Going back and forth in metacommunication threads

Ingrid Teixeira Monteiro   **Eduardo Tiomno Tolmasquim**   **Clarisse Sieckenius de Souza**

Semiotic Engineering Research Group – SERG
Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente 225
22451-900 Rio de Janeiro, RJ
{imonteiro, clarisse}@inf.puc-rio.br, eduardot@aluno.puc-rio.br

ABSTRACT

This paper presents an exploratory study about how savvy end users configure return points in mediated interactions with Web applications through WNH (Web Navigation Helper), a user agent that supports rephrasing, explanation, commentary and elaboration of interaction in scripted Web activities. The study is part of a long-term research on self-expression through software programming. Savvy users creating mediation dialogs for the benefit of other users (third parties) are actually engaged in a human-computer interaction (HCI) design process involved in end user development (EUD). The study shows that decisions about when, where, how and why to go back to previous points in the design of mediated conversation with WNH are very complex, even for savvy users. More importantly, we collected powerful evidence of the participants' interpretation of what interaction through WNH is all about. We believe that such evidence stands for their intuitions about HCI in a broader sense. The main contribution of the study is thus to illuminate aspects of EUD and HCI that haven't been discussed to-date and to propose questions for new kinds of investigation.

Author Keywords

Back navigation; dialog management; mediated metacommunication; end-user development

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

General Terms

Human Factors; Design.

INTRODUCTION

One of the key requirements for good user experience is user control and freedom during interaction [12, 13]. Knowing what to do when in trouble or after a change of mind is part of this control. Mechanisms to *undo* and *redo* actions are pointed out as efficient means for both error recovery and exploratory interaction. In Web browsing, this is typically framed as a case of *back navigation*, supported by the browser's own *back* button or a special Web application interface control specifically designed for this purpose (*e. g.* breadcrumbs). The implementation of 'undo' and *back navigation* has been extensively discussed in the literature for decades [1, 3, 4, 9, 14].

In this paper we discuss *return points* (associated with 'undo' and *back navigation*) in the context of mediated interaction supported by the Web Navigation Helper (WNH) [10], a user agent that allows for rephrasing, explanation, commentary and elaboration of interaction in scripted Web activities. With WNH, a user can indirectly communicate with Web applications by means of an interface interpreter with whom she interacts directly. Mediation dialogs carried out by WNH must have been previously created by savvy users, typically interested in helping or influencing other users' interaction with specific Web applications. Mediation dialogs must be linked to previously scripted Web tasks such as online shopping, for example. In Figure 1 we show a WNH dialog designed to help English-speaking users interact with the website of a popular Rio de Janeiro supermarket, whose interface is entirely presented in Portuguese. WNH is an extension to Firefox and interaction happens in the browser's sidebar (magnified in Figure 1 for sake of legibility).

In the currently implemented version of WNH, after proceeding to the next dialog (by clicking on the 'Continuar' button shown in Figure 1), the user cannot go back to the previous one. However, published user studies with WNH [2, 11] have clearly (and not surprisingly) shown the need to provide an 'undo' mechanism and *back navigation* support during mediated interaction. The challenge for scripted navigation underlying WNH is the same as in professional Web design: returning to a visited Web page may involve complex operations like logging in and out of an application, clearing and resubmitting previously filled up forms, and so on. There is no easy *automation* process to support back turns in mediated interaction created by non-professional end users. They must *design* such turns into the interaction that they are

about to create. For example, during the online shopping process previewed in Figure 1, as the benefitted user checks the content of her cart she may realize that she should buy an extra bottle of mineral water. How does she go back to the "Beverages" dialog? Unless the dialog creator has anticipated and designed this back turn in mediated interaction, the benefitted user is helpless (and interaction with the Web application breaks down).
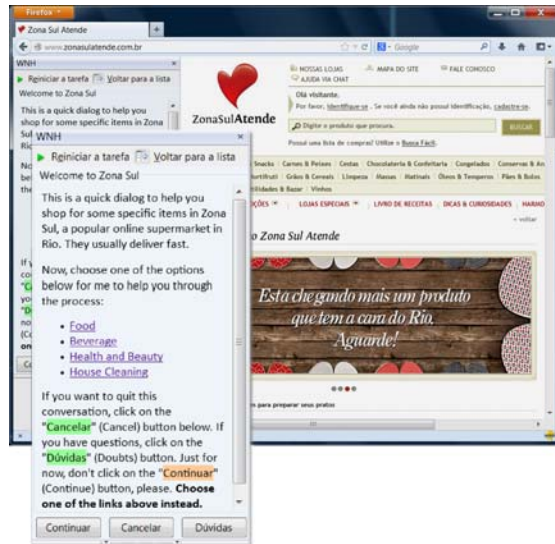

Figure 1. A snapshot of mediated interaction with WNH

The study reported in this paper is part of a long-term research on self-expression through software programming, which touches both on Human-Computer Interaction (HCI) and End User Development (EUD) issues. We are using WNH as an end-user design and programming environment that allows us to capture valuable evidence of self representation and intuitions about how to build software *proxies* for computer-mediated communication with other people (third parties). This time we invited volunteers to take part in a rich EUD experience. Their task was to configure *return points* in a Physics lesson built and delivered with WNH. Results indicate that their decisions about when, where, how and why to go back to previous points in the design of this kind of mediated conversation are very complex, no matter how savvy the users. Moreover, we collected powerful evidence of the participants' interpretation of what interaction through WNH is all about. While some viewed return configuration as a case of *back navigation* (a spatial perspective), others viewed it as *resuming a previous topic in conversation* (a temporal perspective). We believe that such evidence stands for their intuitions about HCI in a broader sense. The main contribution of the study is thus to illuminate aspects of EUD and HCI that haven't been discussed before, especially in conjunction with each other, and to propose questions for new kinds of investigation.

In the next two sections, we briefly describe WNH and introduce the problem investigated in our study. The fourth

section describes the details of the study, the methodology and results achieved with it. The last section discusses results in view of existing related work and points at future steps that we intend to make in the long-term project.

WEB NAVIGATION HELPER (WNH)

WNH is an agent that supports users while performing scripted tasks in the Web. It is implemented as an extension to Firefox, built on top of CoScripter [5] (a macro recorder embedded in WNH). Figure 2 depicts how WNH dialogs are combined with previously recorded macro steps in scripted Web navigation. Notice that WNH is *not* an 'on the fly' mediator that automatically generates dialogs to help users in whatever activity they choose to perform. Rather, it deploys previously specified dialogs intentionally designed to mediate a user's access to the activity whose steps have been defined in advance.
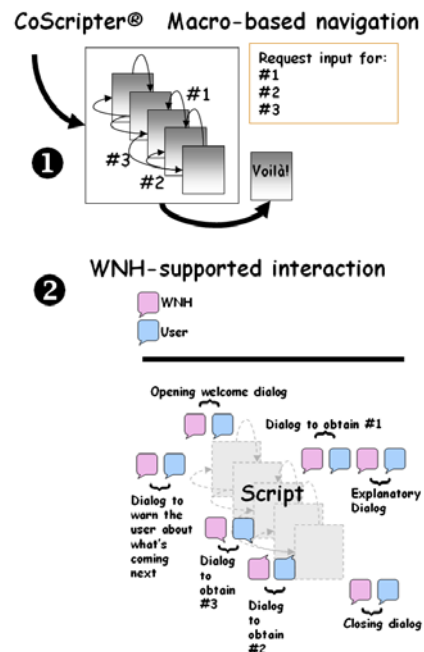

Figure 2. Scripts and dialogs and the relation between CoScripter and WNH

To illustrate how WNH works, we choose to detail the same teaching-learning test scenario that we have used in the study reported and discussed in subsequent sections. Suppose that some Physics teacher named Miguel collects Web pages and decides to organize and expose them to his students as part of an after-class lesson in Mechanics. He uses WNH to achieve his goal in two major steps. First, Miguel uses CoScripter to record the desired Web page navigation, which constitutes the content and organization of his lesson (see number 1 in Figure 2). This is a specification of the sequence of Web pages that Miguel's students will visit and of the kinds of interactions that they will be able to have as they run through Miguel's lesson. Since CoScripter can record on-page interactions like clicks and typing, for instance, on pages where students will be able to enter input and trigger backend computation, Miguel

captures all of the interactive steps that he finds necessary to communicate his teachings to students. Second, Miguel uses a specialized dialog editor to create mediation dialogs with which he wants to guide and teach his students as they follow through scripted lesson stages (see number 2 in Figure 2). An arbitrary number of dialogs can be displayed before script commands are executed. If commands require user input for execution, an important task for Miguel is to decide if such input will be automatically provided by the recorded script or if it will be interactively asked from the students during mediated interaction. The tight coupling between script commands and mediation dialogs is a key concept that dialog creators must master when using WNH.

The gist of Miguel's lesson is as follows:

1. Miguel welcomes the student (the benefitted WNH user), explains the goal of his lesson and gives her general instructions about how to use WNH.

2. Miguel takes the student to a website with explanations about Rectilinear Motion. He asks her to read carefully all content on the page being shown.

3. Miguel then takes the student to another website, where there is an online calculator to determine the values of different variables used in the Rectilinear Uniform Motion (RUM) formula.

4. Miguel explains to the student how to use the calculator, alternating RUM variables shown on the Web page on screen.

5. Miguel takes the student to another Web page showing various Physics exercises, from which he selects one that the student must solve.

The five "message parts" above involve 12 script commands captured by CoScripter and, coincidentally, 12 mediation dialogs designed by Miguel. Although the number of commands and dialogs is the same in this case, they are not in one-to-one correspondence with each other. Some commands don't have a corresponding dialog. For example, `click the "[calculate]" button` is an automatically executed step without requiring any user input. By the same token, there are dialogs that do not correspond to any underlying script command. For example, the very opening dialog is displayed before the first command script is executed. Likewise, the last closing dialog is displayed after the script has achieved its end. These are like *injected interaction* in the previously recorded script.

Figure 3 is a snapshot of the fifth dialog in the lesson, which explains (in Portuguese) the variable called "initial position" in the RUM formula. The text inside the dialog is translated into English in Frame 1. The student is supposed to inform the value of the initial position by typing it into the text box in the sidebar dialog. After clicking the "Continuar" (continue) button, the corresponding script command (`enter your 'initial position' into`

the `first textbox`) is executed. The informed value is then passed on to the corresponding input field in the Web page form (automatically highlighted on screen by CoScripter), which enables the next script step to execute.
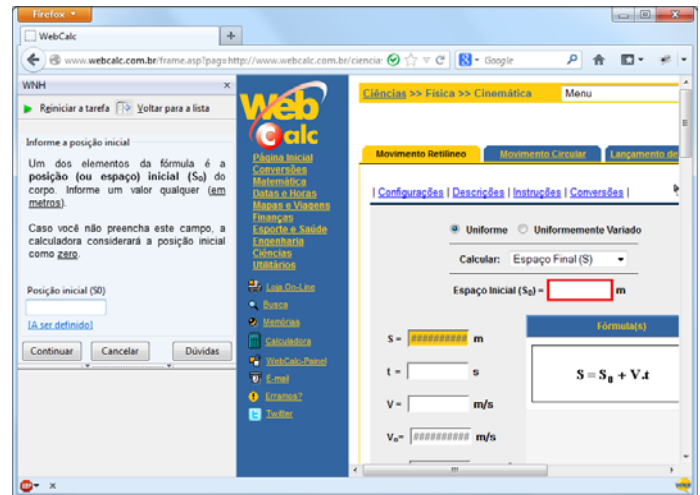

Figure 3. Physics lesson with WNH

Inform the initial position

One of the elements in the formula is the body's **initial position (or space) (S_0)**. Inform any arbitrary value (in meters).

In case you leave this field empty, the calculator will set the initial position to zero.

Initial position (S_0)

Frame 1. Dialog text

Figure 4 shows a diagrammatic representation of the Physics lesson structure, relating script commands (represented by numbered rectangles) with dialog turns (represented by numbered balloons) and backend application pages (represented by numbered "WWW" page icons). The diagram is disposed in four areas separated by dashed lines. In the first (top) area the absence of the "WWW" icon means that the user receives (and interacts with) the first dialog *before* script execution starts. The three remaining areas correspond to the three Web pages that Miguel wishes the students to visit (subject explanation, calculator and exercises) guided by the underlying recorded script. For each Web page, the diagram shows the corresponding dialogs and for each dialog it shows the corresponding script command(s). The relation between dialogs and commands means that the dialog is first displayed; then, as soon as the user clicks on the *continue* button two things happen, the associated command is executed and then the next dialog is displayed. For example, Figure 4 shows that D1 and D12 are not related to any underlying script command (the dialogs are displayed when nothing happens on the page). It also shows that commands C1 and C9 are not related to any dialog (these commands are automatically executed).
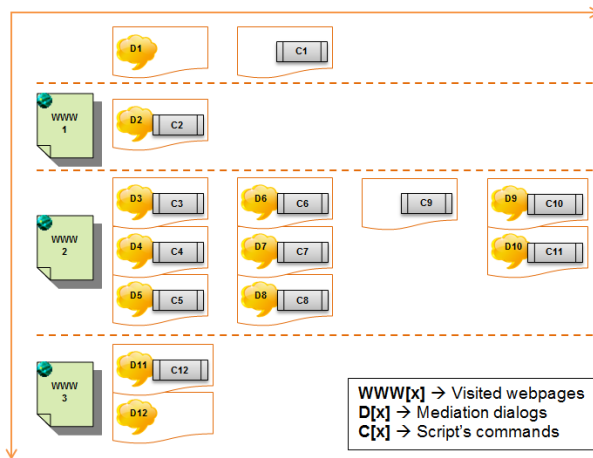
Figure 4. General lesson schema

BACK NAVIGATION IN WNH

As already mentioned, previously published user studies with WNH [2, 11] have pointed to the need of supporting back turns in mediation dialogs. In some of the reported cases, users realized that they had made mistakes while filling in dialog input fields and wished go back to correct them. In others, users wished to reread some instruction presented in a previous dialog, checking for potentially missed information. The only option for such users was to start over the whole task, an obviously severe usability problem with WNH whose solution depends not only on additional feature implementations by WNH developers, but also – and more interestingly for our research – on WNH users' ability to *design* mediation dialogs using the implemented features to benefit *their* targeted end users.

Implementation challenges are easy to anticipate. For example, take two sequential commands (transcribed in CoScripter code) from Miguel's lesson: "clip the '[answer div id]'" (C10 in Figure 4) and "go to '[exercise webpage URL]'" (C11). After the two commands are executed the student is on the exercise page (WWW 3 on Figure 4). Should the return mechanism simply take the user back to the previous command, 'playing' it back causes "clip the '[answer div id]'" to execute on the **wrong page**, leading to a script execution error. The opened page where the user is trying to 'clip' the value referred to by 'answer div id' is no longer available. To go back to the appropriate context of execution (the online calculator page), script commands that load the calculator page must also be executed, stepping through all the corresponding interactions (if any) that enable the *div* content to be correctly clipped. In other words, *going back* requires restoring past context, which must be reconstructed before a selected dialog passage (and its corresponding script step) is reintroduced.

Figure 4 depicts another important aspect of the parallel flow of commands and dialogs. Given a dialog like D9, for example, and its preceding dialog D8, it is not necessarily

the case that the script command corresponding to the one is the subsequent command corresponding to the other. In this example, there is an intermediate command C9 that is automatically executed in the script without any mediated communication taking place.

After carefully analyzing a wide range of *back turns* in interaction with WNH, we built a model of an embedded mechanism to help dialog creators manage the complexity of deciding when, where, how and why to go back to previous points in mediated conversation. The model, which was implemented and tested in a working low-fidelity prototype, offers the following kind of support for end user design and development with WNH.

Let D_x be the designation of any current dialog and D_{x-n} (for $n>0$) the designation of some dialog preceding D_x.

1. By default, the model supports the decision to **return from D_x to D_{x-1}**. The underlying script interpreter automatically restores the necessary context for executing the command associated with D_{x-1}. The effect of such back turn on the benefitted user screen, while WNH is acting as a user agent for mediated Web interaction, is that she sees one or more script steps being automatically executed without her intervention.

2. The model also supports the decision to **return from D_x to D_{x-n}**, the latter being **explicitly indicated by the dialog creator**. In this case, automatic script execution is turned-off and the information exchanged between benefitted user and WNH up to the return request point is retained. The effect of such back turn on the benefitted user screen is that she goes back to D_{x-n} and, from that point onwards, she revisits all intermediary dialogs comprised between D_{x-n} and D_x, confirming or correcting the retained information with which the WNH interpreter fills in dialog input fields. Unlike the situation in 1 above, here the benefitted user re-experiences all the mediated interaction herself, clicking and typing input in the process.

3. The model finally supports decisions to **prevent return from D_x**, that is, from canceling all return possibilities from a given dialog to the preceding ones. The effect of this decision on the benefitted user screen is the absence of any back turn control in the ongoing dialog. The user can only proceed to the next step (see button "Continuar" in Figure 3), request an explanation (see button "Dúvidas" – Doubts – in Figure 3), start the whole script execution over again (see button "Reiniciar a tarefa" – Restart task – on top of the sidebar in Figure 3), or cancel her interaction altogether (see button "Cancelar" in Figure 3).

The proposed model has a lot in common with solutions proposed by Cass and Fernandes [4], as well as by Abowd and Dix [1]. For example, automatic back turn handling entails the retraction of the most recent user action and iterated backward navigation through the history list [4]. It

also requires decisions of which one is a *safe* command to execute before the desired return point, for which we drew ideas from the "forward error recovery" approach [1] (which assumes "that the system has some goal which it is trying to achieve"). We have also used the notion of *cascading selective undo* [4], according to which "an undone action causes the undoing of other user actions until a meaningful state is reached" [4, p. 1]. The cascading selective undo algorithm is supposed to calculate what the authors call "the cascade", defined as "the set of tasks that must also be undone if the requested task is to be undone" [4, p.5]. The main difference in our case is that the "cascade" is not calculated during the "undo" action. It is done at the end of dialogs creation, as a result of explicit correspondences established by the dialog creator between (goal oriented) script steps and their corresponding dialogs.

Having prototyped and tested the functionality of our model, the next step was to find out how users would react to it, before a full-fledged implementation in WNH. Given that the benefitted WNH user's mediated interaction with Web applications necessarily depends on the quality of work performed by dialog creators, the right place to start a user study was with the latter. In the next section we report the study we carried out and the findings that came out of it.

EMPIRICAL STUDY

Our empirical user study was done with participants whose profile matched the targeted population of dialog creators envisaged for WNH. Their task in the study was to configure return points for an existing WNH conversation (Miguel's lesson described in the previous section). Our goal was to learn the participants' reactions to the task of designing return points in a WNH-mediated conversation in view of the features that we provided to support them.

Methodology

Given the novelty of the task, the tool and the technology involved in our research, we adopted a qualitative approach, which is especially appropriate in such cases [6, 8]. Our aim was to reveal and identify meanings that individuals ascribe to a complex set of factors involved in this particular context of end user design, expression and communication through software artifacts. The outcome of qualitative research is an integrated set of empirically-grounded meaningful evidence with which researchers articulate an interpretative framework. The framework can be used in subsequent steps of their own research (as well as in related research carried out by other researchers) in order to reach a deeper understanding of their object of investigation and to formulate further questions (or even test hypotheses) in more explicit and precise terms. We are thus not looking for generalizations, but rather for evidence of meaningful dimensions in our object of investigation. The value of such qualitative approach is to provide empirically-motivated foundations for the elaboration of hypotheses that can be tested for generalization in subsequent research steps.

Participants

We recruited seven experienced Internet users who agreed to participate voluntarily in our tests. An additional recruiting criterion was that they did not know of WNH. In Table 1 we summarize the profile of all seven participants.

	Gender	Age	Educational level / Occupation
P1	Male	30	PhD student in Computer Science / College professor.
P2	Female	28	Degree in Social Communication – Advertising / Press office assistant
P3	Male	29	PhD in Sociology / Journalist
P4	Female	36	Degree in Accounting Sciences / Housewife
P5	Female	26	PhD student in Computer Science
P6	Female	21	Undergraduate student in Social Communication – Advertising / Marketing trainee
P7	Male	20	Undergraduate student in Economy

Table 1. Participants' profile

Procedures and materials

The test with participants, carried out on an individual basis, involved five phases. 1) Participants were invited to watch a 10-minute video introducing WNH. The main goal of this video was to make the relation between dialogs and script commands very clear. 2) We presented a fictitious scenario to the participants. In it, they had a specific role to play: that of an experienced Internet user, worried about his or her godchild's school grades in Physics. The participant's character had a friend Miguel, who was a school teacher using WNH for extra-class activities with his students. Miguel provided the participant with an interesting lesson on RUM, built with WNH. The lesson could be used to help improve the fictitious godchild's understanding of some concepts in Physics (the lesson presented in the scenario was the same that illustrated the section about WNH). 3) We stepped through the entire RUM lesson, explaining each and every step in it to the participants. We also showed them, very clearly, the problem to be solved: Miguel's dialogs did not allow the student to go back to previous points in the lesson. The return points in all dialogs had not been set when Miguel sent the participant a copy of his lesson. 4) Participants were therefore asked to set the return points by filling in an electronic form with structure shown in Table 2. For each dialog in the lesson, the participant should answer all the appropriate questions. 5) After the task was accomplished, we interviewed the participants.

Although the study was carried out within an educational context, we should clarify that we are *not* evaluating the lesson's pedagogical content or adequacy. The context was chosen only for its convenience, since most adults are familiar with explaining a school lesson topic to a child. We thus assumed that they would have less trouble in relating to the proposed communicative strategy adopted by the fictitious character of the test scenario (Miguel).

The back turn interface control to communicate choices made by participants was a link – "A ser definido" ("To be defined") – right above the "Continuar button" in the sidebar dialog (see Figure 3). However, participants were free to suggest other interface controls instead of links.

1	From this dialog, should the user be allowed to go back to a previous point?
2a	If your answer is "no", please justify.
2b	If your answer is "yes", please determine to which dialog (s) the user should be able to go back to.
2b-1	What are the reasons why the user might want to go back to the points indicated in 2b?
2b-2	Please, justify your answer to 2b-1.

Table 2. Configuration form

While answering question 2b-1, participants could choose one or more from a set of expressions that we provided to represent the users' situation if they needed to go back to a previous dialog. If none of the provided expressions seemed fit, participants could again suggest their own expressions. Our proposed set of expressions was:

- Oops! I made a mistake: The user realizes that he/she made a mistake and wants to go back and correct it.
- Let me see it again…: The user wants to review a previous dialog's content, for example in the course of a single explanation extending over multiple dialogs.
- Let me do it again: The user successfully reaches the goal of some dialog sequence, but wants to redo one or more steps from previous dialogs.
- Let me check what I've done: The user wants check information that he/she provided (or some other action or decision he/she made), before proceeding into subsequent interactive steps.
- What if…: The user wants to explore an alternative path of interaction.

The expressions used in our set correspond to previously observed usability issues with WNH, caused by the absence of a back turn control mechanism. We do not claim, however, that the set should cover all the possibilities for back turns. In fact, as will be shown below, participants thought of new and insightful possibilities that had not occurred to us before.

Our evidence came from the following sources: filled up Web forms with specified return points; audio recordings of participants' activities; screen captures of all of the participants' activity in the computer (form filling, use of the browser, etc.); recorded post-test interviews about the user experience during the test; field notes, with anything that caught the observer's attention.

RESULTS
Evidence collected in the study was much richer and more diverse than preceding studies have suggested. Our interpretive analysis of the data is organized into six meaningful categories.
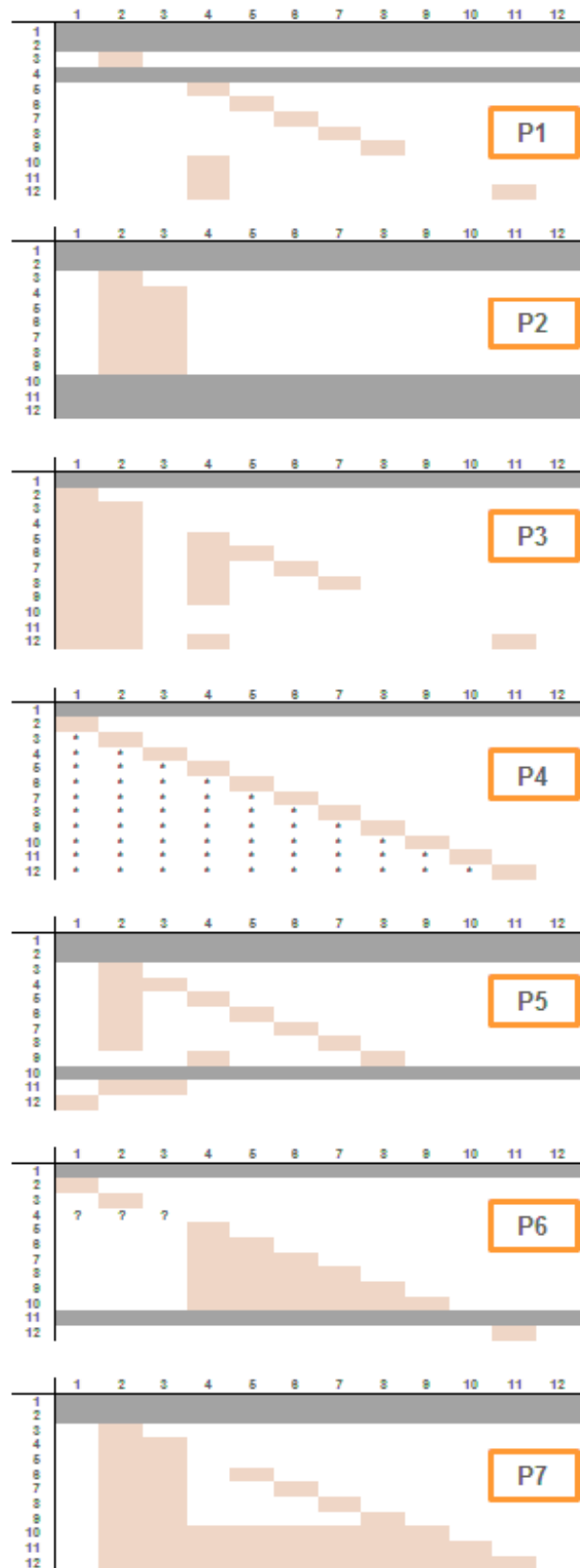
Figure 5. Participants' return points

I. Return path structures

Miguel's lesson involved three different Web pages and twelve mediation dialogs (see Figure 4). Most dialogs (D3-D10) refer to a single Web page (WWW 2, the calculator). Figure 5 shows seven diagrams representing return point configurations done by each participant. Line numbers (1-12) represent the *current* (source) dialog. Column numbers (1-12) represent the *return point* (target) dialog. Light-colored cells mean that the participant decided that the dialog indicated by the cell *line* number should provide a back turn option targeted at the dialog indicated by the cell *column* number. For example, in P1's diagram, we can see that the light-colored cell on line 3, column 2, says that D3 (line 3) provides a back turn to D2 (column 2). Likewise, D6 (line 6) provides a back turn to D5 (column 5). Dark-colored cells indicate that the dialog does not allow the user to return to a previous dialog. For example, in P1's diagram, dialogs 1, 2 and 4 do not support back turns.

The stars in P4's diagram mean that, in this participant's view, all previous dialogs could potentially be reached from any dialog along the way. This feature will be further discussed in subsequent categories. Question marks in P6's diagrams mean that this participant did not clearly indicate the return point(s) for D4.

Figure 5 helps us see the differences and similarities among participants' designs. Regarding the similarities, notice that D2 is a recurring return point for *more than one dialog* (see P2, P3, P5, P7 diagrams). D2 introduces the Web page with a general explanation about RUM suggesting that a content-based criterion was very strong in this case. All the participants that chose this return point selected the expression "Let me see it again…" to represent the student's motivation to go back to D2. P4 is not on the list but she affirmed many times that it was important to be able to go back to D2 and this was her motivation to propose her solution described in details ahead (a confirmation of the content-based criterion). D2 is thus a kind of "hub dialog", accessible from multiple follow-up dialogs. This feature was observed with P2 (hubs in D2 and D3), P3 (hubs in D1, D2 and D4), P5 (hub in D2), P6 (hub in D4) and P7 (hubs in D2 and D3). Of course, as dialogs unfold, *hubs* become more ambiguous. For example, if there are only one or two subsequent dialogs to a potential hub point, is the returning from one or the other a sign that we have a hub case at hand? Or is it just a 'return to previous' case? Regardless of the answer, the meaningful information is the fact that *not all of the participants* decided that D2 or D4, for example, were hub dialogs. Why did P1 not see a hub in D2 nor did P2 not see a hub in D4?

Figure 5 also shows that six out of seven participants set return points for dialogs involving the calculation of the RUM formula (D4 to D8) in step-wise ("stairs-shape" visualization) form. Each dialog returns to the one that immediately precedes it. In these cases, the user would go back to correct a mistake or to check given information.

Participants chose mainly the expressions "Oops! I made a mistake" and "Let me check what I've done" as a representation of why the user would wish to go back to previous dialogs in this context.

Three participants (P2, P5 and P6) prohibited the return from the dialogs 10 and/or 11. Dialog 10 introduces the practical phase of the lesson and Dialog 11 presents the problem to be solved. These participants thought that the student had already accomplished the learning and training phases for RUM and how to use the calculator. The next step should be a *test* (self assessment), as shown in P2's explanation during interview: "It's like [my godchild] had been given all the opportunities to consult [the content] and now that he has already learned the content, that he has read a whole lot [about RUM]; he will try to solve the problem without needing to go back to previous stages. Thus he can focus and see that from that point onwards he must [only] apply what he managed to learn". Additionally, although P3 indicated that dialogs D10 and D11 had special return points, the justification written in his form is: "Because a new stage begins here [in D10]. It seems to me that the only sensible return cases are dialogs 1 and 2, which explain how the lesson works and what the subject is". In other words, he sees the same discontinuity as the other participants who denied the return from D10/D11and clearly manifests his opinion that dialogs 1 and 2 are hubs which subsequent dialogs must point to.

Compared with the model features we had first thought of, results in this category show that only P4's configuration matched the automatic return point configuration mechanism we proposed. All other participants chose to set their own alternative for return points. Also, the majority of participants had at least one dialog for which they prevented back turns altogether. Possibly the most striking result in the comparison is that we *did not* think of supporting returns to multiple dialogs. All but one of the participants (P4), however, felt the need to provide multiple options for at least one of the dialogs.

II. Interface controls

As a natural consequence from different return path structures, participants thought of different interface elements for users to express their return choices. The ones that were explicitly mentioned by participants are:

- Links: Participants followed our suggestion of including a link indicating the return point. P2, P3 and P5 said that the positioning and the textual content of the link on the page should be left for dialog creators to decide.
- List boxes: P6 and P7 suggested that dialog creators should build a list box with all possible return points from a given dialog. The user would pick the desired one. P6 said that the identifier of the dialog in the list should be the dialog's title, whereas P7 suggested that besides the dialog's title there should be a brief explanation about the dialog itself.

- Breadcrumbs: During the interview, P1 elaborated on his previous decisions and concluded that it would be better if the student had access to *all* previous dialogs, being able to choose freely which one he needed to go back to (which means that P1's diagram in Figure 5 does not represent his final opinion). P1 suggested to that the list of previous dialogs should be shown in breadcrumb style (a sequence of visited links in Web pages).

- Back and forward buttons: P1 suggested the inclusion of an arrow-shaped back button to the preceding dialog and the replacement of the "Continuar" button by an arrow-shaped forward button. P4 and P7 also indicated that they would like to see an arrow-shaped back button.

- Search text box: P4 thought of a completely different solution compared to the others. She proposed that the student should type into a text box some text corresponding to what he was looking for. A keyword search would thus lead him to dialogs about the desired content. To do this end, the dialog creator should indicate two or three keyword summarizing the topic of each dialog. This is why there are the stars in P4's diagram in Figure 5. All dialogs should be reached from any subsequent one, suffice it that the student wrote the correct keywords in the search box. The problem is, of course, that an extra mechanism to prevent inadvertent *forward* navigation should be in place as well.

III. Going forward from a back turn

In some cases, participants wondered about what should happen after the student went back to a previous dialog. Some were positive that the student should go through all the follow-up dialogs again, while others suggested the use of "jumps to the future". P1 said: "If links are disposed like *breadcrumbs*, the student can only go backwards [and not forwards]. […] In case I go back to some previous dialogs directly, if I want to get to where I was before, the best is going step by step till there". Likewise, P4 (who had thought about the "search box") suggested the use of a forward button: "[By pressing this button he will see] the next [dialog], always like that, in a sequence". P3, however, said that: "[the student] should go [then] to the last point, where he came from […].For example, if I came back from [dialog] 10 to [dialog] 2, from here [2] I should go next to 10". P5 had a different opinion: "He would go back to some previous dialog and then again to the one that called it. […] I would want change the initial position [variable], but keep the remaining variables the same. So, I would not like to have to go through [the whole thing] over again". However, P5 made tradeoffs clear: "The dialog creator might use this [possibility] in the wrong way, though. So I am not sure if this is so interesting […]. Maybe interesting, but not mandatory or essential."

IV. The learner's reasons to go back

To answer question 2b-1 (see Table 2), participants could choose one or more expressions in the provided set of reasons or else suggest their own expressions, with new reasons. Table 3 summarizes the use of expressions for each participant. The numbers in columns indicate how many times the participant has used the expression in the corresponding line as a reason to go back.

Why going back?	P1	P2	P3	P4	P5	P6	P7	Total
Oops! I made a mistake	8		5	1	5	1	7	27
Let me see it again…	4	7	4	8	7	10	10	50
Let me do it again	7		2		2	1	2	14
Let me check what I've done	2	1	1	9	5	7	10	35
What if…			1					1
I just changed my mind			4					4
To practice						1		1
To test						3		3
To use the calculator					1			1

Table 3. Occurrences of reasons to each participant

We can see that all participants unanimously chose "Let me see it again" as the reason for at least four back turn cases. P6 and P7 chose it for 10 out 12 dialogs. "Let me check what I've done" and "Oops! I made a mistake" were the second and third most frequently invoked expressions and reasons. Notice that all but one participant (P2) used the latter. During the interview, we asked P2 if she had thought of users making mistakes. Her answer: "No, I haven't. I actually thought of the wrong answer [in the calculation] […] but there are all these explanations here on the side." A single case of "What if" appeared in P3's D9 (a dialog showing the answer for the calculation). He said: "All [of these] are reasons for him to start over". Notice (in Figure 5) that P3 set D9 to go back to D4, which is when the calculator page is loaded.

Three participants suggested new expressions and reasons to go back. P3 created "I changed my mind" and used it for back turns in dialogs D5, D6, D7 and D8. Notice (in Figure 5) that all of them give access to D4, which is the dialog where the user chooses what he wants to calculate (position, time or speed). P3's idea is that before the user finishes some calculation he started, "he changes his mind" and picks something else. The other three additional reasons are similar cases. The user supposedly wants to go back to using the calculator in order to test something or to practice what he has learned.

V. Reflection and personal positioning

Both during form filling and interview, we observed some variation in the reflection process carried out by participants. After some thought, P2, P3 and P5 came up with the logic for the lesson's dialogs and then applied it throughout the whole form. P4 changed her mind while filling up the form, and then she applied her new ideas to the lesson. P1 and P7, on the other hand, changed their minds during the interview but did not apply the changes to the lesson. Then we have what P1 and P7 wrote in the form and what they proposed during the interview.

Regarding personal positioning, P2 dived deeply into the proposed scenario. For example, when she started to fill in the form she asked: "Am I supposed to use the character's name or I may just say 'student'?" She pondered carefully about each return point, checking all possibilities. However, as soon as she decided on the logic of the lesson, she just used the same line of reasoning in the remainder of her task. She explained in detail all of her decisions, like when she said: "When he completes this part [of the lesson] and manages to go through all stages till he has the [calculated] answer, he starts the practical part [of the lesson]. I think he must have already apprehended [the content] and will then do [the exercise] for real". This evidence shows how P2 was vividly engaged in the conversation, making design decisions as if she was the actual creator of the lesson communicated through WNH.

At the opposite end there was P4, who was seeking for the most generic solution he could think of. Maybe one motivated by the Physics lesson challenge, but that would be applicable to *all* kinds of lessons. In fact, the solution proposed in P4's diagram (see Figure 5) could be replicated in virtually any 12-dialog conversation with WNH. This is remarkably different from solutions provided by other participants, whose solutions, even with some "generic" parts, included return strategies motivated by their situated positioning in the specific context of the study.

Intensive reflection on decisions (with subsequent changes of mind, which could or could not be implemented in the design stage of the study) was a natural consequence of the procedures we adopted. As soon as we asked participants to *justify* their choices, we triggered a reasoning process that extended over the interview, till the end of the experiment. In and of itself, the various revisions we observed along the way show the complexity of designing back turns in mediated conversation. A potentially more interesting result, however, is how deeply the participants were immersed in the *situation* presented in the scenario. Their personal positioning with respect to the task at hand varied from very close (P2's asking whether she should address the student by his name or not) to very far (P4's seeking for the most generic back turn design solution), with most participants positioning themselves *in between* the two extremes. They were *somewhat* engaged, yet keeping an eye on a higher level *logic* that they could apply regardless of the specific situation where the characters in the scenario were facing. This can be seen in evidence provided by P6, who thought of including list boxes with many back turn options: "I think he must be able to choose [where to go back to]." When asked if she had had any doubts while determining the return points, she answered: "No, because I did that thing [use of lists]. I think the student needs to have the possibility to choose [for himself]."

VI. The meaning of back turns
The analysis of collected evidence allowed us to identify three kinds of meaning assigned to back turns.

- Web-like navigation: Some participants viewed interaction with WNH as comparable to navigating from one Web page to another. The user should be able to go back and forth as if using a regular browser. This was apparent in P1's idea of using *breadcrumbs*, as well as in P4's and P7's suggestions that there should be arrow-shaped buttons in the dialogs to indicate back turn options. P4, in particular, explicitly suggested that it should be possible to go *backwards and* (then) *forwards* in mediated communication with WNH. Moreover, P3 provided one of the strongest pieces of evidence that back turns are like Web navigation moves: "[WNH] brings up this idea of non-linearity, basically [it establishes] a relation of hipertextuality among things."

- Conversation controlled by the dialog creator's intent: Some participants viewed interaction with WNH as conversation, where it is always possible to reintroduce past topics. Participants who deliberately established specific return points and blocked others clearly followed this view, imposing their conversational intent as the underlying logic for conversational turns. The most prominent evidence was provided by P2. She chose only a few, yet enough (in her view) topics to be reintroduced in the conversation with the student. Her dialog structure (see Figure 5) reads almost directly as the equivalent of the following message to her interlocutor: "If you don't understand or remember what I say, I will only repeat what I said in D2 and in D3."

- Conversation controlled by the benefitted user's intent: Among the group of participants who viewed interaction with WNH as conversation, there were some who – unlike P2, for example – decided to hand the control for reintroducing past topics of conversation over to the benefitted user. This is the case of P4's search-based return points. The very use of *keywords* to control topic in conversation shows how strongly P4's solution was influenced by a natural language conversation metaphor.

DISCUSSION AND CONCLUSIONS
The results of our study are significant in a number of ways. First of all, they address one of the least investigated aspects in EUD, namely end user *interaction design*. According to the authors of a recently published comprehensive survey of the state of the art in EUD [15], the essential purpose of end user programming is to create "a program primarily for personal, rather public use." End users, they say, "are usually programming for themselves or for a friend or colleague". One of the consequences of this radical personalistic perspective is that "[the program] users' requirements are able to be implicit, and perhaps not even consciously recognized."

Our study with WNH provides a highly contrastive situation for typical EUD studies in that the *program* that is being constructed by end users is: specifically designed for the benefit of some *other* user; and specifically designed

around and for the purpose of facilitating (mediated) *human-computer interaction*. It presents a number of research topics that have not received much attention from EUD researchers. We believe that all categories emerging from our analysis of the collected data constitute, in themselves, an interesting topic for further research.

The second reason for the significance of our results is that, as shown by quotes from [15], above, compared to professional programming, end user programming is much more about *self* than about *others*. Therefore, the context where our study was carried out was particularly fit for observing self expression and the communication of values, beliefs and attitudes through software programming. Given that the result of such programming was in essence a *mediating interface* to support *other users' interaction* with Web sites and systems, the decisions expressed by participants can be legitimately taken as a sign of their own intuitions about HCI in general and HCI design in particular. For example, the contrast between a navigational perspective on back turns and a conversational one points to the participant's own conception of how interaction is (or should be) designed for the benefit of the targeted user(s). In particular, different views on conversation being dominated by the dialog creator's intent or by the benefitted user's, accompanied by the rationale that justifies them, show how the participants conceive of interaction designed and developed by professionals. An interesting line of investigation to pursue is to find out how such conceptions relate to the participants' experience as *users* of professionally designed software. In other words, are they – as designers – viewing the benefitted users of the dialogs they create as a reflection of themselves?

The third reason for the significance of our results addresses the main object of investigation of Semiotic Engineering [16], namely the computer-mediated communication from HCI designers to the users of the artifact that they are about to build. This process, called *metacommunication*, has been mainly treated in semiotically inspired HCI research as a theoretical construct. However, the results of this study provide powerful evidence of the *ease* with which end users naturally engaged in metacommunication design tasks, covering a vast spectrum of possibilities by simply projecting their intuitions about what HCI is all about.

Our next steps in this research are to improve the design and usability of WNH and to use it as an investigative tool in further studies about self expression in end user metacommunication via software. Besides, since this qualitative study did not try to predict how end users encode the meanings of back navigation in mediated communication designs, it would be valuable carry out further research with quantitative methods to predict the users' design behavior with respect to one or more of the meaningful categories indicated by our current results.

REFERENCES
1. Abowd, G. D., & Dix, A. J. Giving undo attention. *Interacting with Computers*, (1992), 4(3), 317–342.

2. Alves, A. S. *Estudo do uso de diálogos de mediação para melhorar a interação de surdos bilíngues na Web*. Rio de Janeiro: Universidade Federal do Estado do Rio de Janeiro. 156 p. MSc Dissertation. (2012)

3. Berlage, T. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Transactions on Computer-Human Interaction (TOCHI)*, (1994), 1(3), 269–294.

4. Cass, A., & Fernandes, C. Using task models for cascading selective undo. *Task Models and Diagrams for Users Interface.* Springer (2007). 186-201.

5. CoScripter. http://coscripter.researchlabs.ibm.com/

6. Creswell, J. W. *Research design: qualitative, quantitative, and mixed methods approach*. 3rd. ed. Thousand Oaks, CA: SAGE, 2009.

7. Kaasten, S., & Greenberg, S. Integrating back, history and bookmarks in Web browsers. In *Extended Abstracts (CHI EA '01)*. ACM, New York, NY, USA, 379-380.

8. Lazar, J.; Feng, J. H. et al, H. *Research methods in human-computer interaction*. New York: Wiley, 2010.

9. Milic-Frayling, N., Jones, R., & Rodden, K. Smartback: supporting users in back navigation. *Proc. WWW 2004*, 63–71.

10. Monteiro, I. T. *Acessibilidade por Diálogos de Mediação: Desenvolvimento e Avaliação de um Assistente de Navegação para a Web*. Masters dissertation. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2011.

11. Monteiro, I. T.; De Souza, C. S. Embedded Cultural Features in the Design of an Accessibility Agent for the Web. In *Proc.UAHCI'11*, Springer-Verlag, Berlin, Heidelberg, 295-304.

12. Nielsen, J. *Usability engineering*. Morgan Kaufmann, 1994.

13. Sharp, H., Rogers, Y., & Preece, J. *Interaction design: beyond human-computer interaction*. Wiley, 2002.

14. Yang, Y. Undo support models. *International Journal of Man-Machine Studies*, (1988), 28(5), 457–481.

15. Ko, Andrew J.; Abraham, Robin et al. (2011) The State of the Art in End-User Software Engineering. ACM Comput. Surv. 43, 3, Article 21 (April 2011), 44 p.

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