INTERFACE DECONSTRUCTION-RECONSTRUCTION USING INTERTEXTUAL SEMANTICS: RATIONALE AND EXAMPLE

Élias Rizkallah,
Département de sociologie, Université du Québec à Montréal
C. P. 8888, succursale Centre-ville, Montréal, Québec, Canada H3C 3P8
rizkallah.elias@uqam.ca

Yves Marcoux
GRDS – EBSI – Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal, Québec, Canada H3C 3J7
yves.marcoux@umontreal.ca

ABSTRACT

Intertextual semantics (IS; Marcoux & Rizkallah, 2009) is one of several semiotic approaches for the design of *information objects*. It uses *natural language* – shared among designer, developer, and users – as an *explicitation tool* for developing an agreed upon, shared meaning for an artifact. So far, it has been applied constructively to the design of various data structures (documents, databases, etc.). It is also intended to be applicable to interface design, as well as to the *deconstruction* of artifacts (*a posteriori* analysis), but has never actually been used in such ways. This paper presents a first attempt at using IS for the design and analysis of interfaces. We discuss the rationale and basic principles for doing so, then exemplify the process by applying it to (part of) a search interface.

KEYWORDS

Interaction design, Intertextual semantics, Semiotic approaches.

1. INTERTEXTUAL SEMANTICS AND RELATED WORKS

Intertextual semantics (IS) is a general approach to the design of information-bearing objects (IBO), i.e., any object (concrete or abstract) with which humans can interact to draw information (Marcoux & Rizkallah 2009). These interactions take place during sequences of exchanges (dialogues or conversations). IS is indeed a *semantics* in that it seeks to give meaning to objects developed or in development and it is *intertextual* in that the meaning is expressed by a network of interrelated texts. IS aims to guide the design of IBOs by showing how their different characteristics can influence the way humans interact with them.

IS can be situated in the general framework of semiotic approaches to interface and interaction design, more specifically the *Semiotic engineering* of de Souza (2005), with which it shares the view that a semiotic process (communication) takes place between user and designer at interaction time. The designer is thus not limited to *trying to capture* the needs and expectations of the user, but can rely on (and must thus plan) communication with the user *at interaction time*. Despite the affinities between IS and de Souza's approach, the former differs on an important point: IS considers natural language (NL) as the main semiotic system for human communication. Several reasons can be invoked for this: a) NL can describe all artificial languages (Wierzbicka 1992); b) it is the most widely shared in any target community, thus representing a minimum requirement often easily reachable; c) it includes its own rules of use (grammar, sequentiality, linearity, etc.). In contrast to IS and *Semiotic engineering*, the trend in user studies and corresponding usability inspection methods (Nielsen,1995, Hollingsed & Novick, 2007), such as *Heuristic evaluation* (Nielsen 1994), *Cognitive walkthroughs* (Wharton et al. 1994), or more broadly *Usability testing* (Rubin & Chisnell 2008), is to rely on the skill of the designer at capturing users' needs and behavior and on user studies. Note that an IS approach does not preclude or alleviate the need for usability testing; it is intended to guide the choices of the designer

in the creative part of his or her work, and provide a different angle for evaluating artifacts by focusing on the semiotic system that lies and that is shared between the designer and the user.

Although not a development methodology, IS is intended to assume various functions at different times in the life cycle of IBOs. In particular, it can be used *predictively*, to guide choices at the time of designing objects: this is what we call the *constructive* function. So far, IS has been applied in this way to the design of various data structures (documents, databases, etc.; see references in Marcoux & Rizkallah 2009). But IS is also intended to be used *retrospectively*, to help understand and evaluate objects that are already deployed: this is what we call the *deconstructive* function. This paper presents a first attempt at using IS for the design and analysis of interfaces. We discuss the rationale and basic principles for doing so, then exemplify the process by applying it to (part of) a search interface.

2. INTERFACE DECONSTRUCTION: WHY AND HOW

In the IS approach, any artifact is given an "intended interpretation", and the systems designers are responsible for specifying it in *natural language* (prose). In line with this principle, the basis of an IS analysis of an interface will be its *intended interpretation*. This corresponds to the complete message (in natural language) the designers think must be conveyed to the user in order for him/her to understand and use the various interface elements properly. In other words, a message sufficient for the user to fully understand what he/she can or is expected to do with the interface.

In many cases, the intended interpretation of the interface can be derived (or taken directly) from design documents (e.g. specs, needs analyses, etc.), whereas in other cases, like the example illustrated below, design documents are unavailable. IS analysis of the interface is still possible, but the intended interpretation has to be discovered by the analyst either by trial and error (if a working system is accessible), or by asking the designers.

Once the intended interpretation of the interface – i.e., its explicit semantics expressed in natural language (NL) – is established, we propose to divide its various elements (or components) into five groups:

1. (E) those which must be inferable from Explicit elements in the interface;
2. (B) those which can be considered prerequisite Background knowledge on the part of the user;
3. (A) those Acquired by the user through previous use of the same interface;
4. (P) those conveyed to the user by immediately Preceding interactions;
5. (D) those which need only be presented to the user on Demand (through hyperlinks, tool-tips, or other transient mechanisms).

The main purpose of identifying these groups is to determine group E: what must be explicitly conveyed by the interface. Care must be taken to make sure the groups are realistic for the target community. For example: group B must take into account the expected techno-cultural background and other characteristics of the target community (and the feasibility of *ad hoc* training); group A must not be so large as to discourage first-time users; group P must not be so large as to put an unrealistic cognitive load on the user. Note that full "self-containment" of the interface would imply that B contained only very basic communication skills (e.g., the mastery of a given natural language), and that E be the only other non-empty group, a situation which, for practical purposes, is in general neither possible nor desirable.

The rest of the deconstruction process consists in verifying that elements in E are indeed conveyed by the interface. For this purpose, we propose to apply a "quasi-linguistic reading" (QLR) to the interface, which is thus translated into "quasi-natural language" (QNL). Once this is done, this version of the interface is compared with the elements (in NL) of the intended interpretation assumed to be in E. If the two are semantically equivalent, everything should be fine. Otherwise, it indicates a discrepancy between what the interface is actually telling the user and what the system designers' intended it to convey explicitly (group E). The deconstruction is thus a kind of "proof by inverse operation," like one performed after an arithmetic division to verify the correctness of the result[1].

The ideas of QLR and of translating an interface into QNL may appear shaky to the reader, and the actual process of translating interface elements into QNL (as in Section 3.2) may seem to rely on countless unproven claims about users. However, it must be realized that the goal is not to prove or disprove anything about the

[1] Multiplying the quotient by the divider, and checking that the product is equal to the dividend.

users, but to design a usable system or interface. We are not postulating that QLR is indeed how users *actually* interpret interface nor that QLR applied by different users would necessarily lead to similar results. All we are postulating is that using QLR as a *heuristics* in the design process can help designers take "good" decisions, leading to "usable" products.

In a way, IS suggests regarding the interface design process as a *writing* process. Deconstruction-reconstruction is thus expected to be performed by the interface designer in very tight cycles; much like an author iteratively refines a text by rereading it and making minute adjustments in tight alternations. Frederick P. Brooks, in his classical book "The Mythical Man-Month" wrote: "Only when one writes do the gaps appear and the inconsistencies protrude. The act of writing turns out to require hundreds of mini-decisions, and it is the existence of these that distinguishes clear, exact policies from fuzzy ones." (1975, p. 111). The QLR approach serves as a heuristics to guide the designer in taking these "hundreds of mini-decisions" about the interface in development, without having to consult users for each and every one of them.

We do in fact believe that, once the target community of users is defined (in particular, its techno-cultural background), the QLR approach is *semantically robust*. Given an interface or interface elements to "translate" into QNL, different translators are almost certain to come up with different exact wordings. However, unless the interface is ambiguous, we think the different translations obtained will have quite similar semantics, at least regarding usability issues. Even if the interface is ambiguous, we believe the QLR approach can still be useful, because it is likely to make the ambiguity obvious. That is why we believe the QLR approach is suitable as a heuristics for the "hundreds of mini-decisions" an interface designer has to take. Of course, the QLR approach does not alleviate the need for user testing, but we claim it can lead to a more rapid convergence towards a satisfactory interface, by suggesting changes to be made even before the interface is tested by users.

3. A DECONSTRUCTION-RECONSTRUCTION CYCLE

In this section, we go through a cycle of deconstructing and reconstructing a part of an interface using IS. We use the search interface (shown on Figure 1 with an active tooltip window) of the digital library of the International Observatory on Financial Services Cooperatives at HEC Montréal (www.hec.ca/observatoire). We will however treat only the section pertaining to the limitation by document type (i.e., the blocks "a" and "b" identified in Figure 1).

Figure 1 - Complete basic search interface and the blocks treated in the paper

3.1 Intended interpretation and grouping

Based on our knowledge of how the criterion is implemented, the actual semantics of the criterion is as follows: *if and only if* at least one of the document types listed is ticked, then limit the result of the query to only documents of the ticked types. The intended interpretation can thus take two quite different forms:

1. (When no checkbox is ticked) I do not want you to exclude any document on the basis of the document type.
2. (When at least one checkbox is ticked) I want you to keep only the documents that are of one of the following types: (list of checked types).

The semantics of the criterion is not complete until we add that all documents are of exactly one type and that the list of types presented to the user is exhaustive (thus, a grouping of the documents by type constitutes a *partition* of the collection in the mathematical sense). This could be accomplished by adding the following sentences to either sentence 1 or 2 above[2]:

3. Each document in the collection is of exactly one type.
4. There are no other document types than these ones: (list of document types).

We refer to these sentences as **Sentence 1** to **Sentence 4**.

In IS terms, such a prose expression of the semantics of an interface element (or, in general, of a communication artifact) is called its *intended interpretation*.

Finally, here is how we spread the four sentences across groups:

Sentences 1 and **2**: Group E (must be conveyed explicitly by the interface)

Sentences 3 and **4**: Group D (must be accessible on demand)

3.2 Translation into QNL

A quasi-linguistic reading of an interface involves in particular identifying the "blocks" of symbols that belong together, i.e., that make up a "phrase" or a "sentence" in the interface semiotic system.[3] In this example, we assume that the two subareas identified by (a) and (b) on Figure 1 are correctly "parsed" as belonging together, and as jointly falling within the scope of the "Search limits" subtitle above them. We apply QNL translation only to the block formed by areas (a) and (b), which we call the (a-b) block.

The most complex semiotic construct in that block is (b), a vertical list of horizontally aligned checkboxes, each followed by a noun phrase. This assembly forms two columns, for which *no headings* are supplied. By hypothesis on group B (in particular, familiarity of the target community with common interface primitives and constructs), we translate a ticked checkbox in this context by "I select", and an unticked one by "I do not select". Also by hypothesis on B, we join the individual lines of the assembly with the conjunction "and". Thus, for example, <u>if no checkbox were ticked</u>, the whole assembly would read:

(I) I do not select 'Book' and I do not select 'Corporate report' … and I do not select 'Working paper'.

Let us now turn to the block (a), containing the words "Document type". Because of its geometric proximity to the checkbox section, we translate that block to the complex noun phrase:

(II) document type(s) that I select as follows:

This is begging for a verb. To find one, we turn to the subtitle "Search limits," which, although lying outside the scope of our example, is the direct context of the (a-b) block. On the basis of the quasi-linguistic "valence" of the (a-b) block (the way it begs to be completed to form a coherent quasi-linguistic fragment), we convert the subtitle to the following actual verb phrase initial segment:

(III) Limit search to

Globally, again <u>if no checkbox is ticked</u>, the (a-b) block translates into the sequence: (III) (II) (I), i.e.:

(IV) Limit search to Document type that I select as follows. I do not select 'Book' and I do not select 'Corporate report' and … and I do not select 'Working paper'[4].

---

[2] It could be argued that the sentences need not be added in the first case (no restriction by document type). However, we believe it is important for the user to realize exactly what kind of filtering is made possible by the criterion in order to decide whether or not to use it.

[3] Since we are dealing here with a visual interface, we will limit the discussion to the visual modality. However, other modalities, as well as multi-modality, can be treated in similar ways (see Marcoux & Rizkallah 2009).

[4] If some of the checkboxes are ticked, (IV) is adjusted accordingly.

3.3 Comparison

According to our analysis, when the user does not want a limitation by document type, Sentence 1 *must* be in E, i.e., explicitly inferable from the interface. The comparison to be done is thus between (IV) and Sentence 1. But (IV) and Sentence 1 are in direct contradiction: a problem.

When the user wants to apply a limit by document type, Sentence 2 must be conveyed by the interface. Thus, Sentence 2 must be compared to the appropriate version of (IV), and they are indeed semantically very close to each other: no problem here.

A semantic equivalent of Sentence 3 is actually accessible through a tool-tip pop-up in the current interface; Sentence 3 is thus in group D. But there are two problems with it: (a) There is no visible indication in the interface that a pop-up tool-tip is available for this section of the search form and will appear if the mouse pointer is left motionless long enough in the section. (b) The equivalent of Sentence 3 is not shown directly in the tool-tip pop-up itself, but rather in a pop-up window that must first be opened by clicking on a link in the tool-tip pop-up, though increasing the "distance" between the sentence and the place it is needed to assist the user in understanding the interface.

Sentence 4 is nowhere in the interface (not even if we consider the whole interface, not just the (a-b) block).

3.4 Reconstruction

We have reached a point where we have identified four "elements of meaning" to be selectively conveyed to the user in response to choices, and expressed them as four sentences in natural language. We have also identified problems with the way the current interface conveys three of those elements to the user. The task of reconstruction consists in modifying, adding, eliminating, and/or moving interface elements with the following goals in mind:

A. The nature and implications of the choices offered to the user are clear.
B. The way in which the user can express her/his choices is clear.
C. Sentences 1 to 4 are clearly inferable from the interface in due time.

The task of reconstruction is of course a creative process and numerous solutions are possible. The IS framework, however, can serve as a kind of check-list to guide the process. One possible solution is presented in Figures 2 and 3. Figure 2 shows the document-type section when no limitation is selected (the default) while Figure 3 shows it after the user has chosen to apply a limitation by document type.

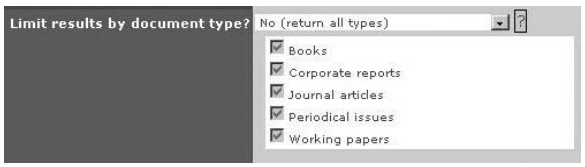

Figure 2 - Proposal (no limitation applied)

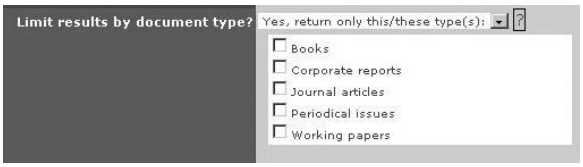

Figure 3 - Proposal (limitation applied)

Goal A: The presence of the question "Limit by document type?" next to a drop-down list makes it clear that a choice is offered to the user. The implication of this choice is made explicit by the use of complete answers in the drop-down list: each answer yes/no, is followed by a phrase explicating the consequence of making that choice.

Goal B: The strategy for telling the user how to indicate her/his choice relies on her/his ability to recognize a drop-down list functions and on her/his understanding of how it works. Thus, a quite reasonable hypothesis placed on B.

Goal C: Sentence 2 is presented to the user in much the same way as in the original interface (once she/he has made the choice of applying a limitation). Now, however, Sentence 1 is also presented to the user when she/he chooses not to apply a document type limitation. It is presented as the sequence of text segments "Limit by document type? Yes, return only this/these type(s):" followed by a list of document types in which some types are ticked.

The proposed solution places Sentences 3 and 4 in D (shown on demand). However, it avoids the "distance" problems mentioned earlier by (a) displaying next to the drop-down list a visible clue that additional

information is available (the "?" enclosed in a rectangle)[5] and (b) placing the equivalent of Sentences 3 and 4 right into the tool-tip pop-up where the text might be something like[6]: Every document has one and only one type. There are no other document types than those shown.

4. CONCLUSION AND FUTURE WORK

There is little doubt that the same conclusions about the analyzed search interface could have been reached without the use of IS but we believe IS can accelerate convergence towards usable solutions by placing semantic issues at the top of the priority list. Still, this solution has not been user-tested yet and might have to be adjusted as user testing remains the ultimate measure of success. However, we claim that an IS analysis makes it harder not to ask right questions and makes it easier to ask them early in the design (or redesign) process. Thus, the solution proposed runs a fair chance of not suffering from any major flaw at the semantic level and can ultimately help targeting more precisely user testing.

Intertextual semantics itself is a work in progress. User experimentation with interface design/redesign will help develop the framework, refine the methods, and evaluate the possible benefits of the approach. The search interface used in this paper is a natural candidate, but we also plan to work on different kinds of interfaces.

REFERENCES

Brooks, F.P., 1975. *The mythical man-month: essays on software engineering*, Reading, Mass.: Addison-Wesley Pub. Co.

Hollingsed, T. & Novick, D.G., 2007. Usability inspection methods after 15 years of research and practice. In *Proceedings of the 25th annual ACM international conference on Design of communication*. El Paso, Texas, USA: ACM, p. 249-255.

Marcoux, Y. & Rizkallah, É., 2009. Intertextual semantics: A semantics for information design. *Journal of the American Society for Information Science and Technology*, 60(9), p.1895–1906.

Nielsen, J., 1994. Heuristic evaluation. In J. Nielsen & R. L. Mack, (eds) *Usability inspection methods*. New York: John Wiley & Sons Inc., p. 25–62.

Nielsen, J., 1995. Usability inspection methods. In *Conference companion on Human factors in computing systems*. Denver, Colorado, United States: ACM, p. 377-378.

Rubin, J. & Chisnell, D., 2008. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests* 2nd ed., Indianapolis IN: Wiley.

de Souza, C.S., 2005. *The semiotic engineering of human-computer interaction*, Cambridge, Mass: MIT Press.

Wharton, C. et al., 1994. The cognitive walkthrough method: a practitioner's guide. In J. Nielsen & R. L. Mack, (eds) *Usability inspection methods*. New York: John Wiley & Sons, p. 105–140.

Wierzbicka, A., 1992. *Semantics, culture, and cognition: universal human concepts in culture-specific configurations*, Oxford University Press.

[5] Note that the meaning of the "?" enclosed in a rectangle is assumed to be already known to the user (i.e., as belonging to B).

[6] The sentences are semantic information that has been moved from the pop-up window to the tool-tip itself.