Evolution of an e-Learning Environment Based on Desktop Computer to Ubiquitous Computing: GUI Design Issues

Ana Isabel Molina Díaz¹, Miguel Ángel Redondo Duque² and Manuel Ortega Cantero³

Abstract - Using the new wireless technologies, mobile devices with small displays (handhelds, Personal Digital Assistants, mobile phones) are present in many environments. We are interested in the effective use of such mobile computing devices for supporting collaborative learning. We want to show its application in a study case: the teaching of Domotics. To reach this goal, we analyze the tasks which are susceptible of improvement through mobile computing. We will take as starting point a collaborative e-learning environment of domotical design, based on the desktop metaphor, called Domosim-TPC. We pretend to adapt this tool to the characteristics of mobile devices. This evolution is based in a task-based analysis. In this paper we describe some learned lessons in the semi-automatic generation of users interfaces for a PDA-type devices. This generation process is carried out starting from the user interface for desktop devices.

Index Terms - Computer Supported Collaborative Learning, user interfaces, mobile computing, task modeling.

INTRODUCTION

The main goal of the work which this article is situated is to incorporate the ubiquitous computing paradigm in the teaching and learning of domains with a high experimental degree in order to take into account mobile computing possibilities. Also, the features of these domains provide an excellent framework to analyze the collaborative process. Thus, we are going to study the methods that allow us to systematize these tasks. We will take as a starting point a collaborative e-learning environment based on the desktop metaphor, following the named “Domosim-TPC [1]”. To achieve our goal, we analyse the tasks which are susceptible of improvement through mobile computing. The aforementioned system is used to support the learning of the design of automated control facilities in buildings and housing, also called Domotics. The term Domotics is associated to the set of elements that, installed, interconnected and automatically controlled at home, release the user from the routine of intervening in everyday actions and, at the same time, they provide optimized control over comfort, energetic consumption, security and communications.

In this paper, we present the process followed to obtain a PDA-based (PDA, Personal Digital Assistants) version of Domosim-TPC system in order to support learning task using mobile devices. In particular, we focus the discussion about the process to design the user interface for PDA-type devices. First, teaching design procedures in domotics are introduced; next, we describe the main features in the Domosim-TPC system and the stages necessary to develop a mobile version of the aforementioned system are summarized. We describe some stages relative carried out to build the user interface for a prototype to be evaluated in real contexts. The following section gives a overview of task modelling. And finally, we show the first results in the evolution process of the asynchronous learning tasks supported in Domosim-TPC towards PDA devices and we will draw some conclusions and outline the future work we will develop.

TEACHING DESIGN PROCEDURES IN DOMOTICS

In Spain the new regulation for “Formación Profesional” (Technical Training) takes professional profiles into account and training in Domotics is considered a must. Some learning stages in electricity and electronics courses are centered on the study of the design and maintenance of singular installations and automation of buildings dedicated to housing. In this area the design of domotics installations have a fundamental role. In this kind of training, the realization of practical experiments is especially important. However, the material necessary to carry out these assignments is usually expensive and in many cases it is not adequately provided. This problem gets worse with the difficulty to bring the student closer to real situations, to replicate accidents and to simulate those chaotic situations which may happen in the real world; Domotics designs should aim to cope with the troublesome situations, too. In order to soften this problem by means of the use of technology, we have developed a distributed environment with support for distance learning of domotics design: Domosim-TPC [1]. This tool is explained in the next section.

THE DOMOSIM-TPC ENVIRONMENT

Domosim-TPC is a collaborative e-learning environment for domotical design learning.
Using the DomoSim-TPC system (figure 1), the activities of practical learning of domotical design are structured in three clearly differentiated stages. In each of them diverse cognitive exercises are carried out and approached and representations of expert knowledge are used.

![Collaborative Planning of Design](image)

**FIGURE 1**

**STAGES AND TASKS OF DOMOTICAL DESIGN LEARNING CARRIED OUT IN AN EXPERIENCE WITH DOMOSIM-TPC.**

Next, these stages are described concisely and the most outstanding tasks carried out in each one of them are pointed out:

- **Specification of models and planning of their design strategy.** In this stage the students, in an individual way, reflect and plan the steps to build a model satisfying the requirements proposed in the problem formulation. The strategy traced by the user is dynamically contrasted with an optimal plan of design for this problem.

- **Discussion, argument and search of consent in the characteristics of the models individually built.** In this stage, the participants discuss about the models built, about their types and about the steps carried out to obtain them. From this process a proposal (model) is obtained reflecting the view point of each participant.

- **Detailed design and simulation in group.** Before checking the validity of the proposed solution, the apprentices should detail and organize the attributes associated to the objects that form the model. Later on, they consider the hypothesis and case studies that should be contrasted by means of the Collaborative Simulation of the behavior of this model.

We are interested in the effective use of such mobile computing devices for collaborative learning. There are tasks in the system DomoSim-TPC which are susceptible of improvement through mobile computing. In particular, the Collaborative Planning of Design is a asynchronous and reflexive task which could be improved using mobile devices.

Towards a mobile access to DomoSim-TPC:

Most approaches in CSCL are based in Direct Manipulation like interaction paradigm. We are interested in the exploration and search for new interactive technologies suitable for use in the classroom. Particular emphasis is being placed on studies aimed at turning the classroom into a ubiquitous, mobile, and computer supported collaborative environment as a solution to some complex problems.

Ubiquitous computing as the interaction paradigm was first introduced by Mark Weiser [2, 3] in Xerox PARC laboratories in Palo Alto in 1991. This interaction paradigm changes the concept of using the computer by distributing multiple low-powered computers or small computer components throughout the environment concealing their use by disguising their appearance adapting them to the traditional tools used in the classroom. We try to fuse the principles of the CSCL and ubiquitous computing. Thus, we consider a classification of systems according [4]to the characteristics of ubiquity involved and to the kind of collaboration adopted. Some of these systems belong to a category that supports collaboration in asynchronous discussion interfaces. This type of systems provides a set of tools for discussion making it possible to present individual work to the group. These systems define both an individual workspace and a group workspace for discussion, and even in some cases areas of results, but only of an asynchronous type.

**Stages in evolution process**

The evolution process of DomoSim-TPC to incorporate PDA-based access to the DomoSim-TPC system consists of several stages:

a) **Analysing tasks** that can be improved in a mobile computing scenario.

b) **Design of tasks** taking ubiquitous computing paradigm principles into account. Modeling and the design of certain tasks must be reconsidered. The devices and protocols necessary for materializing these tasks must be decided.

c) **Implementing a prototype** that applies proposed theories.

d) **Evaluating the prototype** in real contexts.

e) **Identifying the task patterns that could be common in CSCL environments**, based on the resolution of proposed problems and simulation of solutions contributed by students.

f) **Creating a tool that allows**, from a tasks model of a CSCL application, **obtaining the equivalent interface for several mobile devices in a semiautomatic way**.

Actually, we are in (c) stage. We are developing PDA-based asynchronous tools in order to increase the realism and flexibility of the task supported in DomoSim-TPC.

**ANALYZING AND MODELING TASKS USING CTT**

The new context of use implies reconfigurations of the UI that are beyond he traditional UI changes [5, 6], such as the redistribution of widgets across windows or tabs in a tabpanel, the reduction of a full widget to its scrollable version, without using a sophisticated widget, to replace a interactor with a smaller alternative, etc.

There are several solutions to the problem of building device-independent user interfaces. An interface model for separating the user interface from the application logic and the presentation device is necessary. There are several markup languages that help in this purpose (UIML [7], XML,...). This kind of languages allows the production of device-independent presentations for a range of devices. But these solutions do not provide high-level guidance guaranteeing quality across multiple versions of applications. We propose...
the use of a model-based design of GUI [8], which focuses on the tasks supported. The idea is that task analysis provides some structure for the description of tasks or activities, thus making it easier to describe how activities fit together, and to explore what the implications of this may be for the design of user interfaces. A number of approaches to task modeling have been developed (GOMS [9], HTA [10], CTT [11, 12],...). All these methods to task modeling have limitations for our purpose.

GOMS [9] notation (Goals, Operator, Methods, Selection rules) is not a real task analysis technique but rather a task-based dialog description technique. GOMS does not consider user errors, possibility of interruptions and can be inadequate for distributed applications. Also, it considers only sequential tasks. In HTA [10] (Hierarchical Task Analysis), tasks are defined as activities that people do to reach a goal. However, in the notation only a task hierarchy is modelled and the goals are not explicitly represented. The notation HTA sometimes provides many details that are not very useful for the designer. UAN [13] (User Action Notation) supports hierarchical specifications and has operators to express temporal relationships among tasks. This is a textual notation that specifies actions at a low level. We prefer a graphical representation of the tasks and focus on activities. Both features are easier to interpret for designers and users, and facilitates the communication between them.

We have chosen the graphical ConcurTaskTrees (CTT) notation [12] for analysing tasks in Domosim-TPC. Some important features are supported in CTT: hierarchical logical structures, temporary relationships among tasks, and cooperative tasks modelling. Also, a automatic tool for task-based design (CTTE) [14] is available.

Nevertheless, CTT doesn’t provide an optimal support for specifying collaborative aspects. Cooperation and collaboration are terms that are sometimes used interchangeably. We consider the definition presented by Dillenbourg et al [15] “Cooperation & collaboration do not differ in terms of whether or not the task is distributed, but by virtue of the way in which it is divided; in cooperation the task is split into independent subtasks; in collaboration cognitive processes may be divided into intertwined layers. In cooperation, coordination is only required when assembling partial results, while collaboration is a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem”. CTT permits representing cooperation, in which a division of labour exits. Each person (role) is responsible of a portion of work. Collaboration is a mutual engagement of participants in a coordinated effort to solve a problem. Also there are objects that are created and manipulated by many people. The contribution of each person to the final object is not well identified.

Due to the aforementioned limitation the incorporation of capacity for representing collaborative tasks to the CTT notation is a requirement.

**Generating User Interfaces PDA-Based to the DOMOSIM-TPC System**

In this section, we describe the analysis of the main tasks in asynchronous workspace supported in Domosim-TPC. This analysis should be done at a low level. It has to determine the kind of interaction task (for example, enter text, select a Boolean value, select a numeric value) and the kind of domain application objects manipulated by the tasks. This information facilitates the identification of the visual component (widget) that best allows the realization of a particular task, taking target device restrictions into account. The logical decomposition of tasks is also reflected in the selection, consistency and grouping of elements in the GUI obtained.
To obtain the version for PDA of the individual workspace, temporary relationships among the tasks and the domain manipulated objects must be taken into account. This information allows creating the interface in which both the widgets (user interface objects) that show domain application objects (internal objects) and the widgets that allow executing certain actions applicable to these internal objects must appear together. In the design plan editor (individual workspace) two internal objects are handled: the design action and the design plan (a collection of design actions). Figures 3 and 4 show the names (in uppercase) of both objects. They are part of the name of the tasks that manipulate them.

Diagram in figure 3 shows the general functions that can be performed on the design plan. It can be shown graphically. There are two modes of visualization: a list of nodes (a node represent a action) connected by arcs (representing precedence relationships); and the design of the scene that is created for executing the planned actions list. We can also save the design plan. The option Clear eliminates all the information contained in the actions list. These actions are applicable to the plan design object. These must appear in the user interface next to the object related (the list box that shows the sequence of steps in the plan). The resulting PDA interface of these subset of tasks are shown in figure 5 (a).

In addition, the individual plan editor handle design action objects. In the diagram shown in figure 4 the actions Add_DESIGN_ACTION and Delete_DESIGN_ACTION are included. The first one has certain complexity. When a task (that means an operation over a internal object) is of the interaction type, the mapping to a perceptible object (a widget in the interface) is more direct. This kind of operations can be represented by means of buttons, options in a menu or a contextual menu. It has been applied to the mapping of the Delete_DESIGN_ACTION operation, or the aforementioned generic functions, which the user can perform on the DESIGN_PLAN object.

However, when a task has a certain complexity, i.e., when a task is represented by an abstract task, with several abstraction levels and several interaction tasks (this occurs in the New_DESIGN_ACTION task), more complex visual components are necessary (a panel, in a PC version of the interface; or in a PDA, where there are display resolution constraints, a dialog box is a better choice). This occurs in the task that allows creating new design actions, as we can see in figure 5 (b). This dialog box appears whenever a new design action is created.

Next, we show the results of applying the aforementioned method for generating the PDA version of the other workspaces in Domosim-TPC (discussion and result workspaces).
The discussion workspace (Figure 6) is used to organize the semistructured discussion and argumentation, besides the contributions made by the participants in an activity. This way, the participants will be able to give opinions on some aspect, argue about a proposal, and consult the opinions given by the other members of the group. In DomoSim-TPC, a contribution is any element characteristic of the semistructured discussion mechanism established for the activity. The sending and type of a contribution depend on the hierarchical organization and the role associated to the author.

The system presents a careful organization and representation of the contributions, resorting to a tree-shaped hierarchical organization, where the root is the activity itself outlined by the coordinator of the group. At a first level of the tree, as many nodes as sections or epigraphs derive following the outlined activity. The rest of the nodes represent the contributions generated from the discussion in search of a solution, always in agreement with the conversational structure outlined for the activity. The different branches of the tree can expand and contract to facilitate the visualization of the structure of the discussion made by the participants.

Figure 7 shows the PDA-type interface supporting the semistructured discussion and argumentation. We can observe that the same functionality is supplied in this small version of the interface. This interface cannot be split because all visual components are related to the same task.

Finally, the space of results (Figure 8) facilitates the visualization and perception of the parts of the activity that the group has already developed and agreed. The participants can observe the results and solutions obtained during the process, independently of the discussion and the development process necessary for their achievement (this is the main goal of a groupware tool).

Alike the discussion space, the result space has a tree-shaped hierarchical structure, used to organize the information relative to the results obtained in the activity. As the tree root the identification of the activity itself appears together with its symbolic icon and the identification of the coordinator of the group who proposed it. In the following level of the tree the epigraphs or sections that make up the activity are located.

You can observe the contents of some of the obtained solutions. With the selection of some of the node leaves or nodes that represent solutions to an epigraph, the right part of the window (Figure 8) displays the actions that constitute the design plan corresponding to the marked node solution. In this situation it is also possible to make use of a graphic visualization of this solution.
In figure 9 we show the PDA version of the results workspace. Several kinds of representation are available in separate tabpanels. Unlike an desktop user interface, each representation uses a single dialog.

CONCLUDING REMARKS AND FUTURE WORKS

Using mobile computing devices in collaborative learning scenarios we could improve the effectiveness of the learning process. In this paper we have shown its application to a case study: domotical learning by problem solving. To achieve our goal, we have analyzed the tasks which are susceptible of improvement through mobile computing. To do this, it is necessary to restructure the desktop user interface to adapt it to the constraints of size and utility of this kind of appliances. We use graphical ConcurTaskTrees (CTT) notation for analyzing tasks in Domosim-TPC (a collaborative e-learning environment of domotics design). Some important features are supported in CTT: hierarchical logical structures, temporary relationships among tasks, and cooperative task modeling (multi-user cooperation). This notation aims to provide an abstract representation of these aspects. Nevertheless, CTT does not provide optimal support for specifying collaborative aspects. This is an important lack of this notation.

We intend to extend CTT notation to supporting representation of collaborative tasks. Also, we intend to identify common high-level task patterns in CSCL environment and guidelines that allow creating a complete semi-automatic environment that generates CSCL and mobiles tools, independent of the study domain and platform.

ACKNOWLEDGMENT

This work has been partially supported by the Junta de Comunidades de Castilla – La Mancha and the Ministerio de Ciencia y Tecnologia in the projects PBI-02-026 and TIC2002-01387.

REFERENCES