
An Online Multimedia System for Learning to Design User Interfaces

Elizabeth Furtado, Kenia Sousa, Leandro Taddeo, Daniel William
Universidade de Fortaleza - Mestrado em Informática Aplicada
Elizabet@feq.unifor.br

Abstract

This work addresses the problem of learning how to (i) design user interfaces (UIs) and (ii) evaluate their quality. We present an online collaborative multimedia system, the TELE-environment, to help UI developers solve problems of modeling, usability and acceptability during the entire development process. The developer accesses the environment as a tool for assistance, aiming at learning design concepts, guidelines and activities. During the interaction, the environment provides assistance with textual descriptions, examples and similar solution cases in different media. Moreover, the environment provides a collaborative support that allows the interaction among developers in a synchronous or asynchronous way.

keywords: collaborative learning, learning by cases, use case, scenario, UI design, online learning.

1. 1. Introduction

Problems of designing usable UIs are mainly due to the necessity that the developers, i.e., designers and software engineers, have to learn different models (task model, implementation model, etc) with different formalisms [4] in an autonomous and quick way. In addition, they have to follow many guidelines to improve the ways humans interact with computers. For instance, the UI should not display unnecessary information, and the dialogs should be simple enough for end-users to gain control of the system. Moreover, development methods do not provide efficient ways for developers to learn models and guidelines. A method description generally is a guide to help the developers create models [6] and to get a great amount of written guidelines [3]. They are also true because computer-based tools which implement these methods and generate interfaces automatically are generally difficult to use. Using these tools, the developers have to write specifications on the domain classes and tasks to be performed by the end-user. Other models used during the design of the interactive system can be derived from these specifications. However, it's important to point out that the quality of these specifications is a key issue. Some of the problems that can happen with these specifications and their validations are the difficulty in identifying potential end-users, inconsistency in the terminology used, and inadequacy to the end-users' needs.

Instead of giving solutions to the developers, several educational theories claim that a person learns better when brought to reflect about difficulties and discussing solutions with others [5]. In addition, a person learns better by actually doing something [17] and understanding the context and its relationships [16]. In contrast, using UI generator tools, the developers have generally a passive role in the design process. They depend on the tool, where most of the design knowledge resides. This led us to realize the importance of regarding the developers as active participants in the process, making all the important decisions related to design and evaluation.

With the learning system described here, called TELE-environment [7], the developers control their own learning process. The system consists of a multimedia learning environment for the web with different assistances (interactive examples, texts and cases) associated to themes about UI design. Themes correspond to concepts and activities existing in any method of UI design. Although themes are correlated, the developer can access texts and interactive examples in any sequence. An interactive example consists of: (i) presenting a problem related to modeling, usability and acceptability which should be analyzed, then (ii) demanding the developers to build possible solutions using UI design tools or CASE tools. These problems are represented in cases (called problem-cases). Cases represent situations that either comply with or violate several guidelines in a system specification, dialogue design and information presentation. The objective is to help developers understand and apply guidelines during the entire design process. In addition, the environment implements a collaborative resolution strategy that allows developers to discuss their design problems with others on-line.

The remainder of this paper is structured as follows: section 2 describes state-of-the-art tools to support learning and designing UIs. Section 3 provides the basic concepts used in TELE-environment. The TELE-environment architecture is described in section 4. Section 5 describes the process of the user interaction. Section 6 summarizes the main points of the paper.

2. 2. Approaches to User Interface Design

There are some methodological frameworks and tools for UI development [9, 6], called Tools for Working with Guidelines (TFWWG approach), which enable the integration of usability issues into the software development process. The main point of this approach is to design usable UIs that optimize the performance of tasks, through error reduction, higher throughput, and higher user satisfaction, and comfort. [15]

An important research goal of the TFWWG approach is to discover ways to integrate guidelines into the development process without stifling creativity [15]. These research studies include methods and tools that help the developers understand, search and apply design principles, guidelines and standards from the beginning of the design process. Therefore it is necessary to use notations that are familiar to UI developers and tools that are simple to use. For example, the JANUS system [2] is a CASE-based approach for UI development. It employs UML (Unified Modeling Language) with the help of the CASE tool Rational Rose [13] to allow an automatic transformation of the formal specification into a preliminary UI.

A complementary approach to the automatic generation approach is the use of design aid tools to support learning and designing. Some of these tools are: i) SIERRA [14], which offers the developer a detailed computer-based description of many guidelines; ii) Sherlock [9], which allows evaluating the generated UIs by offering the developer some advice about how to solve detected design problems, and iii) GUIDE [11], which offers the developers a CASE-based decision support system. The developers define their current design context and the system will find existing similar cases. New cases can be added to the tool database in order to help with future projects.

These tools support developers with learning and designing, but they do not require the developer to think about UI design problems through practice on the collective experience. One way to help developers in solving real problems is to create a virtual situation through the use of graphical tools. The graphical tool approach can help developers perform operations that can be illustrated through images, which can represent a problematic situation or depict a possible solution to a problem. The operations can be the following: contemplate alternatives, anticipate the implications of a given decision, follow a discourse while commenting on what is being shown, identify guidelines and evaluate designs for usability and conformance to applicable guidelines.

3. 3. The TELE-environment for learning: Ground Concepts

In this section, we describe basic concepts related to assistance provided by the TELE-environment, such as the situations which the developers face during the development process, and concepts related to its usability.

3.1.3.1. The Use Cases on UI Design

As mentioned before, cases are specific situations related to UI design, which developers face when developing a system. One example of a case may be the following: A developer validated a prototype with two end-users after they checked the conformity to their needs and the usability. Cases must be organized in generic situations of UI design in order to contextualize them [1]. To increase the effectiveness of representations, and recovery, we have associated cases to generic situations, such as “validating prototype” and “identifying the end-users’ needs”. These generic situations are associated with the UML notion of “use case”. Cases are therefore variations (instances) of a use case type that happens on UI design. Figure 1 is a simplified diagram of a use case organization for UI design situations. A use case may be split up into several specific use cases existing in a relationship of generalization, where each specific use case is a derivative of a generic use case. Generic and specific use cases form a hierarchy, where a use case may be linked to others from anywhere in the hierarchy. Thus, the use case hierarchy described is not a strict one. In the situation presented in figure 1, there is the relationship **use**, which means that if A uses B, then B can happen during the realization of A and the problems of A can be caused by problems of B. In some level of the hierarchy, problem-cases can be found. A problem-case is a description of a real situation, including characteristics of the context and of the participants. Its main goal is to make the developer think about the problem and possible solutions. An analysis-case is a problem-case with the solution already defined. Its main goal is to make the developer think about the proposed solution and discuss about other possible solutions with other developers.

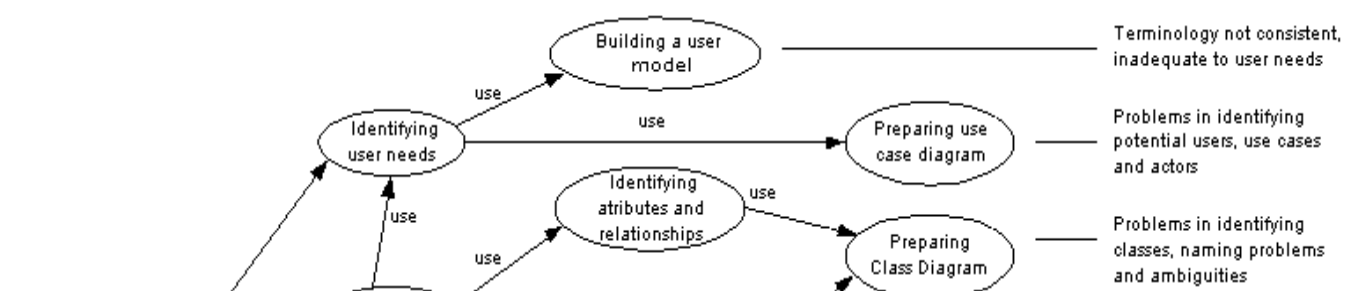


Figure 1 – Use case organization and UI design problems

Figure 1 – Use case organization and UI design problems

3.2.3.2. Usability of the Learning System

Many concepts studied in different areas (HCI, cognitive sciences, ergonomic, AI and pedagogy) have been considered in order to assure TELE-environment's usability. We associate these concepts to some components related to the development of the overall TELE-environment (such as the UI, the application and the end-user) (see Figure 2):

Quality of the User (eg. Allowing the user to collaborate and communicate with other users)	Usability through Interaction Devices	Usability of the User Interface Guidelines Multimedia Interaction	Quality of the Learning Application Related Themes
	Usability of the Computer-Based System		
Usability of the Overall On-line Learning System			

Figure 2 - Usability in the TELE-environment

(modified from [10])

- i) i) Using collaboration and communication mechanism to assure the quality of the end-user: TELE-environment implements the collaborative aspect by allowing a developer to share knowledge (obtained by practice) among the other developers by sharing any application. In addition, it enables developers to interact with others by means of a forum and chats.

- ii) ii) *Using interactive devices to assure the usability:* The developers need to choose the appropriate devices to facilitate the use of the system by the end-users, taking into account any kind of physical disability. The end-user can use the camera and the microphone to communicate with other end-users.
- iii) iii) *Using guidelines to assure the usability with multimedia interaction:* To determine how a case should be shown on the screen, guidelines related to graphical aspects and characteristics about the end-user and the context of use are considered. It allows us to determine the best way a case must be given to the developer during the interaction with this system.
- iv) iv) *Using hypertext to assure the quality of the learning application:* To facilitate the learning process during the exploration of concepts about UI design, themes are related through a hypertext structure, obtained from the use-case organization.

4. 4. General Architecture of the TELE-environment

Figure 3 depicts the architecture of the TELE-environment in three levels (presentation, management and information storage) including five main components:

- CADI, which implements the problem-based resolution strategy. It has an Assistance module, which coordinates the recovering and presentation of cases. A case can be shown on the screen in different ways, such as image, audio or text.
- TELE, which allows the collaboration and communication over the web by chat and the sharing of any application, such as a User Interface Design Tool. This component is described in [12].
- CADInet, which offers a UI Design course and assistance with texts and links. This component is described in [8].
- User interface, which is used by the developers to interact with the environment.
- The knowledge base, which will give support to most of the problem-solution-based process. This base contains the frequently encountered problem-cases (associated to scenarios, text and audio files), the use case organization, themes and rules.

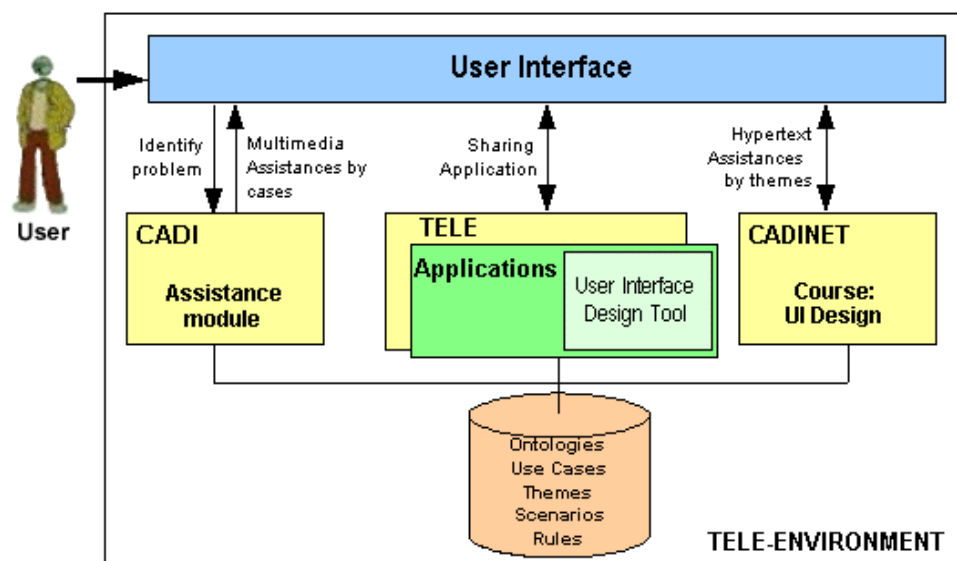


Figure 3 - TELE-environment architecture

The cooperation between the components is the following: the developer who is taking a UI Design course at CADInet, identifies a problem and requests the analysis of it. The UI component, during this analysis, gives the developer some assistance through CADI. The developer is assisted by cases, shown in different ways, which are recovered from the knowledge base by the assistance module. If the developer wants to communicate with other developers who have experienced similar problem-cases, and/or create an analysis-case for the problem, the UI component calls TELE module. The developer's solutions are validated by an expert before being effectively stored and made available to others.

The need for modeling different media has led us to develop a *graphic scenario editor*, described in [7]. A scenario about a usability problem could be, for instance, a picture illustrating an end-user entering data

and, accidentally, selecting a function, which closes the data input window. Mainly, scenarios are used to graphically express human factors (e.g., end-users' characteristics, position and expression, tasks and context). These factors are crucial for knowing which guidelines to use and why to use them. For example, in noisy environments such as shopping malls during holiday time, we must create a UI that works well for the hearing impaired.

In the next section, we will describe CADI's assistance process, taking into account the knowledge represented in the use case organization and developers' characteristics.

5. 5. Interacting with the System to Learn the UI Design

Case study. Figure 4 illustrates an interactive example, which the developers take when navigating in the interface logical design theme available in CADInet. In this example, the developers have to solve the following problem related to a prototype usability: *during the initial prototype validation, the end-users complain that they cannot find the most relevant information on the screen*. What must the developer do? Next, we will show how CADI can help them reflect about a possible solution.

The screenshot shows the CADInet web interface. At the top, there is a navigation bar with the CADInet logo and the text 'CADInet On-Line Users'. Below the navigation bar, there is a breadcrumb trail: 'Beginning > Class Room > Units > Theme > Interactive Example >'. The user's name 'Elizabeth Furtado' is displayed on the left, and a 'Log Out' button is on the right. The main content area is titled 'Discipline: Interface Design'. It contains a table with the following information:

Theme:	Interface Logical Design
Interactive Example:	Analyze a problem-case
Description:	Hi Designer, in the attached file there is a discipline about a problem-case for the Logical Interface Project phase to be analyzed. It is important to reflect on this problem to help you make decisions during the development of your own interactive system. If you want to communicate with another designer, click on the "On-line Users" link above and choose someone to help you build an analyses case sharing any tool or using a chat. In the end, send the analyses case to me.
Related Themes:	UI Concepts, prototypes, guidelines

Below the table, there is a section titled 'File with the Interactive example' containing a link 'Analyze a problem-case'. At the bottom of the page, there is a footer with the text '[UNIFOR - Universidade de Fortaleza]'.

Figure 4 - An interactive example on the WEB

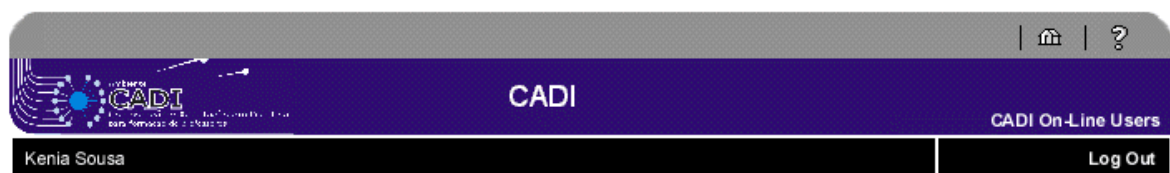
Case Identifying. Using CADI, the developers must at first identify the problem. After identifying it, the developers have two options: i) choose a use case which defines the context of the problem and subsequently the problem itself by selecting a problem-case, and/or ii) specify scenes, which express the situation. Once the problem-case has been accurately defined, the assistance module starts to work, retrieving and showing similar cases stored in the database of cases.

Case Retrieving. The assistance module asks some questions to the developers and applies algorithms and rules in order to retrieve already existing cases from the database. One of the algorithms for case retrieval allows the cross-contextual retrieving. If the developer chooses use cases, problem-cases are retrieved by taking into account the relationship *use* between the use cases. This happens because the developers may initially choose a problem-case thinking they know the problem well and, after answering questions about related use cases, the assistance module may determine that the problem in question is not the problem selected, but a consequence of a problem related to the one selected. This may happen because of the cross-contextual retrieval algorithm, and the loose hierarchical nature of the use case organization. In this example, the developer may choose the following use cases: *evaluating* then *integrating guidelines*. The assistance module asks about some graphical aspect of the UI, for instance, if

the most important information is already on the top left corner of the window. If any problem is found in the guidelines, the module tries to retrieve problems related to other use cases. Taking into account the relationship between *evaluating* and *identifying end-user needs*, some questions about the end-user's characteristics are asked. If the end-user has visual impairments, for instance, a problem-case related to end-users' disabilities can be retrieved, if not, problem-cases related to inadequate end-users' needs can be retrieved.

Case showing. To define how a case must be shown, this module takes into account the characteristics of the developer and of the context of use. After these characteristics are identified, it chooses the best view to show a case. For instance, a view having a combination of sound and image of a case, is not appropriate for a very noisy context (a laboratory, for instance). The views are predefined according to guidelines. These guidelines helped us to define the best way to divide a screen, to show many media at the same time, and to hear a story about a case.

Figure 5 shows a view combining a graphical scenario and a textual story of a similar case to the problem-case in question. It is a picture of an end-user not satisfied with a prototype, because it is difficult to accomplish a certain interactive task and the prototype is not in conformity to the requirements. Other end-users are also waiting to validate this prototype. Then, they will work together to solve the problems found.



Case: Interface Logical Design

The users test the prototype to check the conformity to their needs and to the usability. After the tests, the users find that the prototype is difficult to learn and to use because the analyst was not able to apply the appropriate guidelines to make it easy for the users to accomplish their task.

[Reflexive Questions](#) | [See More](#) | [Didatic Material](#) | [Found my case](#) | [Return to Cases](#)

6. 6. Implementation Aspects of the TELE-Environment

The environment presented here was implemented using different programming languages and technical solutions.

The scenario and rule editors were built in Borland Delphi 5.0. They access the knowledge base, whose internal representation is done by using SQL tables. The scenario editor makes it possible to construct the graphic scenarios through the GLScene component, which uses graphic objects built by the 3D Studio tool.

The graphic cases are shown in 3D and we intend to use the VRML programming language in order to allow the developer to manipulate the 3D virtual graphic scenarios in the web.

CadiNET is composed of approximately 100 pages programmed in Microsoft ASP and it also accesses the SQL tables that contain information on the user and the course as well as texts.

In the TELE implementation, two premises were initially established. The first one concerns the adoption of patterns established by such organisms as ITU (International Telecommunications Union) and IETF (Internet Engineering Task Force); and the second one refers to the needs in adopting the Internet-based environment as basic infrastructure.

We used some ActiveX controls of Microsoft NetMeeting. The interface was worked through the use of a group of parameters that control the objects that will be presented in the user interface. Among the different patterns that NetMeeting supports are the following: ITU H.323 for audio conference and video, ITU T.120 for multi point conference of access for data, and IETF LDAP for directory services.

7. 7. Conclusion

The TELE-environment described here is considered essential for the developer to make a usable UI system. To assure UI usability, guideline rules, and developer's beliefs and preferences have been defined, providing hypertext and multimedia assistance. To assure the quality of the learning process, the use case organization allows for UI themes to be related giving more autonomy to the developer and to associate cases in order to contextualize design and evaluation problems. In addition, cases are expressed in a natural and graphical way through an ontological structure. To assure the quality of the developer's task, a collaborative process is provided.

The problem discussion, the consequences of the solutions, and the collaborative support with assistance help the developer improve the projects and adopt an integrated ergonomic-technological-collaborative approach.

Our work is still in early stages; however some evidences make us to believe that the overall environment has effective possibilities. The knowledge base is based on concrete experiences, obtained from discussions in group. During these discussions, the groups debated about their practice, looking for the expression of knowledge. These discussions helped us to discover 25 beliefs and 80 cases. In the whole process, the method of problem solving guided us, always privileging the cooperation in group. At this point, we are examining these solutions that fit the new educational paradigm, which we believe is raising nowadays.

References

1. Aamodt, A., Plaza, E., *Case-Based Reasoning: Foundation Issues, Methodological Variations and System Approach*. AI Com Artificial Intelligence Communications, v.7, n.11, (1994).
2. Balzert, H.: From OOA to GUIs – the JANUS system. In: Proc. of IFIP Conf. On Human-Computer Interaction Interact'95 (Lillehammer, June 27-29, 1995). Chapman & Hall, London (1995) 319-324.
3. Bastien J.M.C., & Scapin, D.L., *Ergonomic criteria for the evaluation of user interfaces*. INRIA, No.156. (1993).
4. Bodart, F., & Vanderdonckt, J., *Expressing Guidelines into an Ergonomical Style-Guide for highly Interactive Applications*, In S. Ashlund, K. Mullet, A. Henderson, EL Hollnagel, T. White (eds.). In Proceedings of InterCHI'93. (pp. 35-36). (1993).
5. Dewey, J., *Como Pensamos*. São Paulo: Companhia Editora Nacional. (1959).
6. Furtado, E. *Mise en oeuvre d'une méthode de conception d'interfaces adaptatives pour des systèmes de supervision à partir des Spécifications Conceptuelles*. Thèse de doctorat. França. Março.(1997).

-
7. 7. Furtado, E., Mattos, F. L., Furtado, J. J. V., Holanda, R.. *Um Sistema de Aprendizagem Colaborativa de Didática Utilizando Cenários*. Brasilien Review of Learning Systems, Vol. 8. April. (2001).
 8. 8. Furtado, E., Silva, W., Alves, J.J., Távora F. *CADINET: Um Ambiente de Ensino e Colaboração a Distância para Formação e Atualização de Professores da UNIFOR*. Distance Learning International Conference. In press. August (2001).
 9. 9. Grammenos, D., Akoumianakis, D., Stephanidis, C.: Integrated support for working with guidelines: the Sherlock guidelines management system. *Interacting with Computers* 12, 3 (2000) 281-311.
 10. 10. Hix, D., Hartson, R. *Developing user interfaces*. Wiley. (1993).
 11. 11. Henninger, S.: A methodology and tools for applying context-specific usability guidelines to interface design. *Interacting with Computers* 12, 3 (2000) 225-243.
 12. 12. Neto, H., Raimir H., Bezerra W., Sarquis O. *Especificando o Tele-Ambiente no Contexto da Educação a Distância*, Anais do SBIE'2000. Alagoas: Novembro (2000).
 13. 13. Rational; <http://www.rational.com/products/rose/index.jtmpl>
 14. 14. Vanderdonckt, J.: Accessing Guidelines Information Sierra. In: Proc. Of IFIP Conf. On Human-Computer Interaction Interact'95 (Lillehammer, June 27-29, 1995). Chapman & Hall, London (1995) 311-316. Available at <http://www.info.fundp.ac.be/cgi-publi/pub-spec-paper?RP-95-020>
 15. 15. Reiterer, H.: Tools for Working with Guidelines in Different Interface Design Approaches. *Tools For Working With Guidelines* (2001) 225-254.
 16. 16. Schank, R., *Dynamic Memory: a Theory of Learning in Computers and People*. Cambridge University Press. (1994).
 17. 17. Schank, R. Cleary, C. *Engines for Education*. USA: Paperback. (1995).