Investigating the Case-Based Reasoning Process During HCI Design

José Antonio Gonçalves Motta
PUC-Rio
Rio de Janeiro, RJ - Brasil
jmotta@tecgraf.puc-rio.br

Simone Diniz Junqueira Barbosa
Departamento de Informática, PUC-Rio
Rio de Janeiro, RJ - Brasil
simone@inf.puc-rio.br

***Abstract*—Given that the design activity makes use of previous design knowledge, we turned to case-based reasoning (CBR) to help identify opportunities to support the design of human-computer interaction (HCI). Using interviews with professional designers and Semiotic Engineering, we developed a CBR tool called CHIDeK (Computer-Human Interaction Design Knowledge), with which we conducted a study to observe how it influenced the HCI design activity. We found that the cases recorded in CHIDeK supported design by motivating the designers' reflective process, triggering their memories of experiences with similar systems, and helping to generate new ideas. We have also identified limitations in our case representation and case access methods, which offer opportunities for further research.**

Keywords—HCI design; case-based reasoning; design knowledge; design tools; human-computer interaction; semiotic engineering

I. INTRODUCTION

According to [1], it is a well-known fact that the design process relies on knowledge from previous experiences, which are adapted by the designer to solve new problems. In other words, we can usually trace or link many of the ideas designers develop to solve new problems to previous design knowledge.

Design knowledge is usually concerned with how things might be, rather than how things are [2]. However, a designer cannot apply existing design knowledge directly, as if it were a rule or an equation. Additionally, the designer can only acquire design knowledge by designing, since each design problem is unique [2]. Schön reinforces this point when he states that, while the designer is interacting with a representation of his design and is reflecting on it, he is able to learn more about what he is designing while designing it [3].

We have investigated how human-computer interaction (HCI) designers access and reuse previous HCI design knowledge adopting a case-based reasoning (CBR) approach [4] with the support of a CBR tool. Goel and Craw state that CBR is a useful paradigm to apply to the design task [5]. Based on their work, we have explored our assumption that a well-designed CBR tool can help solve HCI design problems.

We divided our work into three stages. The first stage consisted of interviews with professional designers to investigate the expected usefulness of a CBR tool to support design and, if the expectations were positive, to collect requirements for such a tool. We gathered information on the use of existing knowledge in the design activity, the designers' feelings towards the CBR process in their work, and unanticipated requirements and constraints for the tool. In the second stage, we used Semiotic Engineering to analyze the data obtained from the interviews and developed a CBR tool called CHIDeK (Computer-Human Interaction Design Knowledge). Finally, we conducted a study with HCI students, randomly assigning them to two groups: an experimental group and a control group. We gave each student an HCI design problem to solve individually. Students in the experimental group could use CHIDeK, whereas students in the control group could use any other tool available (e.g., internet searches). We then analyzed what happened during the design activity in both situations. Due to lack of space, more details on the methodology and artifacts used in the study can be found in [6].

This document is organized as follows. We discuss the theoretical background that grounded this work: CBR, in Section II, and Semiotic Engineering, in Section III. In Section IV, we describe the preliminary interviews conducted with professional interaction designers and the metacommunication message, as described in [7], we built using the requirements we got from those interviews. In Section V, we describe CHIDeK, the CBR tool we developed based on our metacommunication message. In Section VI, we discuss the study we conducted to investigate the design activity with and without CHIDeK. Section VII details the results of the study. In Section VIII, we contrast our metacommunication message with the messages we assumed our study participants received when interacting with CHIDeK. Section IX discusses the results we got. Finally, in Section X, we wrap up this work with our conclusions and discuss opportunities for future work.

II. CASE-BASED REASONING

Case-based reasoning (CBR), shown in Fig. 1, is a process for exploring the potential use of cases to support problem solving.

The process consists of *retrieving* appropriate cases from a case library, *adapting* the chosen cases according to the problem context, creating a solution, *evaluating* whether the new solution is adequate, and *storing* the new case in the library [8].

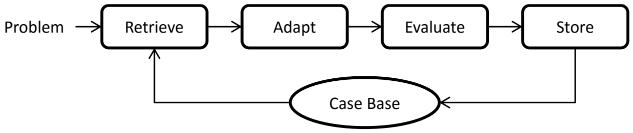


Fig. 1. The Case-based Reasoning cycle.

Despite existing research on CBR in HCI, to our knowledge the practical use of a CBR tool for interaction design has not been investigated. Sambasiva et al. developed a CBR system for interface design called AskJef [9]. Its goals were to help the designer understand the problem he had to solve and to offer an array of past solutions. The designer had to analyze the solutions given by the system and adapt them to solve the new problem. AskJef had three ways to represent a case: graphics, audio and alphanumeric symbols. The designed product, AskJef, had its usability evaluated by experts using guidelines and usability testing techniques, but the authors did not discuss its influence on the interaction design process itself.

Kim and Yoon developed a CBR system for user interface (UI) design, called MCBDS (Multi-level Case-Based Design Support), which focused on the case representation to aid the designer's cognitive process [10]. The cases dealt by this system were UI tasks represented in three abstraction levels: *what the task is about*, *why the task should be executed* and *how the task will be executed* using the system. MCBDS left the tasks of adapting and evaluating the solution to the designer. They described a usage scenario of the system, but did not report on any empirical study with it.

III. Semiotic Engineering

Semiotic Engineering is an HCI theory that views human-computer interaction as an act of communication between two humans (the system designer and the user) mediated by a computer (the system itself) [7]. The designer uses the system to communicate to the user his vision about the user, the user's goals and how to achieve them. The user, in turn, receives the designer's message while interacting with the system and uses what he interprets from the message to perform his task. This communication between the user and the designer mediated by the system interface (the *designer's deputy*, according to Semiotic Engineering), is called *metacommunication*, which can be paraphrased as follows [7]:

"Here is my understanding of who you are, what I've learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision."

For the metacommunication to take place, the designer can use a variety of representations, or *signs*, which the user must interpret. A sign is anything that conveys some meaning to someone [11]. Semiotic Engineering describes three kinds of metacommunication signs:

- **Static signs**: signs that convey meaning without the influence of time. Usually, these signs can be interpreted using a single snapshot of the user interface;
- **Dynamic signs**: signs that depend on time to convey meaning. Usually, these signs can be interpreted by a sequence of images or a video depicting an unfolding interaction;
- **Metalinguistic signs**: signs whose purpose is to explain other signs. They can be either static or dynamic. The content of a help file is an example of a metalinguistic sign.

In the next section, we show how we used the metacomunication message to analyze the requirements obtained from the interviews. In Section VII, we also use Semiotic Engineering concepts to analyze the results we obtained from the study we conducted.

IV. Preliminary Interviews

This section reports on the interviews conducted at the first stage of our work, before developing CHIDeK. To carry out this study, we conducted semi-structured interviews with eight people who have professional experience in HCI design.

The study consisted of a pre-presentation interview to identify the participants' profile, a presentation about CBR and a post-presentation interview to discuss the CBR process and the envisaged creation of a CBR tool. The CBR presentation consisted of an explanation of the definition of cases, the CBR cycle and a scenario in which a CBR tool is used during the design of a mobile system. All interviews were recorded and transcribed, with the participants' consent.

We asked the following questions in the post-presentation interviews:

a) What do you think of CBR?

b) Do you see any relation with how you work? Which ones/Why not?

c) Can a CBR tool contribute to your work? How? What should it have to be as useful as possible?

d) How can this tool get in the way? What shouldn't it have?

e) What's your opinion about the introduction of a CBR tool to support the design in your work? How long (in terms of number of projects and months) would you invest to learn to use it?

We analyzed the transcripts using a technique called Template Analysis [12], a thematic analysis technique [13] that involves the development of an initial coding template, which evolves iteratively and is used as a means to analyze textual data. In our case, the textual data were the transcripts of the interviews.

As a first step in our analysis, we coded the interviews with the following question in mind: What do a CBR tool and its cases need to be useful for HCI design? Table I shows the coding template we generated after analyzing the data and the corresponding requirements we derived from the themes.

Using the requirements listed in the table, we created the first part of the metacommunication message of CHIDeK. To do that, we used the metacommunication message template and replaced its first sentence (in boldface, below) with the information we have collected during the interviews.

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

The message, after the replacements, is detailed below. We linked some of its parts with the themes listed in Table I to inform the source of its content.

“You are an HCI designer who works with a variety of artifacts, such as wireframes, prototypes and documents, (A.1) and deals with multiple domains (A.2a) and restrictions (A.2b). To make your job easier, you turn to examples to find out what competitors are doing (B.1), to reuse previous solutions (B.2), or to talk about design ideas with other stakeholders (B.3). Also, you like working with images (C.2). Depending on the situation, you can adopt either a problem-based or a solution-based strategy (C.3). You would like to have a library where you store existing design knowledge (C.1). To be useful, the description of that knowledge must explain the problem that was to be solved, the developed solution and its evaluation (D.1). You want your library to have lots of cases, but you don’t want redundancy (D.2). Since you often deal with code, cases should provide access to source code for a solution (D.3). You don’t have much time to spare, so the tool must speed up your progress (E.3). Therefore, you don’t want to spend a long time to grasp the idea behind a case (D.2). In the same way, you don’t want to spend a long time looking for an interesting case (E.1).”

TABLE I. CODING TEMPLATE CREATED AFTER ANALYZING THE INTERVIEWS, INCLUDING THE REQUIREMENTS ASSOCIATED WITH EACH THEME.

First-level themes	Second-level themes	Requirement
A) Example features	A.1) Work material	Cases should be able to deal with design artifacts, such as wireframes.
	A.2) Example contents	a) Cases should be flexible enough to describe experiences from a variety of domains. b) Cases should also explain the restrictions the designer has, such as technology limitations.
B) Example uses	B.1) Competitive advantage	Cases should be able to help the designer know more about the state-of-the-art of systems related to a domain.
	B.2) Reuse of past solutions	Cases should ease the reuse of previous design solutions.
	B.3) Design communication	Cases should be useful to help communication between various stakeholders.
C) Designer’s practice	C.1) Usage of an example library	The tool should be able to serve as a library of examples (cases).
	C.2) Affinity with images	The tool should integrate images.
	C.3) Design strategy	Cases should be amenable to problem-based or solution-based design strategies.
D) Case library requirements	D.1) Case content and structure	Cases should describe problem, solution and evaluation.
	D.2) Conciseness	The set of cases should avoid redundancy and cases themselves must be succinct.
	D.3) Must contain code	Cases should contain some sort of source code.
E) CBR tool requirements	E.1) Efficient access to cases	The tool should provide an efficient way to find cases.
	E.2) Integration with other tools	The tool should have some sort of integration with other tools to fit into the designer’s usual workflow.
	E.3) Not time-consuming	The tool should not make the designer spend a long time using it.

As we designed and developed CHIDeK, the second part of the tool’s metacommunication message was incrementally built and refined. Doing so helped us not to deviate from the requirements we discovered previously, because in order to create the second part of the message, we had to pay attention to the first part to keep the whole metacommunication consistent. So, in this section we report how we elaborated the second part of the metacommunication template (in boldface, below), completing the metacommunication message.

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

Just like before, we used the themes from Table I to explain which requirement was addressed in each part of the metacommunication message.

"To help you in your work, I developed a case library called CHIDeK, where you can store past design knowledge in the form of cases, which you can then use to solve new problems (C.1). When you create your case, you must give it a title and a description. To create the description, I provide you with a rich text field with many formatting options and other functionalities, such as adding links or images (C.2). I opted for this because it gives you freedom to describe your case regardless of the domain (A.2a) or other restrictions (A.2b). In order to create a good case, I suggest you answer a few questions regarding the problem you needed to solve, how you solved it and how the solution fit the problem (D.1). However, you are free to ignore the suggestions and describe the case in any way you want (C.3). You can also attach any kind of file to your case. For instance, if you developed a prototype to solve a design problem, you can attach the source code to the case, in the event you want to reuse it later (A.1, B.2, D.3). After that, you must select the tags that describe the case. I have already created many tags of different types for you. There are tags focusing on the problem you need to solve, such as the action tags, and focusing on the solution you might already have in mind, such as the widgets and UI patterns tags (C.3). If you do not find any tag that suits you, you can create your own tags. However, you cannot create duplicate tags. Also, if you create a tag that is similar to an existing one, I will warn you so that you can be wary of any redundancy that might occur if the tag is created (D.2). In addition, if you create a case which is tagged the same way (or almost the same way) as another tag, I will warn you, so that you can check whether both cases are similar (D.2). You can also specify how the new case is related to other cases in the database. To do so, you can create a list of related cases. For each related case, you can specify the kind of relation it has to the new case. To populate the related case list, I give you some options to speed up the search such as using the new case tags as a filter or searching only for cases you created (E.3).

The case library shows the title of the cases and a small thumbnail to help you have an idea of what the case is about (C.2). To find a case, you can browse the library, use the search function or use the tags as a filter. Tags can help you to quickly have a general idea of the cases stored at the database. They can also help you find interesting cases in case you see a tag that seems important to you. If you know the word you are looking for or if none of the tags helped you, you can use the search box to look for cases (E.1). Once you select a case, you can click on it to examine its details. Depending on how the case was created, you can use it in many ways such as an input to a new solution, to communicate ideas with other people or to have an idea of how similar systems deal with the same problem (B.1, B.2, B.3). At the bottom of the case details, you can see the attachments, if any. These attachments can speed up your work if you want to reuse previous solutions (B.2, E.3). If you attached images, I show a thumbnail of each of them so that you can quickly preview its contents (C.2, E.3). After those, I list all cases that are related to the current one. I expect that, if the case you are looking at interested you, some of the related cases might also be helpful somehow (E.1). I also list all the tags related to the current case, so that you can use them to refine your search through tags. Finally, I offer some space for other users to comment on cases. With this, you can share knowledge with other people, such as other designers or clients (B.3)."

V. CHIDeK

The metacommunication message described in section previously guided the development of a case library tool called CHIDeK (Computer-Human Interaction Design Knowledge).

CHIDeK provides faceted navigation and semantic relationships to help with case library exploration, addressing requirement E.1. Faceted navigation is a popular way to allow users to browse a collection of items. Hearst et al. showed that faceted navigation is an effective method that can be helpful during a design activity [14]. CHIDeK's facets are based on: *task types* based on the work of Gonzalles-Caleros et al. [15], which are verbs, such as search or navigate; *design artifacts*, such as screenshots or documents; *UI widgets*, such as buttons and text fields; and *UI patterns* extracted from UI pattern libraries,[1,2] such as wizard and breadcrumbs. The task types facets were created to give the designer an option to browse cases adopting a problem-based approach or using a more abstract representation than widgets, whereas the UI patterns and widgets facets were created to provide a more concrete representation when following a solution-based approach. With this, we addressed requirement C.3. Table II lists all facets used in CHIDeK, and their corresponding values.

We briefly define the task types below:

- **Analyze**: examine an item set (compare, evaluate, decide);
- **Search**: identify a specific item in a set;
- **Communicate**: exchange information with the system;
- **Control**: control the flow of an action (e.g., play, stop);
- **Create**: create a new item;
- **Inform**: provide an information to the system;
- **Choose**: pick an item in a set;
- **Filter**: filter items in a set;
- **Insert**: include a new item in a set;
- **Move**: change the position of an item in a set or across sets;

[1] van Welie's pattern library (last accessed on May, 2014): http://www.welie.com

[2] UI Patterns pattern library (last accessed on May, 2014): http://ui-patterns.com

- **Navigate**: find one's way through pages;
- **Distinguish**: tell apart an item from other items;
- **Remove**: remove an item from a set;
- **Edit**: change an item.

TABLE II. CHIDEK'S FACETS

Task types	Artifacts
Analyze, Search, Communicate, Control, Create, Inform, Choose, Filter, Insert, Move, Navigate, Distinguish, Remove, Edit	Screenshot, Code Snippet, Wireframe, Document
Widgets	**UI patterns**
Progress bar, Button, Checkbox, Drop-down list, Image, Link, Map, Radio button, Slider, Spin box, Table, Text box, Tooltip	Tabs, Autocomplete, Breadcrumbs, Carrousel, Collapsible panels, Country selector, Coupon, Date selector, Directory navigation, Drag and drop, Fly-out menu, Form, List builder, List entry view, Pagination, Pull-down button, Rating, Search results, Search box, Slideshow, Table sorter, Thumbnail, To-the-top link, Wizard

CHIDeK's semantic relationships are based on the work by Janeiro et al. [16], which proposed a set of semantic relationships for UI patterns. We use semantic relationships to connect cases, as an alternative to help find related cases for the current design problem. When the user is examining a case, he can navigate to other related cases using these connections. Thus, we addressed requirements E.1 and E.3. The semantic relations we used are [16]:

- **X is (often) used with Y**: this is used when X and Y describe aspects of a system that are usually present together. For example, X can be a case describing the search feature of an online store selling theater ticket and Y can be the seat picking feature of the same kind of system (it does not need to be the same system described in case X, though);
- **X is similar to Y**: this is used when cases X and Y have similar characteristics. For example, the case describing a search feature of a ticket online store can be similar to a case describing a search feature from an academic paper database;
- **X conflicts with Y**: this is used when the feature described by case X must not be used with the one described by case Y. For example, case X describes the bureaucracy that users must undergo in an online banking system to make financial transactions (for safety reasons), and Y describes how a real-time system for an oil platform help users quickly perform actions that can avoid explosions in an oil well;
- **X implements Y**: this is used when case Y describes a system in an abstract way (e.g., using models or scenarios) and case X is an implementation of the system described by Y. For example, case Y shows a task model and a prototype of a new system, and case X describes how the system was implemented and feedback received from its users;
- **X abstracts Y**: this is the opposite of the 'implements' relation;
- **X enhances Y**: this is used when case X describes a feature which is an improved version of the one described in case Y. For example, cases X and Y describe ways a user can find a computer in online stores. In case Y, the user must use drop-down lists to pick the model, the manufacturer, price range or generic features of the computer (such as processor speed or memory capacity). As for case X, the user can look for a computer using a search box with autocomplete or faceted navigation, being able to refine his search as he goes on. In this case, since both case X and Y talk about the same goal, but in case X the search functionality is an improvement on the one described by case Y. We can say then that case X enhances case Y;
- **X simplifies Y**: this is the opposite of the 'enhances' relation.

To address requirements A.2 and C.1, we inserted 30 cases in CHIDeK to represent design knowledge related to web systems from multiple domains, such as online stores and car reservation systems. We performed a thorough inspection of those systems to collect the information needed for each case, such as who the targeted user is, what he intends to do and how the system tried to help him.

Fig. 2 shows CHIDeK's main page. The user can search for cases using keywords on the upper left box (label 1 in the figure). He can choose to perform the search in case titles, in case descriptions, or in both. Below that box lies CHIDeK's faceted navigation feature (label 2). At the wider pane at the right hand side (label 3), CHIDeK shows the cases found according to the selected search criteria. While the user has not searched for anything, CHIDeK shows all cases. Cases are listed using the title and a thumbnail, so that users can have an initial idea of what the case is about (requirement C.2).

Fig. 3 through Fig. 5 show how we decided to represent cases in CHIDeK for the study we conducted later. To make a case title descriptive, we adopted a verb + object + system name structure (e.g. Search Tickets – Ingresso.com, Buy Plane Tickets – Azul). When describing the cases, we tried to focus on using images and used as little text as possible (requirement D.2).

To address requirement D.1, we divided the case description in three parts: problem, solution, and evaluation. Fig. 3 shows the first part of the case with the case problem and the beginning of the solution. We described the case problem in a paragraph, introducing the intended (targeted) user along with what we believe she would want to do with the system and how. We represented the case solution by a short introductory text and by partial snapshots of the user interface with text balloons describing some design aspects of the system that we deemed important to support the user in performing her task. By choosing this representation, we addressed requirements C.2 and D.2.

Fig. 4 shows the evaluation part of the case, which is a paragraph describing an assessment of positive and negative aspects of the solution for the corresponding problem.

The representation we chose is flexible enough to allow the designer to insert cases describing different systems of the same domain (requirement B.1). He can also use the case description do discuss ideas with clients.

Fig. 5 shows case file attachments, case images, case relations, list of associated tags, and comments. The file attachments section lists the files of any kind which the user attached to the case, such as source code files, models and so on (requirements A.1, B.2, and D.3). The images section lists all the images attached to the case. The relations section shows the case links to other cases via the aforementioned semantic relationships. The list of tags is used to describe the case and find it through the faceted navigation mechanism. Users can also make comments about the case, such as extensions, adaptations, corrections, suggestions or criticisms.

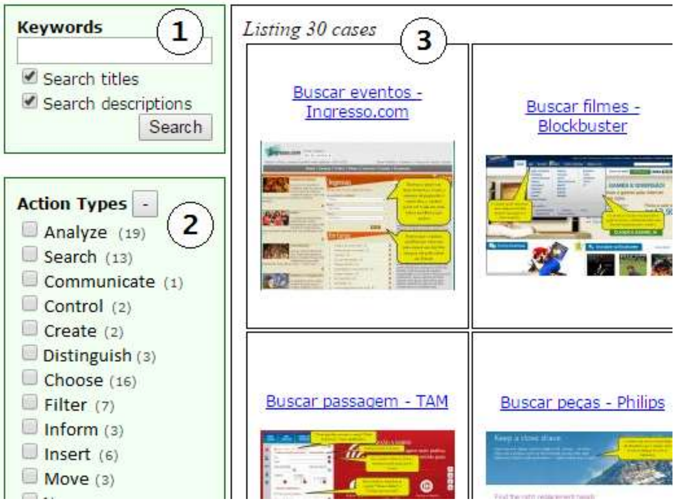
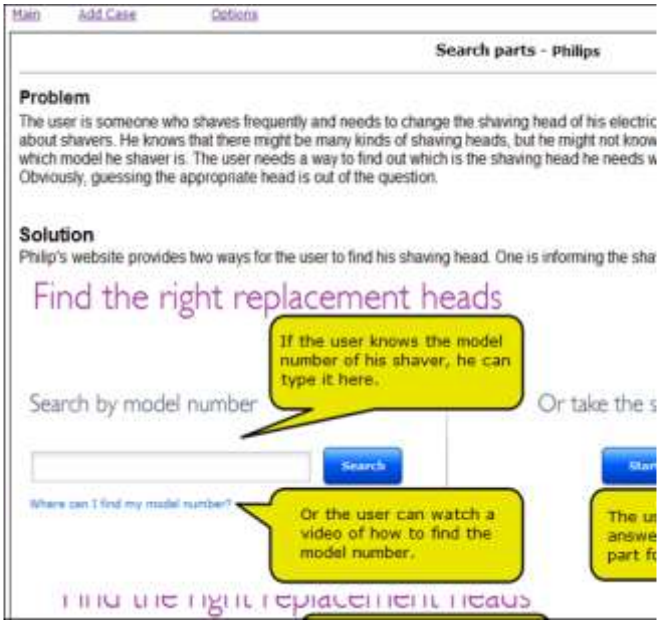


Fig. 2. CHIDeK's main page, cropped.



Search parts - Philips

Problem

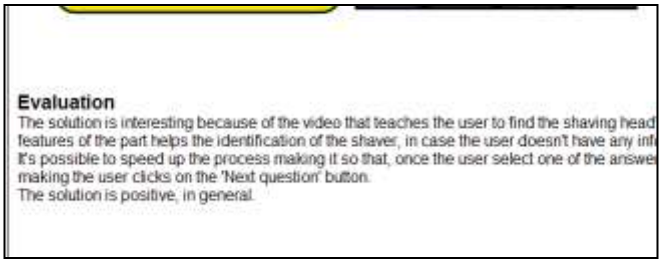
The user is someone who shaves frequently and needs to change the shaving head of his electric
about shavers. He knows that there might be many kinds of shaving heads, but he might not know
which model he shaver is. The user needs a way to find out which is the shaving head he needs w
Obviously, guessing the appropriate head is out of the question.

Solution

Philip's website provides two ways for the user to find his shaving head. One is informing the sha

Find the right replacement heads
If the user knows the model number of his shaver, he can type it here.
Search by model number
Or take the s
Search
Star
Where can I find my model number?
Or the user can watch a video of how to find the model number.
The u answe part fo
Find the right replacement heads

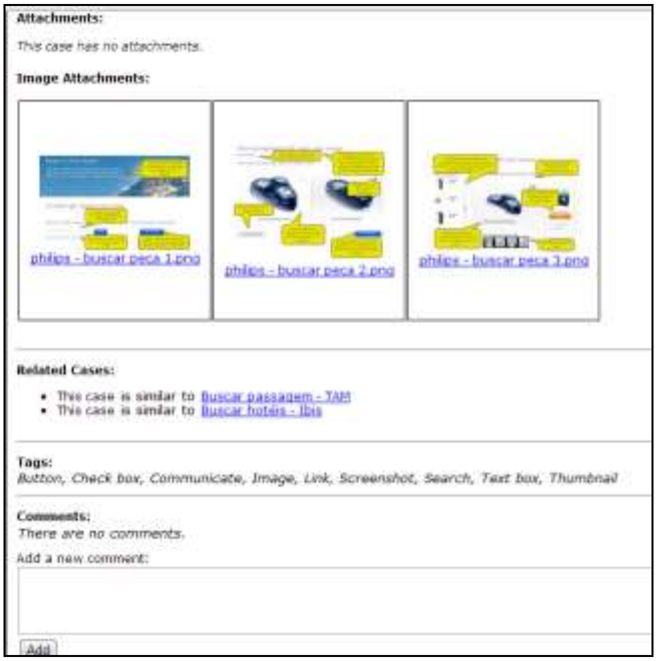
Fig. 3. Case description (problem and solution), cropped.



Evaluation

The solution is interesting because of the video that teaches the user to find the shaving head
features of the part helps the identification of the shaver, in case the user doesn't have any inf
It's possible to speed up the process making it so that, once the user select one of the answe
making the user clicks on the 'Next question' button.
The solution is positive, in general.

Fig. 4. Case description (evaluation), cropped.



Attachments:

This case has no attachments.

Image Attachments:

philips - buscar peca 1.png

philips - buscar peca 2.png

philips - buscar peca 3.png

Related Cases:

- This case is similar to Buscar passagem - TAM
- This case is similar to Buscar hotéis - Ibis

Tags:

Button, Check box, Communicate, Image, Link, Screenshot, Search, Text box, Thumbnail

Comments:

There are no comments.

Add a new comment:

Fig. 5. Case description (attachments, images, relations, tags list and comments), cropped.

VI. Evaluating the Design Activity with and without CHIDeK

This section describes the study in which participants split into two groups performed an individual design activity with or without CHIDeK support, according to their assigned group. The design activities and scenario were the same for all participants across both groups. The design sessions involved a single participant and the observer.

We selected 10 participants (4 men and 6 women) for this study, who were Computer Science students with a focus on HCI: one undergraduate student and nine graduate students. Despite our original intentions, we could not select participants with the same profile as the interview participants because time and location restrictions did not make it possible for professional HCI designers to participate. The goal of this study was to analyze how HCI students performed a design task when they had a CBR tool available or not.

The participants were randomly assigned to the groups. The participants of group A (the experimental group, with CHIDeK support) will be referred to as P1, P2, P3, P4, and P5. The participants of group B (the control group, without CHIDeK support) will be referred to as P6, P7, P8, P9, and P10.

The study consisted of three steps: first, the researcher explained what the participant had to do and gave him the scenario for a first read to clarify doubts; second, the design activity itself took place; third, the researcher conducted a brief interview to obtain more information on the participants' design process, and answer any questions that the observer had while watching the participants.

The scenario was about an online system through which the user can reserve bikes. The bikes are kept in stations distributed throughout the city. The user has to inform which type of reservation he wants (the options were: reserving a bike for 30 minutes, reserving a bike for one trip, and reserving the bike for one day), the station where she plans to get the bike and, optionally, a station where she plans to return the bike. She can also inform the amount of time she intends to stay with the bike, if desired.

To obtain data from the participants' design activity, we used the think-aloud technique [17], because we concluded it was a good way to have access to the participants' rationale. At the beginning of the session, the researcher asked the participants to speak what came to their mind without worrying about what it was, as if they were alone in the room talking to themselves.

During the sessions, we collected a movie of the participants' interaction with CHIDeK and a log of their interaction with the system (only for participants in group A), a video of their sketching activity, and the audio of their discourse (both during the task and during the interview). Both audio and interaction movie were recorded and transcribed with the participants' consent.

VII. Results

We analyzed the data from both groups using Thematic Analysis [13], a flexible technique for analyzing patterns (themes) within data. While transcribing the audio and video recordings, we started to analyze the data, coding data fragments and grouping them into higher-order themes that described the participants' design activity. After the analysis, we identified the following problem items each participant on both groups addressed during their design:

a) **Reservation type**: how the system user will choose the reservation type;

b) **Reservation date**: how the user will inform the system the day she wants her bike to be available;

c) **Reservation place**: how the user will choose in which station she wants to retrieve the bike;

d) **Delivery time**: how the user will inform the system when she intends to return the bike;

e) **Delivery place**: how the user will inform the system where she intends to deliver the bike;

f) **Procedure**: how the reservation process will be organized, i.e., the necessary steps and how they are sequenced;

g) **Reservation history**: a list of all of the user's previous reservations;

h) **Delivery**: a merge of the delivery time and delivery place items;

The reservation date and reservation history (items b and g) were addressed only by P1. The delivery item (item h) was addressed only by P4. All other items were addressed by all participants.

A. Group A Results

Group A (with CHIDeK support) showed that CHIDeK supported the participants' design activity in different ways. One way was by providing design knowledge that can be used during the design activity. For instance, once P1 saw the Unidas rent-a-car case in CHIDeK, he investigated the Unidas web system, which gave him ideas that he added to his design proposal. For instance, based on a car reservation system he saw in a case, he decided to use the wizard pattern to divide the bike reservation process in steps. P1 was the only one who consulted a web system outside CHIDeK. However, we believe that, if not for the study environment, the participants might have visited other systems more often, especially if links to the websites depicted in the case library were included in the cases themselves.

Another way that the cases supported design was described by P5, who said that even a slight stimulus such as seeing that CHIDeK has a case about a familiar domain can be helpful, because it can trigger the designer's memories about experiences with systems from that domain.

P5 used the action type tags most frequently. He even used the words from the action type tags, such as inform or choose, while thinking aloud throughout the activity, which we take as a positive result. In one instance, for the reservation type item, he was looking for a tag called "select", but found "choose" instead, which he thought fitting for the situation. From that moment on, he used the word "choose" for anything related to that item. Although we cannot say that he used that word because of CHIDeK, we can say that the terms used for action type tags were representative of what P5 had in mind. He would usually look for ways to "inform time" or "choose a reservation type".

We observed that CHIDeK's tags were helpful in some situations, but not so much in others. P2 picked the "choose" tag to find interesting cases related to the reservation type item, but he was not fully convinced that the results contained an interesting case. He said he needed to build a more complex query and did not feel that CHIDeK could help him formulate it. However, he could not explain during the interview what that query was. P3 had the same problem of not knowing what to look for when he had to deal with the optional user input fields described in the scenario. This might mean that, although faceted navigation can be a good tool to explore design options, it may not be enough to help the user in a complex search task, or that the set of tags needs to be expanded. The same can be said about CHIDeK's search mechanism, which is a simple keyword search.

The case titles played an important role in helping the designer identify the essence of a case. P1 used the case title to select which of the cases he would examine further. P4 ignored the problem section of the case, because he felt he understood the case problem just by reading the title, even though the case title contains much less information than the problem description in the case.

We also observed that each participant used various cases from different domains. For instance, P1 used cases from both a car reservation system and a hotel room reservation system. P2 used cases from an airplane ticket purchase system. P3 used cases from a DVD rental system. P5 used cases from Google Maps. This indicates that a case library which covers different domains might have a higher chance of being found useful by different designers.

When it comes to the case content representation used in CHIDeK, the participants' opinions diverged. For instance, while P1 found the thumbnails on the case list useful, because they gave him an idea of what the case was about, P2 did not find them useful at all, because he still needed to examine the case description to understand it. Another example is related to the representation used to describe the case solution. P2 wanted a more abstract representation of the cases, such as a description of the flow of the solution (e.g., a user-system interaction diagram) rather than a focus on the concrete user interface. Conversely, P3 seemed more comfortable with the more concrete representation delivered by the images. He also said that the description in CHIDeK should be more specific to each sub-problem and solution. In addition, P1, P3 and P5 complained that the cases had too much text. In general, they avoided reading them and tried to capture the gist of the idea by looking at the images and the balloons with annotations to the solution.

The evidences described so far show that the case representation we used is not very helpful if the designer shows a less exploratory behavior, such as P2, who wanted the case descriptions to focus on exactly what he wanted to see. He complained about the fact that, once he selected a case he chose by selecting some tags on the case list, he wanted the case description to focus on the tags he chose and not to describe the whole case in detail, i.e., he wanted the case description to adapt to focus on his selections.

It came to our attention that the case representation we chose lacks support for dynamic signs, which are important to describe interaction. This was evident while observing P1 who, to understand some aspects of the solution described in a case, decided to visit the web system the case discussed. Conversely, this could mean that the case description of easily accessible systems does not need to be very detailed, especially when it comes to their behavior. If the designer wants to experiment with the solution, he can always check out the system itself.

During the analysis, we realized that a case describing HCI design can be seen as the system designer's communication about his view of the system to other designers or to himself in a future moment. This idea bears some similarity to the metacommunication message, which is the designer's message about his view of the system to the user.

Also, it became clear the importance of static signs (such as the case images), dynamic signs (in our case, due to the lack of a way to effectively describe interaction) and metalinguistic signs (such as the text in the case, the notes on the images or an evaluation for each case) in the cases. We can say that the case description is a way for the system designer to communicate to other designers about the system's metacommunication and, to do so, he should make use of those signs to effectively convey his message.

This suggests that it is possible that Semiotic Engineering can be used to investigate better ways to describe HCI design knowledge in the form of cases. However, it should be noted that a case represents a different context of the system it describes. Not only is the case not restricted by the same signs used by the system, but the receiver of the case message (another designer) is different from the receiver of the metacommunication message (the system user). Still, it may be possible for Semiotic Engineering to help create better cases. However, instead of using the metacommunication message to learn about the designer's vision of the system, the receiver learns about the designer's vision of another system (the one described in the case).

B. Group B Results

Among the participants in group B (without CHIDeK support), P6 and P7 did not consult various web systems during the design. P8 and P9 only visited Bike Rio[3] to keep their proposals consistent with that system, which was familiar to them and assumed to be familiar to their users. P9 was the only participant who visited different systems, like Apontador (a phone and address directory web site), to try to solve the reservation place item. We came up with two hypotheses to explain this: either the scenario was too simple, allowing the participants to propose solutions using only their own knowledge; or, since the participants only had to make a draft to help in a first discussion with a client, they did not feel it was necessary to look into details. The latter is illustrated by P9, when he was thinking about how the system could recommend stations closest to the user. Although he put in his proposal that the system would do so, he did not detail *how* the system would do it.

Only the participants in this group seemed to be concerned about making their proposal consistent with the current Bike Rio system. This can be seen as a form of design fixation [18], since the participants did not seem to reflect on the quality of the current solution and used it "as is". Silva discussed this aspect of design, stating that designers tend to innovate only when they have a strong reason not to use the current solution, for fear that the user may reject or not be able to use an unfamiliar solution [19]. He concluded that designers should have existing design cases at hand to help identify bad solutions, lowering the degree of unwanted design fixation.

VIII. METACOMMUNICATION MESSAGE DISCUSSION

Table III summarizes all the metacommunication message fragments we assumed the participants received during the design activity with CHIDeK, as reflected by their verbal statements or by their behavior.

We contrasted the fragments we obtained from the study with the metacommunication message we created to help develop CHIDeK. With this, we show that, despite having used study participants with different profiles than the interviews participants, there is consistency between how we emitted our message, based on the expert designers' opinions, and how HCI students received it using CHIDeK. Therefore, the results of our research remain valid.

TABLE III. METACOMMUNICATION MESSAGE FRAGMENTS ASSUMED FROM THE PARTICIPANTS' DESIGN ACTIVITIES

Participant	Message
P1	**P1.1)** You can use the cases to help you reflect about the issue you are dealing at the moment or other issues you need to solve to complete your design. **P1.2)** You can use parts of the cases' solutions in your design. **P1.3)** Cases can also help you be sure whether a solution you have in mind is appropriate to solve the problem. **P1.4)** CHIDeK has many cases, which help you analyze tradeoffs or create new solutions by combining previous ones. **P1.5)** The description of the cases is not meant to help you understand the interaction of the systems described in the solution. **P1.6)** Cases provide a small summary of the solution to give you a general idea of how the system's designer decided to address the user's needs. You should use this information to decide whether the solution can be useful to solve your problem or not. **P1.7)** To find cases, you can use the tags on the left side. However, you can also browse the case list on the right side. To help you identify interesting cases, we titled them with names that gave a general idea of the problem they deal with. Furthermore, the case list shows a thumbnail picturing an example of what you will find in the case. **P1.8)** The solution section of the cases make heavy use of images to help you quickly grasp the idea of the solution implemented by the system designer.

[3] Bike Rio is a system that allows people in Rio de Janeiro to use bikes that can be reserved at several stations scattered throughout the city. At a station, the user can unlock a bike via phone calls or a mobile application.

P2	**P2.1)** Tags are domain independent. Therefore, to effectively use of them, you must consider more generic aspects of the problem or solution you want to explore. **P2.2)** If you already have a solution in mind, you can try using the widget tags to find cases. **P2.3)** You can use cases to find new solutions and reflect whether they are suitable for your problem. **P2.4)** If you are adopting a solution-based approach and has a concrete need in mind, the widget and pattern tags can be especially helpful. **P2.5)** CHIDeK offers many cases from which you can extract a solution and adapt it to solve your problem. **P2.6)** The cases can help you visualize how a solution you came up with was implemented in other systems. **P2.7)** CHIDeK's description of the solution focuses on interface issues and does not mention interaction aspects in detail.
P3	**P3.1)** Cases can offer solutions that help you remember your own experiences with similar systems or help you remember about how different systems implement similar solutions. **P3.2)** You can reuse the solutions described in the case in your design. **P3.3)** Cases can help you explore different solutions for the same problem. **P3.4)** Cases can also help you remember how systems you know solve a problem. **P3.5)** You can only search for cases using make broad queries.
P4	**P4.1)** You can use multiple cases, compare their solutions and identify a pattern of how different systems solved the same problem. **P4.2)** Cases help you remember interface details more quickly. **P4.3)** The images in the case help you see how the systems were implemented and its notes can guide you to the important parts of the solution. **P4.4)** If you do not know the domain the case deals with, you can read the problem section to have a better understanding. **P4.5)** The case title and the thumbnail can help find an interesting case more quickly.
P5	**P5.1)** Images can help you identify solutions that you might want to use in your design. **P5.2)** Cases can help you remember about your own experiences with similar systems. **P5.3)** You can use cases to help discussing ideas with other people. **P5.4)** If you are a novice designer, you can use tags dealing with concrete aspects of interface, such as widget and UI pattern tags. Otherwise, you can use the action type tags and see how other designers created solutions based on the actions users wanted to perform. **P5.5)** CHIDeK's case library concentrates designs from several systems, which helps you find previous knowledge without spending too much time searching the web. **P5.6)** The action type tags are generic actions the user can do with a computer system. By being generic, these tags allow you to find interesting cases quickly.

First, we observed that HCI students had some views in common with professional designers about using past design knowledge. We saw during the interviews that designers use previous knowledge as a source of reflection, to reuse solutions or to help communication. Here is part of what we envisioned as their need:

"(...) you turn to examples to find out what competitors are doing, to reuse previous solutions, or to talk about design ideas with other stakeholders."

This is in line with the messages that the HCI students received using CHIDeK. They used its cases to help reflection (see fragments P1.1, P1.3, P1.4, P2.3, P3.3, and P4.1), to reuse past solutions (see fragments P1.2, P2.5, P3.2, and P5.1) and expressed a wish to use the cases to help communicate with other people (see fragment P5.3).

We have also identified consistency between how we envisioned an appropriate case representation and how the students seemed to understand our choice for that representation.

"(...) you like working with images."

In the message students received from us, CHIDeK designers, images were selected to represent solutions because they are easier to understand and quicker to communicate ideas (see fragments P1.7, P1.8, P4.3, P4.5, and P5.1). They seemed to understand that images were used to speed up the identification of interesting cases and the understanding of the idea behind a case. This is in line with our vision that the designer does not have much time to spare using the tool.

"You don't have much time to spare, so the tool must speed up your progress. Therefore, you don't want to spend a long time in order to grasp the idea behind a case. In the same way, you don't want to spend a long time looking for an interesting case."

The students also received messages from us that made them identify other ways to speed up the design. P1 stated that the small summary at the beginning of the solution section was supposed to help designers decide quickly if the solution is useful or not (fragment P1.6). P4 understood that, by looking at cases, he could remember details from interfaces more quickly (fragment P4.5), while P5 understood that, by providing the action type tags, which, according to him, have a more general nature, we wanted CHIDeK users to find cases quickly.

The students understood our vision regarding the variety of design strategies. Our metacommunication message said:

"Depending on the situation, you can adopt either a problem-based or a solution-based strategy (...) There are tags focusing on the problem you need to solve, such as the action tags, and focusing on the solution you might already have in mind, such as widgets and UI patterns tags."

P2 realized that, in our vision, widget and UI pattern tags might be more suitable for a solution-based strategy (fragment P2.4), while P5 noted that the action type tags might be more suitable for a problem-based strategy (fragment P5.4).

Fragment P2.7 shows P2's interpretation of a limitation in the case representation we chose for this study. While, at first, this might seem like a disparity between our metacommunication message and how P2 seemed to receive the message, it actually shows that our initial vision of how the designer works matches what P2 wanted to do (to view cases at a more abstract level). More specifically, in our metacommunication message, we said:

"You are an HCI designer who works with a variety of artifacts, such as wireframes, prototypes and documents, and deals with multiple domains and restrictions. (...) it [a rich text field] gives you freedom to describe your case (...) You can also attach any kind of file to your case."

This shows that we anticipated that the designer might want to see a variety of representations in the cases. However, due to the short time we had to finish our study, we could not work on multiple representations for each case, such as task models or interaction models.

We have also observed an unexpected interpretation of our message. The message received by P1 and P2 stated that they could use cases as a way to attest the quality of solution they had in mind (fragments P1.3, and P2.6). Although this was not expected when we developed our metacommunication message, it does not represent disagreement; instead, it adds to the original message. This unexpected interpretation introduces a discussion about how a CBR tool can aid novice designers and expert designers. While none of the expert designers mentioned the need for a verification of the applicability of their solution (and, therefore, this was not implied in our initial metacommunication message), two HCI students interpreted that they could use the solution described in cases as a way to test their solution implemented in real systems. A possible reason for this is that seeing the solution they have in mind implemented in an existing system gave the HCI students confidence that their solution might be appropriate. P1 reinforces this finding during the interview when he said that it was "scary" to see a solution deemed appropriate by him to have a bad evaluation in the case.

These results suggest that a CBR tool might help HCI students (novice HCI designers) learn about HCI design. In fragment P5.4, the participant mentions this point when he said that widget and UI pattern tags might be more appropriate for novice designers, since they might not have much confidence in their knowledge.

IX. General Discussion

Participants in Group A did not seem to bother to keep their proposal consistent with the current Bike Rio system. Most of the time, they would either adapt the solution found in a case to their problem or use it directly after pondering whether it would be useful or not. We came up with two hypotheses for this difference: the first is due to the random assignment of the participants to the groups. Group A had only one participant who knew Bike Rio (P3), whereas Group B had four participants (P6, P7, P8 and P10) who knew that system. The second hypothesis is that having available a library with cases describing different systems in different domains motivates the designer to look for new alternatives. For example, although P3 knew the Bike Rio system, he used the library to view cases from other systems, such as Google Calendar, Blockbuster, and Apontador. However, more studies are necessary to test these hypotheses.

We observed an interesting behavior from one of the designers in Group B: he put himself in the user's shoes. P6 viewed himself as a user of the system he was designing when considering an option to reuse the station from a previous reservation to make a new one. In his reflection, he considered the place where he lives and his routine to come up with a solution. He used the same kind of reflection to discard the solution later. A similar fact occurred when P7 was justifying the need of a tolerance for the delivery time using the traffic of the city he lives in, and the fact that people are not usually punctual. These occurrences called our attention to the fact that the designer can sometimes consider his own lifestyle or the culture he is immersed in during the design process. Since this enriches the design activity by making the designer discover new requirements or create new solutions, as we saw during the activity, case libraries should also take into consideration contextual and cultural aspects. This is something the case content we used for the study sessions with CHIDeK lacked, and which can be motivated by the questions CHIDeK provides to guide the content creation.

Four out of the five participants in Group B used radio buttons to address the reservation type item in their design proposal. There were variations in the explanation of each reservation type, but we could not identify any striking differences on the overall solution. The only exception was P10, who, instead of making a proposal for a web system, made a proposal for a mobile system. His solution had buttons with the reservation type options. Conversely, in Group A, we identified two kinds of solution: the radio buttons with the reservation types, and a table analog to the one used by Blockbuster to show its DVD rental plans. This solution was used by P3 and P4. P3 said that the solution was inspired by the rental plans table he saw in the Blockbuster case and by his experience with other DVD rental websites. Although P4 did not mention Blockbuster or any other system when sketching his table, he had the Blockbuster case open on the computer monitor. This might suggest that a case can motivate different solutions even for simple problems, such as the reservation type.

X. Conclusion

In this paper, we have discussed the importance of previous design knowledge in the design activity, used by designers to understand further the problem they are working on or to develop solutions for it.

The interviews we conducted with professional designers revealed how they access and reuse previous design knowledge, as well as their motivations and expectations regarding the use design knowledge aided by a CBR tool. This resulted in a list with some requirements for a CBR tool to support HCI design, which is the first contribution of our work.

Our second contribution is the CHIDeK tool itself, which, as we observed, supported the design activity by providing solutions that the designers explored and used to solve the problem described in the scenario. CHIDeK also motivated participants to remember their own experiences with systems similar to the ones described in the cases. We observed, however, that CHIDeK could not equally help all designers find relevant cases. While some of them managed to find cases by using the facets or simply by looking at the case list, others could not find cases when they had a more specific need in mind. This shows that the quality of the user experience with CHIDeK strongly depends on the case library, its content and organization, to ensure that the set of facets and the direct search would help find the relevant cases.

We have also contrasted the metacommunication message we created originally, based on the interview with expert designers and the message we assumed the HCI students (novice designers) who used CHIDeK received. We discovered that, despite our metacommunication message being targeted to expert designers, the HCI students managed to understand most of it. The analysis also showed that research on how a CBR tool can impact the design from novice and expert designers is due.

On the sessions without CHIDeK, we observed that participants did not access external design knowledge very often. Two participants did not look for anything at all, creating their proposals entirely from their own mental recollection of existing systems. Three participants consulted external design knowledge. Two of them, however decided to consult only Bike Rio, because they saw their proposal as an extension of Bike Rio's current system and wanted to keep both interfaces consistent. Conversely, the participants who used CHIDeK consulted many cases from a variety of domains, such as online maps, airplane ticket purchase, car rental and DVD rental systems.

Based on the analysis of the data collected, we have formulated some questions that can guide further work on CBR tools for HCI design.

First, three participants of Group B (without CHIDeK support) claimed to know the Bike Rio system, while only one of the participants of Group A (with CHIDeK support) knew about it. This had a major impact on the proposals of the participants of Group B, resulting in evidences of design fixation. However, our study did not establish a clear relation between the support (or not) of the CBR tool, the participant's knowledge of the domain and how he designed his proposal. So, a few questions that come to mind are: How does the design activity with the CBR tool compare with the design activity without the tool *when the scenario is more complex and unfamiliar to the designer*? How do the activities in both conditions differ *when the participants are knowledgeable of the domain*?

We observed some participants used the faceted navigation, others the search function, and some simply browsed the case list. Despite the variety of access mechanisms, not all participants were fully satisfied with the results they obtained. These participants also did not know how to build more complex queries using CHIDeK. Moreover, browsing the case list would not be practical if the library had hundreds of cases. So, additional questions raised in this work are the following: How can we *improve the access* to cases? How can the case library help the designer *formulate his query* ranging from those who have a more generic need to a more specific need?

We also raised a few questions regarding the case content representation. In the study, we observed that, although most of the participants liked the use of images, the case representation we used had a few problems. P2 wanted to see the solution at the level of the interaction and not of the user interface. Likewise, P1, who liked the case representation we used, also had difficulties in understanding one of the cases, because the images did not show the behavior of the web system it described. This raises some additional questions, such as: How can the case representations *satisfy multiple preferences*, ranging from designers who prefer to deal with more concrete solutions to those who prefer more abstract solutions? Considering the interviews, when the designers said that they did not want to spend a long time using the tool and that the library must be concise, how can we *make informative cases without increasing the complexity of the library and the case description*?

Also concerning the case representation, we observed during the study that the Semiotic Engineering's metacommunication message is analog to the message described in a case. Therefore, some questions that come to mind are: How can *Semiotic Engineering further contribute to improve the quality of the case description*? How can *cases help the designer learn about the metacommunication message of other systems* and *contribute to his reflection* of the metacommunication message of the system he is working on? We can use this knowledge to develop guidelines to help designers think about the structure and content of the cases they want to create.

CHIDeK also presents several opportunities for improvement in terms of tool features, such as better support for search, for managing tags, for hiding balloons on top of the case images, and for more sophisticated support for dynamic signs. Another possible improvement is to make the semantic relations more visible. One idea would be to show the relations in a graph, like a visual thesaurus.[4] Finally, considering one of the requirements elicited in the interviews, integration with other tools, CHIDeK could read files from other HCI tools, such as mockups created by prototyping tools, and display their contents within the case.

Acknowledgment

We would like to thank CNPq for funding our research (process #308490/2012-6) and all the participants involved in the interviews and studies we conducted.

[1] R. Oxman , "Prior knowledge in design: a dynamic knowledge-based model of design and creativity," in Design Studies, 11(1), 1990, 17–28. doi:10.1016/0142-694X(90)90011-Z

[2] B. Lawson, What Designers Know. Architectural Press, 2004.

[3] D.A. Schön, The Reflective Practitioner: How Professionals Think In Action (1st ed.). Basic Books, 1983.

[4] J. Kolodner, Case-based reasoning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[5] A.K. Goel and S. Craw, "Design, innovation and case-based reasoning," in Knowledge Engineering Review, 20(3), 2005, 271–276. doi:10.1017/S0269888906000609

[6] J.A.G. Motta, Investigating the Case-Based Reasoning Process During Early HCI Design. Master Thesis. PUC-Rio, Brazil, 2013. Retrieved from http://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=23635@2

[7] C.S. de Souza, The Semiotic Engineering of Human-Computer Interaction. The MIT Press, 2005.

[8] J. Kolodner and D. Leake, "A Tutorial Introduction to Case-Based Reasoning," in D. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, & Future Directions (pp. 31–65). AAAI Press/MIT Press, 1996.

[9] J.B. Sambasiva, J. Barber, S. Bhatta, A. Goel, M. Jacobson, M. Pearce, E. Stroulia, "AskJef: Integration of Case-Based and Multimedia Technologies for Interface Design Support," in Proc. of the Second International Conference on AI in Design (AID'92 (pp. 457–475). Kluwer Academic, 1992.

[10] H. Kim and W.C. Yoon, "Supporting the cognitive process of user interface design with reusable design cases," in Int. Journal of Human-Computer Studies, 62(4), 457–486, 2005. doi:16/j.ijhcs.2004.12.001

[11] C.S. Peirce, Collected Papers of Charles Sanders Peirce: Science and Philosophy and Reviews, Correspondence and Bibliography. Harvard University Press, 1958.

[12] N. King, "Doing Template Analysis," in G. Symon & C. Cassell (Eds.), Qualitative Organizational Research: Core Methods and Current Challenges (Brief.). Sage Publications Ltd., 2012.

[13] V. Braun and V. Clarke, "Using thematic analysis in psychology," in Qualitative Research in Psychology, 3(2), 2006, pp. 77–101. doi:10.1191/1478088706qp063oa

[14] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Yee, "Finding the flow in web site search," in Communications of the ACM, 45(9), 2002, pp. 42–49. doi:10.1145/567498.567525

[15] J.M. Gonzalez-Calleros, J. Guerrero-Garcia, J. Vanderdonckt, and J. Munoz-Arteaga, "Towards Canonical Task Types for User Interface Design," in Proceedings of LA-WEB '09, 2009, pp. 63–70, IEEE. doi:10.1109/LA-WEB.2009.33

[16] J. Janeiro, S.D.J. Barbosa, T. Springer, and A. Schill, "Semantically relating user interface design patterns," in Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems, PEICS'10, 2010, pp. 40–43, Berlin, Germany. doi:10.1145/1824749.1824759

[17] K.A. Ericsson and H. A. Simon, Protocol Analysis, Revised Edition: Verbal Reports as Data. MIT Press, 1993.

[18] D.G. Jansson and S.M. Smith, "Design fixation," in Design Studies, 12(1), 1991, pp. 3–11. doi:10.1016/0142-694X(91)90003-F

[19] B.S. Silva, O Uso de Casos na Reflexão em Ação em Atividades de Design de IHC. Doctoral Thesis. PUC-Rio, Brazil, 2010. Retrieved from http://www.maxwell.lambda.ele.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=16990@1

[4] http://www.visualthesaurus.com