Multi-level Communicability Evaluation of a Prototyping Tool

Vinícius Segura, Fabiana Simões, Gabriel Sotero,
and Simone Diniz Junqueira Barbosa

Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil
{vsegura,fpsimoes,gsotero,simone}@inf.puc-rio.br

Abstract. Semiotic engineering views human-computer interaction as a form of human communication between designers and users, mediated by a computer system. If we consider a design application, such as a prototyping tool, this communication is about the construction of a second communication, one between the user of the prototyping tool (in the role of the designer) and another user, who will interact with the system being designed. This article explores an extension to the Communicability Evaluation Method for design tools. This extension focuses not only on considering the kinds of communicability breakdowns, but also on what abstraction level they occur.

Keywords: semiotic engineering, communicability evaluation, prototyping tools.

1 Introduction

Semiotic Engineering studies Human-Computer Interaction from a communication-centered perspective, defining it as a form of human communication mediated by a computer system [9]. To Semiotic Engineering, computer systems are *meta-communication artifacts*, that is, artifacts that communicate the designer's meta-message.

When considering the case of interface design tools (DT), and more specifically the case of prototyping tools, the complexity level of this communication between designers and users increases. Apart from the aforementioned meta-communication between the designer of the DT and the user of the DT (about how the DT can and should be used), there is another meta-communication being defined during the use of the DT: one between the user of the DT, in the role of the designer of a new computer system, and the users of this new system.

The Communicability Evaluation Method (CEM) [5] evaluates the communicability of an interactive system, which is the quality of the designer's meta-message, not only in terms of the effectiveness of his direct messages to users, but also of the afforded conversational paths. CEM evaluates the reception of the meta-message, i.e., it identifies some interaction breakdowns that occur while a user interacts with a computer system.

In this paper, we studied how to apply the CEM in the evaluation of a DT, taking into consideration the fact that DTs comprise more than one meta-message during interaction time. The tool chosen for this analysis was UISKEI (User Interface Sketching and Evaluation Instrument) [8], focusing on its interaction behavior definition functionalities. Due to the plurality of meta-messages, while analyzing the collected data and

M. Kurosu (Ed.): Human-Computer Interaction, Part I, HCII 2013, LNCS 8004, pp. 460–469, 2013.

applying the CEM, we found it necessary to extend the method. The main change in the original method is that analysis is not only based on the exposition of the communicability breakdowns, but also on the abstraction levels in which they were observed.

The next section briefly describes UISKEI, the design tool we evaluated. Section 3 presents the original CEM and describes the evaluation scenario. Section 4 reports the results of a preliminary evaluation with the original CEM. Section 5 presents the extensions to CEM we found necessary to better evaluate user interface design tools, and section 6 presents the evaluation with the extended CEM. We conclude the paper with a discussion of the benefits of the extended CEM and point to future work.

2 UISKEI

UISKEI [7,8] is a pen-based prototyping tool developed to aid designers in the early stages of interactive systems design, namely: interface building; behavior definition; and prototype simulation. For this study, we focused only in its behavior definition functionalities. Behavior in UISKEI is defined according to an event-condition-action (ECA) model: when an *event* occurs, if all the *conditions* are met, then all the *actions* are executed. In its canvas, UISKEI shows the current selected ECA as a mind-map, and also all the created ECAs in the composition in an ECA Manager side-panel.

The pen-based interaction is maintained while defining behavior. To define an ECA, the user has to first select the element that will trigger the event and add an ECA to it. To do so, the user interacts with buttons in the top of the canvas, which allow him to add, remove, clone and navigate between ECAs. Given a selected ECA, the user may add conditions and actions. This is accomplished by drawing lines, with their starting and ending points determining not only if an action or a condition is being added, but also, in the case of an action, its kind. The final parameters of an action/condition being added are defined through a pie menu. This method allows the user to define an action/condition without raising the pen, in a single stroke.

3 CEM

CEM aims to evaluate the quality of the designer's meta-message reception by users [5,4]. Participants are invited to perform tasks using the system being evaluated and the interactions are recorded in videos. Later, evaluators analyze the recordings, aiming to understand how each user interacted with the system, which communication breakdowns happened, and when.

Communication breakdowns can be considered "moments of interaction in which the user demonstrates that he did not understand the metacommunication of the designer, or moments when the user finds difficulty to express his intention of communication in the interface" [9,5,4,1]. They are categorized by user utterances – expressions in natural language that allow the evaluator to presume what the user could have said when the breakdown occurred. CEM has 13 user utterances: Where is it? What now? What is this? Oops. Where am I? What happened? Why doesn't it? I can't do it this way. I can do it otherwise. Thanks, but no, thanks. Looks fine to me. Help! I give up.

3.1 Test Structure

The test was conducted with four participants, all of them interaction designers with no previous experience with UISKEI. The participants were divided in two different groups, according to their profile: those with programming experience, coming from a Computer Science background (P1 and P2); and those with no programming experience, coming from an Industrial Design background (D1 and D2).

The main objective of the experiment was to observe how (or if) UISKEI's ECA model was understood by users. Moreover, we also aimed at testing the hypothesis that the ECA model is easier to grasp by people with programming experience than by people without it.

In the experiment, during which both audio and video were captured, participants used a Wacom Cintiq 24 tablet and its provided stylus. Before starting the experiment, they were asked to complete a training session based on a step-by-step guide on how to define interaction behavior and use the simulation mode with UISKEI. Participants were encouraged to use the training material as a help resource during the tasks execution.

The test scenario read as follows:
An interaction designer is working in the design of a computer system and decides to create a functional prototype to present some of his ideas to his co-workers. He has already sketched the interface of this system and now wants to define how this interface behaves when a user interacts with it. He writes down a list of imagined behaviors of the system and now has to define it using UISKEI.

Each item in the imagined behavior list was presented to participants as a task. There were 5 tasks in total to be executed in sequence (T1-T5). With this scenario in mind, participants received an UISKEI project with the interface pre-drawn and some initial behavior defined. The tasks were chosen in order to cover most UISKEI's features regarding the creation and editing of behaviors. At any point in the test script participants could use the "Simulation" mode to evaluate the behavior they specified.

4 Evaluating UISKEI with the Original CEM

Participants did not experience operational difficulties creating ECAs, but had some problems in conceptually defining them. Only P1 and D1 executed all five tasks successfully. D2 had many problems and gave up in all tasks, except in T2. He spent a long time in T1 and T3, which made him quickly give up in T4 and T5. P2 created more ECAs than necessary in T1 and T5. Also, in T3, P2 did not add all the required conditions asked in the task description, because of wrong assumptions about UISKEI defaults. Additionally, as we encouraged participants to think aloud [3] during the experiments, the time they took to perform the tasks was not measured.

The results obtained with CEM are depicted in table 1. It is possible to see that most breakdowns were "Oops!" and "Looks fine to me.". The latter shows a big difference in number of occurrences between the two profiles. Other labels also show discrepancy between profiles, such as "Why doesn't it?", "I can't do it this way." and "What happened?".

During the test, P1 constantly mentioned that he missed an *else statement* to define ECAs. This clearly shows that he attempted to map basic programming structures he

Table 1. Number of utterances by profile and ordered by total number of occurrences

Utterance	P*	D*	Total	Utterance (cont.)	P*	D*	Total
Oops!	33	38	71	What is this?	8	8	16
Looks fine to me.	8	55	63	Where is it?	8	5	13
Why doesn't it?	16	26	42	What happened?	11	1	12
Help!	16	16	32	Where am I?	4	3	7
I can't do it this way.	5	15	20	I give up.	0	4	4
What now?	6	11	17	I can do otherwise.	0	1	1

knew onto ECAs. Another point raised by P1 was the lack of a clearer overview of the created ECAs. These missing features were tagged as "Where is it?"

P2 failed in 3 of the 5 tasks. However, he was not conscious of these failures, because the tests he made during the simulation did not cover enough cases. These failures were the result of a sequence of partial and temporary failures. For example, P2 did not notice the system response when he tried to clone an ECA. So he repeated the operation several times before realizing that he had already created several clones. This partial failure was tagged as "Why doesn't it?", and the final result (the total failure), as a "Looks fine to me."

When we examined the results of D1 and D2, it was clear to us that the main problems they faced were related to the underlying language of UISKEI, and not to its user interface. Their frame of mind and the way they modeled the problem, i.e., the solution they developed to the given task, were not easily mapped onto the ECA structure. This led us to consider abstraction levels when applying CEM, described in the next section.

5 Extensions to CEM

When analyzing the participants' utterances, we noticed that many of them occurred before the actual use of the tool. We were able to observe these utterances while participants were planning their solutions to the task at hand. This is related to the nature of the problem, since UISKEI is a tool to support interaction design, i.e., to create a new meta-communication message (developed by the user, in the role of designer of a new solution) within a meta-communication (the one between the UISKEI's designer to the UISKEI's user).

Building upon the work of Leitão and de Souza [2], we classified the communicability breakdowns not only by the utterances defined in CEM, but also by the abstraction levels in which they occurred. Figure 1 shows the abstraction levels model that we used.

To define this model, we considered that the behavior definition process with UISKEI happens as follows:

- The user receives the task and reformulates it using his own words. This happens at an **interpretative** level, since it relies on the user interpretation of the task description.

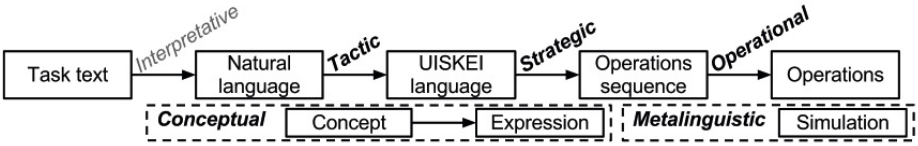

Fig. 1. Adopted abstraction levels

- The user's interpretation of the task is translated into the ECA model. This translation occurs at the **strategic** level, since this is related to the process of establishing objectives for using UISKEI.
- The user then reaches the **tactical** level, settling an action plan with a sequence of operations needed to reach the objective(s) defined at the strategic level.
- Finally, the user performs the previous planned operations. This stage happens at the **operational** level, focusing on the individual expression of each operation.

In this model, we also consider the problem of mapping a concept to a term in the user interface. For example, consider the definition of a condition for the status of a textbox (enabled or disabled). On the one hand, if the user does not know that enabling a textbox means making it possible to enter information, there is a concept problem in comprehending the effect of enabling a widget. On the other hand, if the user knows that effect, but refers to it as "activate" instead of "enable", we have an expression problem. We considered that concept problems are **strategic** problems, since it reflects the definition of the objectives. The expression problems were considered as **tactical** problems, because they reflect on how the user will describe the action plan to achieve his goals. In some cases we were not able to identify in which side of the concept/expression specter the problem occurred. When this happened, we said that the problem was at a **conceptual** level.

Once the user has defined a behavior, he has the possibility of verifying his solution using the simulation mode. By doing so, the user (in the role of the designer) evaluates whether the solution defined with UISKEI results in the expected behavior. Because the prototype simulation is a form of evaluating the prototype meta-communication, we considered problems found during the simulation to be at a **metalinguistic** level. Breakdowns at this level often cause the user to re-evaluate the products of the previous abstraction levels.

In addition to the need for different abstraction levels, the experiments revealed the recurrence of a phenomenon that did not fit any of the 13 user utterances of CEM. Participants often showed symptoms of uncertainty and insecurity when defining behaviors. We have observed that these symptoms occurred in three different situations:

- **Uncertainty of the project:** When the participant is not certain whether the defined behaviors will have the expected result. This symptom appears frequently and precedes the prototype simulation. Participants often attributed the expression "Will it work?" to this symptom. It was more common with participants with CS background.
- **Uncertainty of the product:** When the participant is not sure, even after the simulation, about the defined behaviors. Typically, this symptom came with the expression

"I think I'm done", in a doubtful tone. The participant defines the behavior, tests it in the simulation mode, apparently obtaining success, but still questions himself regarding the success of the task. In general, this symptom is followed by a new execution of the simulation mode, or by a revision of the created ECAs. It was also more common among participants with CS background. The difference between this utterance and "Looks fine to me." is that the participant has actually reached the goal, but is uncertain of that.

– **Insecurity about the adequacy of the tool:** When the participant manifests his belief that the tool is not for him. This symptom was perceived among participants with both backgrounds. Typically, it was presented with the expression "This is not for me." or similar. Participants with CS background, in general, presented this symptom when comparing the definition of ECAs and programming languages, manifesting that they "could do it much faster with Notepad". Participants with Design background presented this symptom as soon as they saw the mind-map structure. Comments like "this is a programmers' issue" were frequent.

Another interesting point is the use of the simulation mode for validating of the logic being defined. A comparison can be made with the discussion of *reflection in/on action* [6], since the participants chose to reflect about the created behavior after the simulation (*on action*) instead of being critical about it while they were defining the behavior (*in action*).

6 Evaluating UISKEI with the Extended CEM

We revisited the tagging from our first evaluation, this time considering the aforementioned abstraction levels, resulting in table 2. The table also describes the symptoms most commonly associated with the occurence of a given tag at a given abstraction level. In this analysis, breakdowns that occurred at an interpretative level were not considered, since they reflect problems outside the scope of the tool. Moreover, when a breakdown was perceived in more than one abstraction level, it was tagged only at the highest perceived level (the left-most cell in figure 1).

Table 2. Number of utterance occurrences separated by abstraction level and participants profile

Utterance	Lvl	P*	D*	Total	Symptoms
Oops!	conc	6	29	35	The user recovers himself of an error caused by an undesired effect or a wrong terminology choice.
	tat	1	2	3	The user quickly recovers from an action plan formulation error.
	op	31	35	66	The user quickly recovers from an error when executing an operation from the action plan.
Looks fine to me.	conc	6	29	35	The user expects wrong effects or uses a wrong terminology, without noticing the error.
	str	1	3	4	The user elaborates objectives that the designer can't comprehend, but is not aware of the problem.

Table 2. (*continued*)

Utterance	Lvl	P*	D*	Total	Symptoms
Looks fine to me.	tat	0	14	14	The user elaborates an incorrect action plan and doesn't notice the error.
	op	1	9	10	The user incorrectly performs an operation of the action plan and does not notice the error.
Why doesn't it?	str	1	0	1	The user doesn't understand why the designer can't comprehend his objectives.
	tat	1	1	2	The user insists in repeating an action plan that doesn't produce the desired results.
	op	8	1	9	The user insists in repeating an operation that doesn't produce the desired results.
	meta	6	24	30	The user, when checking the developed solution, notices that it isn't executing according to what he expected.
Help!	conc	2	0	2	The user can't understand what an interface expression means and searches for information through explicit meta-communication.
	str	2	2	4	The user doesn't know how to express his objectives and searches for information through explicit meta-communication.
	tat	3	9	12	The user does not understand the difference between the interface signs, staying unsure about which one he must use to build his action plan. After that, the user searches for information through explicit meta-communication.
	op	9	5	4	The user doesn't understand an interface sign and searches for information through explicit meta-communication.
I can't do it this way.	str	2	4	6	The user convinces himself that his formulation of objectives is wrong and tries do reformulate them.
	tat	2	10	12	The user gives up an action plan that he developed and decides to elaborate a new plan, restarting the task.
	op	1	1	2	The user tries to do an action several times, without getting the desired effects, convincing himself that he is doing something wrong.
What now?	str	0	2	2	–
	tat	6	8	14	The user knows what is the desired effect, but doesn't know how to define the action plan to accomplish it.
	op	0	1	1	The user does not know how to perform an operation from his action plan.
What is this?	conc	0	1	1	The user does not understand the meaning of an interface expression and searches information through implicit meta-communication.
	tat	0	1	1	The user doesn't understand the difference between the interface signs, ending up unsure of which one he will use to elaborate his action plan.
	op	8	6	14	The user doesn't understand the meaning of an interface sign and searches information through implicit meta-communication.

Table 2. (*continued*)

Utterance	Lvl	P*	D*	Total	Symptoms
Where is it?	str	0	2	2	–
Where is it?	tat	5	1	6	The user knows what he is trying to say, but does not know what term he should use among the system's options.
	op	3	2	5	The user knows what he is looking for, but can't find an interface element to do it.
What happened?	op	11	1	12	The user doesn't understand the system's answer to what it is told to.
Where am I?	tat	2	3	5	The user says something to the system that should be preceded by another utterance, not making sense when said without context.
	op	2	0	2	The user says something to the system that would be appropriate in another communication context.
I give up.		0	4	4	The user believes that he is not able to achieve his objective and interrupts the interaction.
I can do otherwise.	op	0	1	1	The user communicates his action with unexpected signs, since he does not understand what the system is saying about the solutions to reach his goals.

No occurrences of the "Thanks, but no, thanks!" utterance were observed. Despite the 4 occurrences of "I give up", we decided not to specialize it, since it represents a breakdown of the user interaction at all the levels of abstraction. For the original CEM, "Looks fine to me", as well as "I give up", characterizes a complete breakdown during which the user is unconscious of his failure. In a similar way, for this study, we used "Looks fine to me" as an utterance for when the user manifested unconsciousness of failure, however, without necessarily corresponding to a complete failure. The simulation mode made it possible for users that presented "Looks fine to me" symptoms to gain consciousness of their failure while testing the prototype. This utterance, therefore, was also used for classifying temporary failures as well.

The multi-level application of CEM let us identify in which communication context each breakdown in interaction occurs. On the one hand, breakdowns in operational and tactic levels were more frequently observed in the communication between UISKEI's designer and its user. On the other hand, breakdowns in strategic and metalinguistic levels were observed, in general, when the message of the participant was specified for the users of the system being prototyped (UISKEI's user in the designer's role). The problems at those levels reflect moments of the interaction in which the user demonstrates difficulties in expressing his communication intentions as a designer. Therefore, the multi-level application of CEM let us identify in which communication context each breakdown in interaction occurs. This information let us identify which role the user of the DT was playing when the breakdown occurred: user of the DT or designer of a new system.

The identification of these different types of problem takes us to different kinds of improvement suggestions to the DT being evaluated. For instance, considering the paraphrase of the designer meta-message [9] below, table 3 shows which parts of the designer's meta-message should be redesigned based on the level in which a communication breakdown occurs:

“Here is my understanding of *who you are* (1), what I’ve learned *you want or need to do* (2), in *which preferred ways, and why* (3). This is the system that I have therefore designed for you, and this is *the way you can or should use it* (4) in order to fulfill a range of purposes that fall within this vision”

Table 3. Relationship between the adopted abstraction levels and the meta-message parts.

	Conceptual	Strategic	Tactic	Operational	Metalinguistic
Passages	1	1, 2, 4	2, 3, 4	3, 4	1, 2, 4

6.1 Interpretation of Results

"Looks fine to me" occurred in 4 of the 5 levels of our model. Participants tended to recover more rapidly from the occurrence of "Looks fine to me" when this utterance occurred at the operational level. At the strategic and tactical levels, this recovery was harder, usually occurring only after the use of the simulation mode.

The "Why doesn’t it?" utterance stands out for its occurrences at the metalinguistic level, when the user first simulated his solution and then observed the existence of an error. This user utterance was usually followed by a recovery from previous failures of which the user was unaware – typically, the results of "Looks fine to me" user utterances. In most cases, this recovery spotted situations where the user was able to assimilate the strategic/tactical model of the tool. When this recovery did not happen, new failures were observed or even a new occurrence of the "Why doesn’t it" user utterance. The use of a multi-level CEM allowed us to observed that, at the metalinguistic level, the user was trying to reformulate his approach to the problem based on the results of the simulation.

"Help!" is another example of a user utterance that was observed at several levels of our model. At the operational level, its use is similar to its traditional use in CEM, that is to identify a situation where the user is trying to comprehend the communicative act of the system designer. With the multi-level approach to CEM, we could discern situations where the user tried to find out how to establish his own communicative act using the UISKEI language (*strategic*) or translating his own "speech" to the UISKEI language using a series of operations (*tactical*).

Each of the other utterances were more strongly characterized by one of the levels of the model. For example, the most frequent utterance was operational "Oops!", with more than 60 occurrences. Almost all participants had problems with the pen-based input, for example, inadvertently deselecting elements or misperforming the gestures for defining conditions and actions. The high frequency of this utterance shows that UISKEI lacks adequate feedback for user actions.

"I can’t do it this way." occurred around 20 times, especially at the tactical level. Participants took a wrong interaction path several times and this led them to reformulate their plan: they had to reformulate either an ECA (*strategic*), the sequence of actions needed to define it (*tactical*), or one operation on this sequence (*operational*). This utterance usually occurred only after the participant ran the simulation Mode, when he would stop and reflect on what could be wrong.

7 Conclusion

CEM investigates the breakdowns that occur in user-system communications. While evaluating a DT, however, the set of breakdown utterances had to be expanded in order to capture the context in which the breakdown occurs: if it is between the DT designer and his user, or between the DT user taking on the designer's role and the user of the system being designed. The multi-level approach of CEM proved to be useful when capturing this information, helping to shed light on the breakdown causes, the user's role when it occurred, and the part of the designer's meta-message that should be revised.

In order to apply the multi-level approach of CEM to other DTs, further revision of the levels schema is needed. The one discussed in this work was developed taking into consideration the inherent characteristics of UISKEI, which not only is a DT, but also explores pen-based interaction. Nevertheless, the evaluation at multiple abstraction levels showed itself promising while working with systems that are used to generate new systems.

Acknowledgements. We thank all study participants who selflessly devoted their time to our research. We also thank the financial support of CNPq (#313031/2009-6 and #308490/2012-6).

References

1. Barbosa, S.D.J., Silva, B.S.: Interação Humano-Computador, 1st edn. Campus-Elsevier (2010)
2. Leitão, C.F., de Souza, C.: Semiotic Engineering Methods for Scientific Research in HCI. Morgan and Claypool Publishers (2009)
3. Lewis, C.: Using the "thinking-aloud" Method in Cognitive Interface Design. Research report, IBM T.J. Watson Research Center (1982)
4. Prates, R.O., Barbosa, S.D.J.: Introdução à Teoria e Pática da Interação Humano Computador fundamentada na Engenharia Semiótica. In: Jornadas de Atualização em Informática (JAI), pp. 263–326. JAI/SBC, Rio de Janeiro (2007)
5. Prates, R.O., de Souza, C.S., Barbosa, S.D.J.: Methods and tools: a method for evaluating the communicability of user interfaces. Interactions 7(1), 31–38 (2000)
6. Schön, D., Bennett, J.: Reflective conversation with materials. In: Bringing Design to Software, pp. 171–189. ACM, New York (1996)
7. Segura, V.C.V.B., Barbosa, S.D.J.: Shape-based versus sketch-based UI prototyping: a comparative study. In: Proceedings of the 10th Brazilian Symposium on on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, IHC+CLIHC 2011, pp. 162–166. Brazilian Computer Society, Porto Alegre (2011)
8. Segura, V.C.V.B., Barbosa, S.D.J., Simões, F.P.: UISKEI: a sketch-based prototyping tool for defining and evaluating user interface behavior. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI 2012, pp. 18–25. ACM (2012)
9. de Souza, C.S.: The Semiotic Engineering of Human-Computer Interaction. Acting with Technology. MIT Press (2005)