marcus and Associates, Inc., Berkeley (CA), U.S.A.

`aaron.marcus@amanda.com`

**Abstract.** In our thinking and acting, natural language plays a central part. This language defines a structure even before we form something, and it can be regarded as the architecture of design. To be able to grasp the expression of these structures in HCI, we chose the perspective of linguistics and semiotics. The semiotics perspective in the context of HCI is increasingly popular in presenting a different approach to UX. In our paper, we take this perspective to build a set of semiotic heuristics which we then used to evaluate a complex UI example. We present a semiotic evaluation method and report the results of our in-depth investigation.

Keywords: Analysis and design methods, Evaluation methods and techniques, Interaction design, Qualitative and Quantitative Measurement and Evaluation, Semiotics

### 1 Introduction

In our thinking and acting, natural language plays a central part. This language defines a structure even before we form something and can be regarded as the architecture of design. Our consciousness is the result of language informing design. Grammar allows for many combinations of objects and actions, but ideology establishes the privileged connection of the two of what is correct and possible. In order to set forth the privileged connection, ideology employs different forms of persuasion. To grasp the expression of these structures in HCI, we chose the perspective of linguistics and semiotics.

By semiotics we mean a theory of signs. We combine the Anglo-American semiotics (semiotics) perspective with the French semiology (“sémologie”) approach. [3]

According to Peirce, a sign is "something that stands for someone or something in some respect or capacity." [4][page 99] Four dimensions form the sign: lexical [5], syntactics, semantics, and pragmatics. [10]

The semiotics perspective in the context of HCI is increasingly popular in presenting a different approach to UX planning, researching, analyzing, designing, implementing, evaluating, documenting, training, and maintaining. [7] The classical linguistic and semiotic foundations of HCI were previously set down by, *e.g.*, Nadin [11], Andersen [2], and De Souza [13]. In our paper, we take their contribution to build a set of semiotic heuristics that we then used to evaluate a complex UI example. We present a semiotic evaluation method and report the results of our in-depth investigation.

## 2 Semiotic Foundation

Our view is based on the assumption that HCI takes place between different actors (users, systems, designers) in a setting or paradigm. The semiotics of interaction is closely related to language as a system of signs. The semiotics of interaction is by definition time-based and the same holds true for language. Because of this, we may find some interesting parallels. The UI designer establishes grammar rules (syntax) for the combination of its elements. UIs are built from different components (metaphors, mental models, navigation, interaction, appearance) [8]. 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 language grammar. The choice of elements is then subject to the goal (pragmatics) of the entire UI. Therefore, we should structure the UI language according to the actors and audience we want to address.

Interaction is subject to the languages present in any UI: a language both of display and actions. [14] We define the former as a visible language which is a systematic language of expression conveying specific information that can be translated from one kind of language to another. The latter is based on user input which makes it an interaction language. Based on our decision to act (or refrain from acting) on an object, we start an interaction. The visible and interaction language is expressed through UI components.

Looking at the UI components from the language perspective, we can structure them organically to create a UI grammar. UI grammar is composed of basic elements: interaction sentence, interaction games, rhetorical tropes, interaction phases, and patterns. The grammar elements concern both the noun and verb phrase of a sentence. Discrete elements are the smallest elements to have a meaning. The interaction sentence is a meaningful unit describing a task in a user's interaction. A set of interaction sentences with the same goal form an interaction game. The narrative in UI is made both by the designer's meta-communication and the temporal aspect of perceiving UI elements. Rhetorical tropes are devices of persuasion and emphasis, often presented as metaphors. Patterns are typical configurations of UI language components in different settings. From the defined semiotic and UI language principles, we extracted a set of heuristics which could be used as an UI glossary both for evaluation and design.

## 2.1 Semiotic Heuristics

**Actors, audience and paradigm.** Defining who the UI users are, who the intended audience of the UI is and what the leading interaction paradigm is. Communication from the UI to the actors should be concise, clear, and unambiguous. The audience can be revealed, *e.g.*, by contrasting two or more UIs with in similar semantic spaces (*e.g.*, functions).

**Symbols.** Different kinds of symbols connote different semantic spaces, cultural backgrounds and address different audiences. The symbols should be intelligible for the audience and should not carry a pejorative or contrary meaning. Symbols should be chosen to support the rhetorical tropes and can be formed by signs or their attributes.

**Syntax.** Signs should be used in any given context only once and should not be in conflict with its context. Similar signs should be placed in similar places. The signs should be divisible into identifiable elements and allow for building meaningful chains. The signs should be internally (within a UI) and externally consistent (across multiple UIs). The system processes are revealed by UI language components.

**Rhetorical tropes.** The most common rhetorical tropes in the UI are devices of substitution: metaphor, metonymy, prosopopoeia, and synecdoche. The rhetorical tropes used should be both intelligible for the audience and minimal. The general metaphor of the UI should help users build correct expectations of future interaction through consistent mental models.

**Interaction phases.** The interaction should form meaningful temporal units. The beginning of the interaction should be consistent with both middle and end. All the parts of the interaction should follow user's expectations and should pertain clearly to the current interaction game. The user should not be forced to perform a different action than intended. The signs present on a UI should lead the user in a sequence towards the goal of the interaction game through a controlled narration.

**Patterns.** UI language components form different kinds of patterns. The number of the expected interaction sentences should be as low as possible, possibly not exceeding the  $7\pm 2$  limit [9]. The interaction sentence should have as little number of words as possible. The results of similar functions should be returned in a similar time frame.

### 3 Research Method

We developed a semiotic analysis method that takes as input the interaction sentence transcript with figures of the UI. Because it is an evaluation method carried by experts, we wanted to compare it with a well-known method to see whether the results would differ and how. To compare the methods of expert evaluation we chose heuristic evaluation (HE) and semiotic analysis (SA). For HE heuristics, we chose those used by Aaron Marcus and Associates, Inc. [7]. Our criteria for the methods were: fast and easy to do, results accessible to non-experts and comparable to previous data. The goal was to validate the SA against a non-semiotic method.

As input for this comparison, we chose a UI corpus [3] consisting of similar portions of two complex graphic design applications: Adobe® Photoshop® and the GNU Image Manipulation Program, or GIMP. Traditionally, these methods have been employed for goal-oriented work applications. However, they can be also applied to a range of different applications, including entertainment applications, where the goal is not always clearly defined, both in a static or mobile setting. In order to better compare and analyze the corpus, we aligned the semantic spaces (*e.g.*, the features, tools, functions) of the application by their title from the products' marketing publications and user manuals. [12] The following list of actions constitute the semantics of the selected UIs: Barrel distortion, Clone objects in perspective, Customize the UI, Eliminate an object, Reduce red-eye.

### 4 Results

We present our analysis results from the SA and HE of both of the compared UIs following the extracted UI corpus.

#### 4.1 UI Annotation

We annotated the UIs using a transcript of interaction sentences from actions. As an example of the action we chose, *Clone objects in perspective*. Table 1 contains the interaction sentences that were extracted from both applications.

**Table 1. Interaction sentence comparison of Adobe Photoshop and the GIMP.**

	Adobe Photoshop	GIMP
	Open the picture to adjust.	Open the picture to adjust.
	Find the proper function in the menu or tool palette.	Find the proper function in the menu or tool palette.
	(a) The sub-task involved was to look at the toolbox for a button resembling the intended action. Nothing like that was found.	(a) The sub-task involved was to look through the menu items (especially in what seemed as most related: Image -> Adjustments, Filters, and Tools -> Transform Tools) for a relative command (it was found under Tools ->

		Paint Tools -> Perspective Clone).
	<p>(b) Alternatively to look through the menu items (especially in what seemed as most related: Image -&gt; Adjustments, and Filter) for a relative command (it was found under Filter -&gt; Vanishing point...).</p> <p>(A window called "Vanishing Point" appears. The window sports a live preview, "Create Plane Tool", and "Clone Tool", among others.)</p>	<p>(b) Alternatively to look at the toolbox for a button resembling the intended action. It was found as Perspective Clone.</p> <p>(Boxes on each corner of the image appeared, the pointer changed to cross-hair with the tool attribute and the toolbox expanded to show the "Modify Perspective" selected and the "Perspective Clone" radio button.)</p>
	Click the four corners according to the information line provided ("Click the four corners of a perspective plane or object in the image to create an editing plane. Tear off perpendicular planes from the stretch nodes of existing...").	Drag the four corner boxes to define the perspective plane to clone.
	Select the "Clone tool".	Click on the "Perspective Clone" radio button to change the tool.
	Option-click in the plane to set the source ("Opt+click in a plane, to set a source point for the clone. Once the source point is set, click+drag to paint or clone. Shift+click to extend the stroke to last click.").	Control-click the source in the defined plane.
	Click-drag (to paint) several times to clone in the perspective. See the proposed results in the preview.	Click-drag (to paint) several times to clone in the perspective.
	Click "OK" to apply the changes.	Save changes to the file.
	Save changes to the file.	

## 4.2 Semiotic Analysis

### Adobe Photoshop SA Analysis.

*Actors, audience and paradigm.* All of the objects involved in the interaction pertain to the leading paradigm of "Window, icon, menu, pointing device" (WIMP). The paradigm is constituted by the menu bar, tool bars, main window containing the image, dialog windows, icons and pointer. The paradigm is bound to the GUI metaphor. Adobe Photoshop is meant for professionals. This distinction of audience is manifested implicitly by the channel of distribution (commercial software) and explicitly in the marketing documentation (Adobe Photoshop's slogan reads: "The professional standard in desktop digital imaging" [1]). The menu paradigm is constructed by combining noun-verb or verb-noun items which seems deliberate (only one model should

be chosen.) A more specific audience for this function is photographers and advertising designers.

*Symbols.* The users are addressed by symbols pertaining to the user domain. In this case, the application icon and splash screen of Adobe Photoshop features a colorful feather. The connotations are elegance, simplicity and naturalness which one would expect from a professional tool. What might break the expectation, however, is the historical usage of the image that symbolizes a writing pen. The other screens (and toolbars) are very compact and grey. The menus are only text-based, whereas the toolbar has only icons (with a textual label). The icons in the toolbar are related to their object in different ways but are connected to the prevailing metaphor and follow the application genre conventions.

*Syntax.* The system processes are constituted by UI language components, as described earlier. In the interaction transcript, we can find all the elements mentioned. There are basic lexemes (“click”, “option-click”), interaction sentences (“Open the picture to adjust.”), rhetorical tropes (*e.g.*, metaphors, such as “Vanishing Point”, or “Clone Tool”), interaction games (these are the complete functions enabling us to accomplish our goal, *e.g.* “Clone an object in perspective”). The designer’s narration element is found in the tool-tip help reinforcing the icon meaning, in the status bar of the window or a dialog window for which help is given by instructions regarding use of the tool and in other dialog windows which presents the user with different choices. Finally, in the Help menu, the text comprehensively describes the program functions. In the “Vanishing Point” window, the designer’s narration gives detailed instructions for all the steps involved.

*Rhetorical tropes.* The most prominent of the rhetorical tropes in this context is the metaphor. The program metaphor builds upon the concepts of a painter’s canvas or photographer’s studio. The product tries to transfer the environment into the present paradigm. For that reason, the image is placed on a “canvas”, the pointer changes to different “brushes”, the user can further apply different optical “filters”, or use a choice of retouching “tools”. By applying this approach, a number of inconsistencies emerge which force users to alter or update their interpretation of the metaphor. The canvas, for example, is in fact infinite and can be resized in different ways at any time. The picture “lying” on the canvas may consist of multiple layers. Almost any tool can be customized using the “brush” metaphor: one can modify the thickness, shape, or profile of the brush. A filter can be used afterwards, applied as a part of retouching. More fundamentally, time can also be manipulated through the “undo” function that cycles back through the history of actions.

*Interaction phases.* On the level of interaction sentence, the interaction changes to reflect the constant evaluation of results on the user’s part. The interaction sentence is then modified or repeated accordingly. Considering the example from the transcripts, the action is modified after the system’s feedback (when clicking on the plane to clone with the clone tool, the user is instructed to option-click on the source plane first), the action is repeated (drag the brush several times to paint the object in the new perspective), or the action is needed only once (when applying the changes by pressing the “OK” button). The middle of the interaction game differs from the beginning

and end because a new window is shown keeping the user accordingly away from the picture he or she opened.

### **GIMP SA Analysis.**

*Actors, audience and paradigm.* All of the objects involved in the interaction pertain to the leading paradigm of “Window, icon, menu, pointing device“ (WIMP). The paradigm is constituted by the menu bar, tool bars, main window containing the image, dialog windows, icons and a pointer. The paradigm is bound to the GUI metaphor. GIMP is intended for amateur/semi-professionals and programmers. This distinction is manifested implicitly by the channel of distribution (open-source) and explicitly in the marketing documentation (in GIMP it is by stating, that “[i]n the free software world, there is generally no distinction between users and developers.” [6]. The menu paradigm is constructed by combining noun-verb or verb-noun items which seems deliberate (only one model should be chosen). GIMP shares the same user group as Adobe Photoshop (photographers and advertising designers).

*Symbols.* The users are addressed by symbols related to the user domain. In this case, Gimp’s icon of a stylized dog head connotes playfulness, fun and also ease of use. The icon is not used on the splash screen, however, in favor of a planet picture. The toolbars and other screens show larger and more colorful button icons and larger dialog windows which are easily reached by the pointer. The icon symbols used in the menus, *e.g.*, in the “Tools” menu, makes no distinction between nouns (*e.g.*, Pencil, Eraser, Text) and verbs (*e.g.*, Zoom, Measure, Heal) which could be helpful. Also, the symbols are created by different methods (*e.g.*, the Pencil tool has an iconic representation of a pen but the Zoom tool icon is created by metonymy with its action and uses a zooming lens; other are connected only loosely, as in the case of Swap Colors).

*Syntax.* The system processes are constituted by the same UI language components analyzed above for Adobe Photoshop. In GIMP, there is only a difference in the tool metaphor used (“Perspective Clone Tool”).

*Rhetorical tropes.* Perhaps the most prominent of the rhetorical tropes in this context is the metaphor. As is the case of syntax, the same set of metaphors is shared with Adobe Photoshop.

*Interaction phases.* The interaction phases are similar to those mentioned above in the Adobe Photoshop analysis. Also, the interaction sentence level is similar. However, the middle phase (where the user works on the picture) seems to be more consistent with beginning and end. This is because the user keeps working in the image window and is not distracted by other windows or palettes.

## **4.3 Heuristic Analysis**

### **Adobe Photoshop HE Analysis.**

*Direct manipulation/see and point; Error prevention.* Although the user can use the tool directly on the image, they are reminded every time to select a source region first. Instead of forcing the user to go “backwards“, the program should allow the user

to select the region afterwards. Such change in the perceived interaction timeline also violates the principle of Error prevention.

*Modelessness.* By selecting the vanishing point function, the user is presented with a new window (named “Vanishing Point”) containing the image to manipulate and a reduced set of controls (buttons, check-boxes, and drop-down menus). After the adjustments, the user has to click “OK” to transfer the changes to the image in the main window underneath. A better solution seems to be using standard controls and not introducing a different working environment. By doing so, we would also eliminate the extra step of applying the changes.

*Recognition rather than recall.* All of the needed actions are visible and the system provides inline help. However, the toolbar on the top-left does not show which tools are necessary for the operation and in which sequence they should be applied.

*Visible interfaces/WYSIWYG.* The vanishing point function was not present on the toolbar and was only accessible through the menu bar. Since it is one of the advertised features, it should be as readily accessible as possible.

### **GIMP HE Analysis.**

*Direct manipulation/see and point; Error prevention.* Although the user can use the tool directly on the image, he or she is reminded every time to select a source region first. Instead of forcing the user to go “backwards“, the program should allow the user to select the region afterwards. Such change in the perceived interaction timeline also violates the principle of Error prevention.

*Recognition rather than recall.* All of the needed actions are visible and the system provides inline help. However, the toolbar on the top-left does not show which tools are necessary for the operation and in which sequence they should be applied (“Modify Perspective” or “Perspective Clone?”).

*Match between system and real world.* The Perspective Clone tool is located under Paint Tools and thus supports the metaphor of painting on the picture. However, in the virtual environment this could be problematic as the clone tool is connected with image transformation and/or filtering.

## **5 Discussion**

By comparing the output from the HE and SA analysis, HE proved to be more concise. However, of the 16 heuristics used, only a small number could be applied on each occasion. The application of the 6 elements of SA tended to be more verbose, but, on the other hand, the elements could be applied every time. Whereas SA could seem repetitive in some instances, it provided a solid context of analysis. Both the methods (HE and SA) could be used not only on the interaction sentence level but also as for a general appreciation of the entire UI. During the general analysis only portions of the UI are selected and suggestions made to other similar parts of the UI.

In summary, our study demonstrated the depth of investigation and breadth of insight that SA can achieve in HCI and how this could enhance the current practice. Both methods could be merged to provide a best-of-both solution.



## 6 Acknowledgements

The author recognizes the assistance of the AM+A library and document archive for providing information and examples for this text.

## 7 References

1. Adobe, Inc. (2005). What's New in Adobe® Photoshop® CS2. [last accessed 18 Feb 2013]. Available from: <http://www.adobe.com/aboutadobe/pressroom/pressmaterials/creativ esuite2/pdfs/ps/PSCS2-WN.pdf>.
2. Andersen, P. B. (1997). A theory of computer semiotics: semiotic approaches to construction and assessment of computer systems. Cambridge University Press.
3. Barthes, R. (1977, April). Elements of Semiology. Hill and Wang.
4. Buchler, J. (1955). Philosophical writings of Peirce. Dover.
5. Eco, U. (1979, February). A Theory of Semiotics. Advances in Semiotics. Indiana University Press.
6. The GIMP Team. (2009). GIMP - Development. [last accessed 19 Sep 2009]. Available from <http://gimp.org/develop/>.
7. Marcus, A., et al. (2003). Heuristic Evaluation Guidelines. Aaron Marcus and Associates, Inc. Internal document.
8. Marcus, A. (2002). "Dare we define user-interface design?" Interactions 9 (5), 19-24.
9. Miller, A. G. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review, 101(2), 343-352.
10. Morris, C. W. (1970). Foundations of the Theory of Signs. The University of Chicago Press.
11. Nadin, M. (1988). "Interface design: A semiotic paradigm". Semiotica 69 (3), 269-302.
12. De Souza, C. S. et al. (2006). "The semiotic inspection method". In: Proceedings of VII Brazilian symposium on Human Factors in Computing Systems.
13. De Souza, C. S. (2005). The Semiotic Engineering of Human-Computer Interaction (Acting with Technology). The MIT Press.
14. Sutherland. (1966). On-line graphical specification of computer procedures. <http://oai.dtic.mil>. PhD Thesis.