RichMoLIC: Coupling Interaction and Interface Design

Ugo Sangiorgi
SERG, Departamento de Informática, PUC-Rio
R. Marquês de São Vicente, 225
Gávea, Rio de Janeiro, RJ, Brasil, 22451-900
usangiorgi@inf.puc-rio.br
http://www.serg.inf.puc-rio.br

Simone D.J. Barbosa
SERG, Departamento de Informática, PUC-Rio
R. Marquês de São Vicente, 225
Gávea, Rio de Janeiro, RJ, Brasil, 22451-900
simone@inf.puc-rio.br
http://www.serg.inf.puc-rio.br

Abstract
Storyboarding is a low-fidelity prototyping technique with large acceptance among User-Centered Design (UCD) adopters. We argue that interaction design could benefit from a model of interaction bound to interface sketches commonly used at storyboards. In our proposal, the designer first models the interaction using the Modeling Language for Interaction as Conversation (MoLIC), and then relates it to an execution model in which states are associated with user interface sketches. MoLIC reduces the gap identified in using common HCI representations such as scenarios, task models, and storyboards, encouraging designers to structure and represent mechanisms for interaction breakdown recovery. It can also help to evaluate the design’s consistency and comprehensiveness, aiming to support design decisions that might otherwise be hard to represent.

Keywords
Interaction Design, Prototyping, Semiotic Engineering

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User interfaces – Evaluation/ methodology
**Introduction**

Low-fidelity prototypes are approximate representations of an envisioned design for some problem and are useful when discussing ideas with stakeholders \[7\]. They are cheaper to modify than an actual piece of software and help users to realize how they would like the final product to (or not to) perform. The most important point, however, is that they help designers and stakeholders to discuss about design alternatives and their consequences.

HCI design has traditionally embraced storyboards, scenarios and sketches \[3, 4\] in order to build interaction prototypes to improve discussion at early design stages. Given that they are easily understood by stakeholders and can be easily changed, they are used in various stages of design.

In this paper we articulate the coupling of interaction – for what we use the Modeling Language for Interaction as Conversation (MoLIC)\[1\] and interface design – with an execution model based on sketches, inspired on storyboarding techniques. This approach is supported by MolicDesigner\[^1\].

MoLIC has its roots in Semiotic Engineering \[5\] – an HCI theory that views the user-system interaction as a communicative process involving users, system, and designers, and in which the interface plays the designer’s role in a conversation with the user. Since the designer is no longer present at interaction time, he/she needs to tell the users, through the user interface, how they can or must interact with it in order to achieve their goals. The quality of this communication from designers to users is called communicability \[5\].

**Coupling Interaction and Interface Design**

MoLIC was first devised as an informal language to be read by humans but not processed by computers, emphasizing the epistemic value of the representation in supporting design decisions. However, we have come to realize that it would also be valuable to, at a later design stage, support a more concrete realization of the design, by associating storyboard snapshots to MoLIC elements. This way, we take advantage of a more structured, executable representation such as a state machine, in which each state represents a moment during user-system interaction. MoLIC seems to fit this task, since it depicts the turn taking between the user and the system, clearly sequencing interaction while allowing alternative interactive paths to be specified for the same goal.

We are aware that anticipating user interface design at an early stage might put an unnecessary overload on decisions that might not even get to be coded. As the interaction design matures, however, it becomes useful to start elaborating sketches to help envision the solution at a more concrete level. Sketches are often preferred over a functional prototype, because they can usually be more easily changed. The RichMoLIC notation – in which we propose to bind sketches to MoLIC elements – is flexible enough to allow the designer to work on both interaction and user interface design at alternate moments throughout the design process.

Figure 1 shows the RichMoLIC notation. To illustrate the idea, we have reverse-engineered Google’s web search. It represents our vision of the communication that takes place between the user and the designer’s deputy. For the sake of simplicity, the example only comprehends the search functionality itself, and not the advanced search, language selection, and so on.

The sketches inside each scene share the scene’s conversation topic. In state machine terms, each sketch represents a different interaction state. With this coupling, we are associating the conversation defined in MoLIC with an executable specification defined through the interconnected sketches, in which MoLIC conversational utterances are mapped onto state transitions. They may be connected to the utterances

\[^1\]http://code.google.com/p/molic-designer/
Figure 1: RichMoLIC model representing Google’s search

that have their sources or targets at the scene. The approach exemplified here assumes that all sketches within a scene share the same subject, varying in content. In this way, scenes may be viewed as groups of states that share a common discourse, while individual sketches are states of the interface itself.

There are two scenes, each one with different states represented at this example: Search, where user and deputy talk about search terms (represented by the dialogue $d+u$: search terms); and View Results, where both talk about the results from the searched terms.

From state 1, the user may utter to search or jump straight to the first result, if he "is feeling lucky". Both user utterances end up on processes (the deputy's turn to talk) and from there, our interpretation of what may happen is: a) The user typed no terms and uttered "search" (empty search) – In this case, the user is taken to the exact same page again, without any additional information on what has happened. For us, it appears to be the designer's interpretation of a breakdown (in fact providing a Supported Recovery opportunity). He may have thought the user wanted to search for something and mistakenly uttered search (he wouldn't be on a search site if he did not have anything to look for). So the designer's choice was to just get the user back to the same page again; b) The user typed no terms and uttered "I'm feeling lucky" – In this case, the breakdown is more explicit, since the user is led to an explanation of what this feature means (state 2). In MoLIC, a breakdown typically leads to an explanation, back at the original scene, or to a scene with an explanatory content. It is important to notice that the "I'm feeling lucky" is only available from state 1; c) The user typed some terms and uttered "search",
and one or more pages were found – In this case, the result page is shown. In our perspective, that is represented by the deputy saying \textit{d:results were found}, and by showing the results at the \textit{View Results} scene, at the respective sketch (state 3). From this state, the user may utter to go to a specific result (\textit{u: go to link}), which is represented by a dotted line, or to search again (\textit{u: search}; or \textit{d).The user typed some terms, uttered “search” or “feeling lucky” and no pages were found – In this case, the user is led to a page with a message showing that nothing was found (state 4). For us, the designer doesn’t seem to have treated it as a breakdown. The functionality of “I’m feeling lucky” becomes clear in this example. It bypasses the whole scene of results and goes straight to the end of the interaction, in which Google’s role actually ends. Our view is that the utterance of the system to “say” a result was found is exactly to go straight to the found result. From any state inside the \textit{View Results} scene, the user may initiate a conversation aiming at another search (instead of having to get back to the previous \textit{Search} state). This is represented by the utterance \textit{u: search} from the \textit{View Results} scene. It is important to notice that, with MoLIC, the designer might choose, at the occurrence of a breakdown, to lead the user through different interaction paths.

Discussion
How one can tell a bad design apart from a good design depends on the context; what works well for a domain may not work in another. To provide a pop-up dialog with a warning, a label at a page, or no label whatsoever is a very simple example of the kind of decisions that we are willing to support with RichMoLIC. If we are able to identify patterns and good practices at the interaction model, we could reuse them to give a consistent interaction experience to users [2]. For instance, when a user triggers \textit{search} without search terms, it seems to be common practice among search engines to take the user to the same page, with no warnings whatsoever. MoLIC’s main goal is to make explicit the communication that takes place between \textbf{designers} and \textbf{users} through system’s interface. Whereas Google’s search signifies the empty search behavior as a breakdown, \textit{Ebay} shows a page with all their categories, meaning that the user may want to know more about what he is able to search. Regardless of what an “empty search” means in any given domain or application, the crucial matter is to make the decisions explicit.

References