Model-Based Evolution of an E-Learning Environment Based on Desktop Computer to Mobile Computing

Ana I. Molina[1], William J. Giraldo[2], Francisco Jurado[1], Miguel A. Redondo[1], and Manuel Ortega[1]

[1] Department of Information Technologies and Systems,
Computer Science and Engineering Faculty,
Castilla – La Mancha University,
Paseo de la Universidad, 4. 13071 – Ciudad Real. Spain
{AnaIsabel.Molina,Francisco.Jurado,
Miguel.Redondo,Manuel.Ortega}@uclm.es
[2] Systems and Computer Engineering, University of Quindío, Quindío, Colombia
wjgiraldo@uniquindio.edu.co

Abstract. In the last years a great amount of collaborative applications have been developed. Advances in wireless technology and its integration on mobile devices offer support to user-to-user interaction on the move, becoming any place a potential collaborative scenario. With the aim of obtaining an appropiate support for the development of multi-plataform groupware applications we propose use a model-based approach for the development of interactive groupware applications (CIAM). This approach can be used for supporting the evolution of existing systems based on desktop metaphor towards mobile support. In this paper we show the application of this method to a case study, the teaching of Domotics. We will take as a starting point a collaborative e-learning environment called "Domosim-TPC".

Keywords: Mobile computing, Methodological framework, Graphical User Interfaces evolution process, Computer Supported Collaborative Learning, Model-Based Design.

1 Introduction

In the last years a great amount of collaborative applications have been developed. On the other hand some of them have been developed according to the paradigms of ubiquitous or mobile computing. Advances in wireless technology and its integration on mobile devices offer support to user-to-user interaction on the move, becoming any place a potential collaborative scenario.

Most of mobile collaborative systems are carried out in the same manner as other applications are developed, without taking into account the special characteristics of these paradigms. Therefore, the requirements that characterize these paradigms may not be considered in the most appropriate way; in special, we have to mention the aspects of user interface (UI) development and the perception of the context of the application. From our point of view we need appropriate frameworks and tools (Methodologies, processes, specifications techniques, CASE tools, etc.) to help in the analysis and design processes of these complex applications [1, 2]. Clearly groupware activities can be improved in an important way by taking into account mobility aspects. However the current approaches do not offer an integrating and efficient solution that tackles jointly mobility issues and group work aspects.

The main goal of this paper is to incorporate the mobile 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.

With the aim of obtaining an appropiate support for the development of multiplataform groupware applications we propose use a model-based approach for the development of interactive groupware applications (CIAM) [3]. This is a methodological framework supported by a set of notations for modeling and designing interactive and collaborative tools. This approach can be used for supporting the evolution of existing systems based on desktop metaphor towards mobile support [1]. We intend to identify common high-level task patterns in *Computer Supported Collaborative Learning (*CSCL) environments and guidelines that facilitate the creation of a complete semi-automatic environment that generates CSCL and mobile tools, independent of the study domain and of the platform.

In this paper we are going to show the application of our method to a case study, the teaching of Domotics. We will take as a starting point a collaborative e-learning environment called "Domosim-TPC" [4]. The paper is structured as follows. Section 2 describes the main features in the Domosim-TPC tool (used as a starting point environment). In the following section, the stages necessary to develop a mobile version of the aforementioned system are enumerated, that is, our model-based evolution process is described. This process is based on the use of conceptual specifications using the CIAN notation. This notation has been proposed in the context of a methodological approach called CIAM. In this section the CIAM methodological approach is shown, enumerating its several stages, and the aspects that are specified in each. Finally we will show the evolution process of the asynchronous tools in Domosim towards PDA support (individual workspace to design models) and we will draw some conclusions.

2 Domosim-TPC

The domain where our investigation is being applied is 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, when installed, interconnected and automatically controlled at home, release the user from the routine of intervening in everyday actions and, at the same time, provide optimized control over comfort, energetic consumption, security and communications. In this kind of training, the realization of practical experiments is especially important. 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 [4].

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.

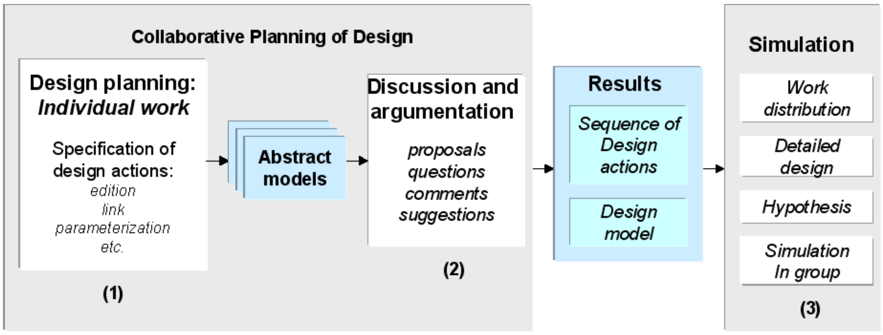


Fig. 1. Stages and tasks of domotical design learning carried out in an experience with DomoSim-TPC

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 viewpoint 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.

3 Model-Based Evolution Process

Our purpose entails improving the traditional classroom environment with the collaborative and the mobile computing paradigms. We intend to implement a prototype in a particular environment for collaborative learning (Domosim-TPC). We have to adapt the asynchronous tools of Domosim to the characteristics of mobile devices. To do this, it is necessary to restructure the user interface to adapt it to the constraints of size and utility of this kind of appliances.

In order to sustain a learning activity, we identify the concept of *space*. This is, a virtual structured place with resources and tools to perform a task (to solve a problem). There are three spaces: (1) *An individual workspace to design models satisfying a specification.* (2) *A shared space for discussion and argumentation* about the design actions that the learners have planned. (3) Another *shared space with the results of the discussion process.*

The process of evolution of Domosim-TPC towards mobile computing consists of several stages:

1. *Analysing tasks that can be improved by using mobile computing.*
2. *Design of tasks* taking mobile computing paradigm principles into account.
3. *Implementing a prototype* that applies proposed theories.
4. *Evaluating the prototype in real contexts.*
5. *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.
6. *Creating a tool that allows*, from a tasks model of a CSCL application, *obtaining in a semiautomatic way the equivalent interface for several mobile devices.*

If we want to generalize this process to the learning environment of other disciplines, we can automate the transformation process of the user interface. For CSCL tools developers, this introduces the problem of constructing multiple versions of applications for different devices. There are many dimensions to consider when designing context-dependent applications (environments, platforms, domain, users,…). With the aim of obtaining an appropiate support for the development of multi-plataform groupware applications we propose use a model-based approach for the development of interactive groupware applications (CIAM) [3]. This is a methodological framework supported by a set of notations for modeling and designing interactive and collaborative tools.

3.1 CIAM: A Methodological Approach for Modeling Interactive Groupware Applications

In this section the CIAM (*Collaborative Interactive Applications Methodology*) proposal is presented. CIAM is a methodological approach for the development of CSCW (*Computer Supported Cooperative Work*) applications that takes into account the modeling of group work and interaction issues. Unlike other existing proposals in the fields of conceptual modelling of CSCW systems and modeling of issues related with the Computer Human Interaction, CIAM considers the joint modeling of both issues, as well as the differentiation between the concepts cooperation and collaboration [5].

This approach consists of three main elements:

- A *conceptual framework* that clearly defines the concepts studied and modeled in each one of the phases in the methodological proposal [6].
- A *methodological framework* that defines the set of phases that compose the proposal, as well as the set of specification techniques to use in each of them [3]. In the figure 2 we can see the stages of the CIAM proposal. In each of them several collaborative and interactive systems issues are specified. The

information represented in each of the stages serves as basis for the modeling to be made in the following stage; so that this information is extended, related or specified with a greater level of detail in the following stage of the process.

- A *notation*, called CIAN (*Collaborative Interactive Applications Notation*), that allows expressing the peculiarities of the interactive groupware systems (figure 3).

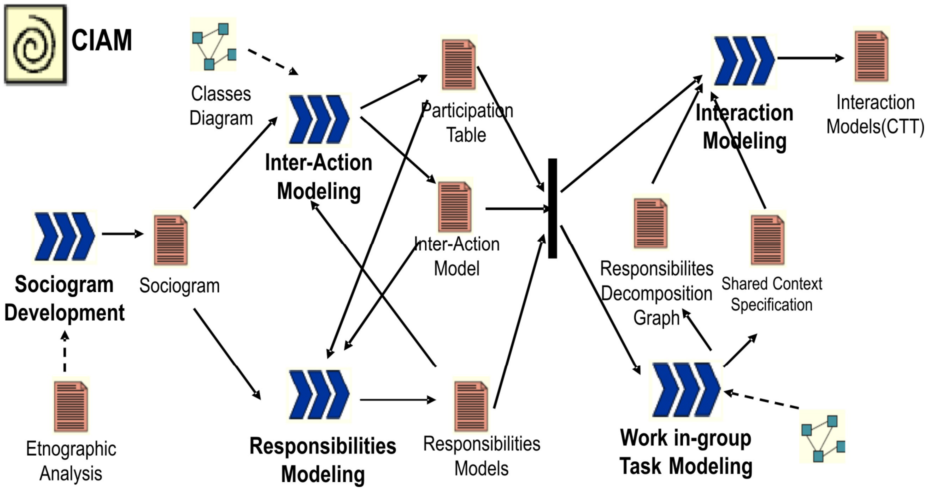


Fig. 2. CIAM methodological proposal stages

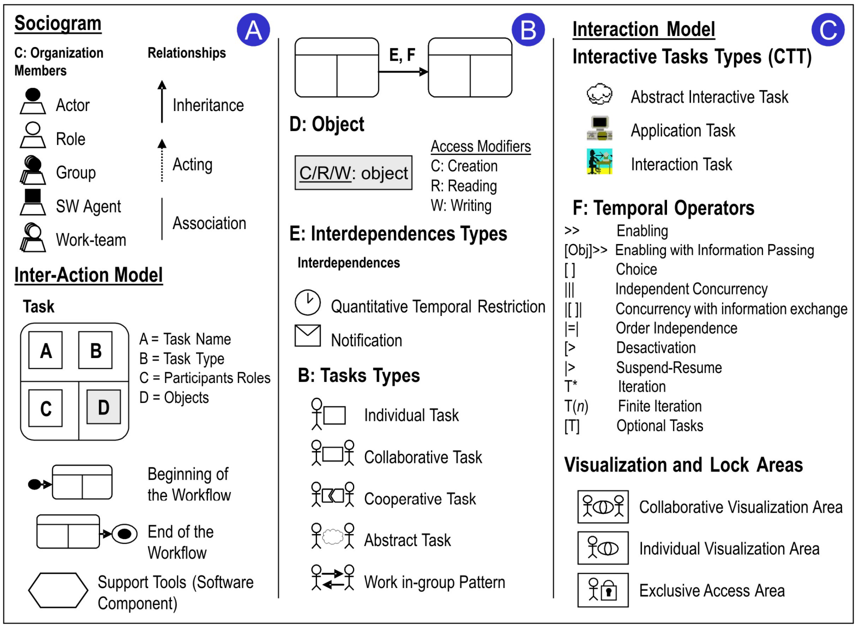


Fig. 3. CIAN Notation

In Table 1 we summarize the specification techniques used in the several stages of our methodological approach, the notation used and the product obtained in each.

Table 1. Specification techniques and result obtained in the several stages of CIAM

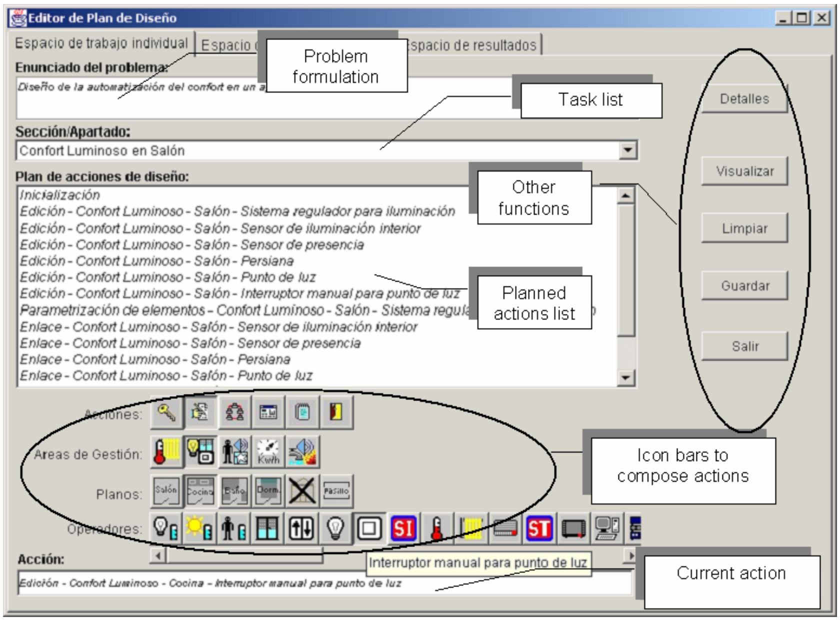
CIAM Methodology Stage	**Specification Technique**	**Representation Type**	**Result obtained**
Sociogram Developmen	Sociogram	Graphic	Organizacional Structure Specification *(roles, actors, software agents, etc)*
Responsibilities Modeling	Responsibilities Model	Textual	Detailed specification of the responsibilities of each role
Inter-Action Modeling	Participation Table	Textual	Relationships between the main tasks ans roles of the system
	Inter-Action Model	Graphic	Work Structure and workflow to be performed by the organization
Work in-group Tasks Modeling	Access Control Matrix	Textual	Relationships between data objects and roles at level of work in-group task
	Table of Operation Permissions on the Shared Context	Textual	Relationships between operations and roles at level of work in-group task
Collaborative Tasks Modeling	Responsibilites Decomposition Graph	Graphic	Responsibilities distrivution in a cooperative task specification
Cooperative Tasks Modeling	Shared Context Specification (classes diagram in standard UML notation)	Graphic	Specification of the Shared Context in a collaborative task, the division in visualization areas and the finalization policy
Interaction Modeling	Interactive Tasks Decomposition Tree (CTT notation enriched with icons for specifying visualization areas in collavorative tasks)	Graphic	Interaction Modeling at individual responsibility level; interaction with the shared context in collaborative tasks

4 Applying CIAN for Adapting Collaborative Tasks in Domosim-TPC to Mobile Computing Support

We intend to obtain the mobile version, and in particular, the PDA version of the individual plan edition space in Domosim-TPC (figure 4). The evolution process aforementioned is based on the use of conceptual models of the starting system. Next, we describe the analysis and modeling of the main tasks in asynchronous workspace of Domosim-TPC using the CIAN notation. The models, created in the context of CIAM, allows to adapt the user interface of Domosin-TPC to mobile computing support. Also, these conceptual models are used for automating and generating this evolution process.

Next we are going to explain the creation of the *sociogram* and the *inter-action* models associated with the system Domosim-TPC. In figure In Figures 5 and 6 we can see the appearance of both.

In the ***Sociogram** Development* stage, the organization structure is modeled, as well as the relationship between its members. In the example we have the following roles: *Teacher*, *Student*, *Observer* and *System* (that adds expert knowledge to the learning process). We identify some specialized students called *Planner* and *Designer* that take part in the two main tasks in the environment. The inheritance relationships



Enunciado del problema:

Diseño de la automatización del confort en un a

Sección/Apartado:

Confort Luminoso en Salón

Plan de acciones de diseño:

Inicialización
Edición - Confort Luminoso - Salón - Sistema regulador para iluminación
Edición - Confort Luminoso - Salón - Sensor de iluminación interior
Edición - Confort Luminoso - Salón - Sensor de presencia
Edición - Confort Luminoso - Salón - Persiana
Edición - Confort Luminoso - Salón - Punto de luz
Edición - Confort Luminoso - Salón - Interruptor manual para punto de luz
Parametrización de elementos - Confort Luminoso - Salón - Sistema regula
Enlace - Confort Luminoso - Salón - Sensor de iluminación interior
Enlace - Confort Luminoso - Salón - Sensor de presencia
Enlace - Confort Luminoso - Salón - Persiana
Enlace - Confort Luminoso - Salón - Punto de luz

Interruptor manual para punto de luz

Acción:

Edición - Confort Luminoso - Cocina - Interruptor manual para punto de luz

Fig. 4. Plan Editor User Interface

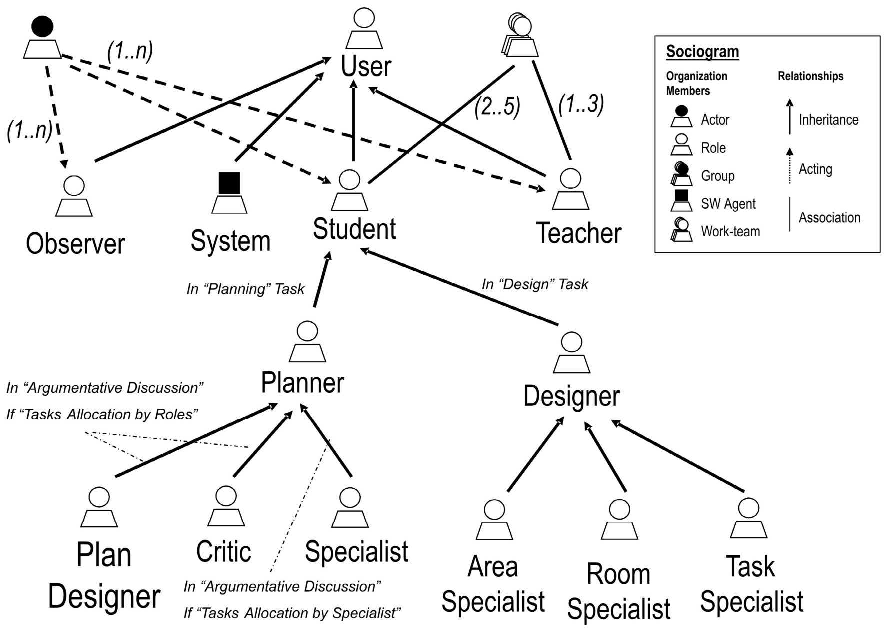


Fig. 5. Sociogram of Domosim-TPC

can be enriched with the definition of *conditions*. For example, we show that a *Student* can be specialized in the *Designer* role in the context of the *"Design"* Task. The *Planner* role has several subroles (*Plan Designer, Critic* or *Specialist*). A *Planner* can play any of them depending on the "*Task Allocation*" chosen for the activity. Once the *inheritance* (generalization / specialization) *relationships* among roles established, the *actor-role acting relationship* is added for the main roles in the diagram. This kind of relationship can be labelled when we want to express the cardinality (minimum and maximum) in the cases in which the specification establishes restrictions on the matter. The diagram can also show the relationships among roles that can, at a certain moment, work together. These relationships are expressed by means of *association relationships.* In figure 5 we can see that the *Student* role and the *Teacher* role are associated, creating work group. This indicates that there are tasks in which both with their respective responsibilities take part.

In the following two stages (***Responsibilities Modeling*** and ***Inter-Action Modeling*** stages) the abstract tasks (or processes) that define the work in-group developed in the organization, as well as the temporal and data dependencies that exist among them are described. It is created the so-called *participation table*, which provides an initial idea about the division of the work at the highest level of abstraction. Once the participation table has been constructed, the *Responsibilities Model* can be defined. It specifies in more detail and under an individual perspective the responsibilities associated to each one of the roles in the organization, adding to its shared responsibilities those that are exclusive to it. The information specified in these two stages is supplementary, being necessary for both models to be coherent.

Figure 6 shows the *inter-action model* associated to the system taken as example. In this example we use a symbol to express *Abstract Task* (6.d), that is, group work tasks that can be decomposed into others in a lower level of abstraction and of different kinds (6.g and 6.h). Collaborative and Cooperative tasks must specify the roles involved in its execution (6.a), whereas in the individual tasks only a role must appear. For all the tasks the objects manipulated and their access modifiers are indicated (6.b). For each task we can specify the so-called *Domain Independent Support Tool* (6.c). These are the supporting tools that implement well-known patterns or interaction protocols. Between the tasks *Configure Experiences* and *Planning* we can see temporal and data dependencies (6.c), indicating that the data *Activities* are transferred and the relation between these tasks is sequential (>>). Between task *Planning* and *Design and Simulation*, there is a period dependence (6.f) and a condition that must be checked (6.e).

In the ***Work in-Group Tasks Modelling*** stage the group tasks identified in the previous stages are described in more detail. Two different kinds of tasks are distinguished and modelled in a distinctive way: the *collaborative tasks* and the *cooperative tasks.* The *collaborative tasks* specification is based on the *shared context* definition [7], that is, the set of objects that are visible to the users and the actions that can be performed.

Finally, in the ***Interaction Modelling*** stage the interactive aspects of the application are modelled. For each task of individual nature detected in the previous stages of the process, an *interaction model* is created. The notation used for the interactive models is the *interactive tasks decomposition tree* from CTT [8]. As for the collaborative tasks, CIAM allows obtaining the interactive model directly from the *shared context* definition [9]. In particular, the division of the shared context made in the previous stage is used for organizing the UI in *workspaces*. The interaction model obtained facilitates the obtaining of the final UI.

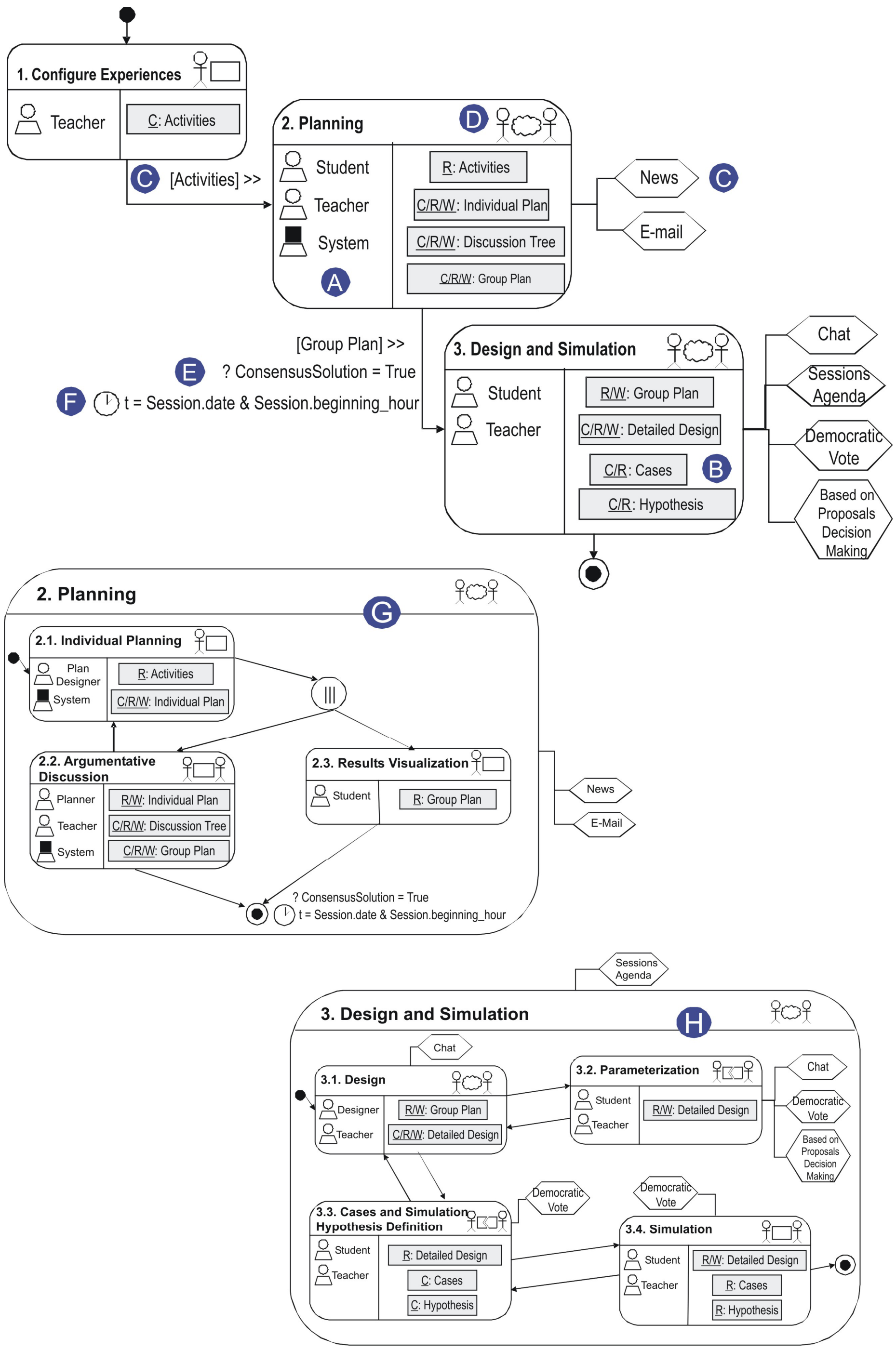


Fig. 6. Domosim-TPC Inter-Action Model

Figure 7 shows the task model in CTT notation for individual task "*Individual Planning*" (task 2.1 in figure 6). 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 *INDIVIDUAL_PLAN* (a collection of design actions). Figure 7 shows the names (in uppercase) of both objects. They are part of the name of the tasks that manipulate them.

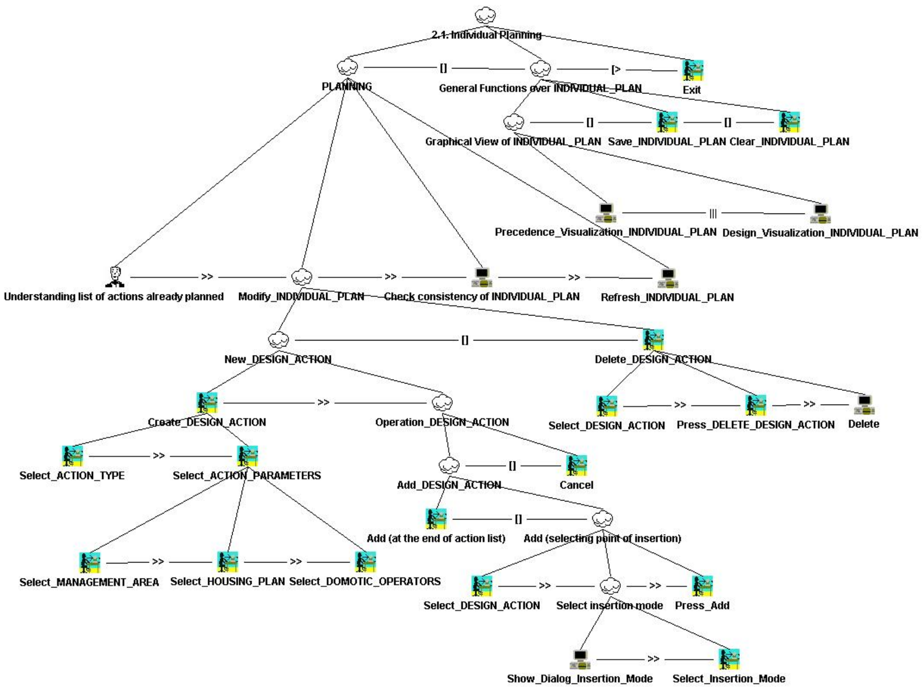


Fig. 7. Modeling of abstract task *"2.1 Individual Planning"*

Diagram in figure 7 shows the general functions that can be performed on the individual plan. It can be shown graphically. There are two modes of *visualization*: a list of nodes (a node represents an 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 individual plan. The option *Clear* eliminates all the information contained in the actions list. These actions are applicable to the *INDIVIDUAL_PLAN* 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 8 (a).

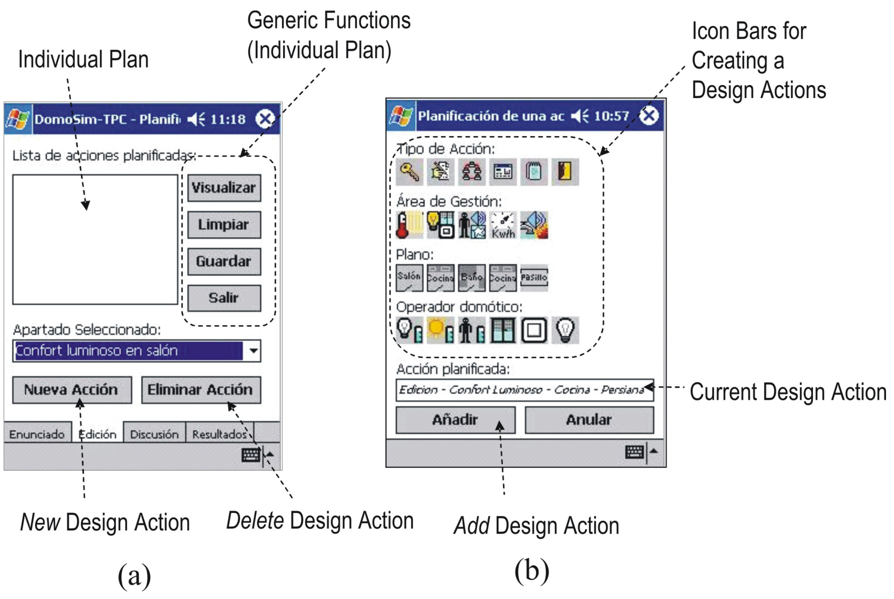


Fig. 8. PDA Version of the interface to individual plan edition. (a) Interface that allows showing and performing actions on the INDIVIDUAL_PLAN. (b) Dialog box that allows the creation of a new INDIVIDUAL_ACTION.

In addition, the individual plan editor handle *DESIGN_ACTION* objects. In the diagram shown in figure 7 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 *INDIVIDUAL_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 8 (b). This dialog box appears whenever a new design action is created.

In figure 9 we can see the process followed by the CIAM proposal until the obtaining the interaction models. These models of interaction are expressed in CTT and works as an entry point to use it with development tools of user interfaces based on models as is the case of TERESA [10] or the Dygimes framework [11]. We have proposed a method for obtaining the final user interfaces that also considers the object manipulated in the context of interactive tasks. We have shown as applying our method an user interface is obtained starting from an interaction model that gathers the requirements to support collaborative tasks.

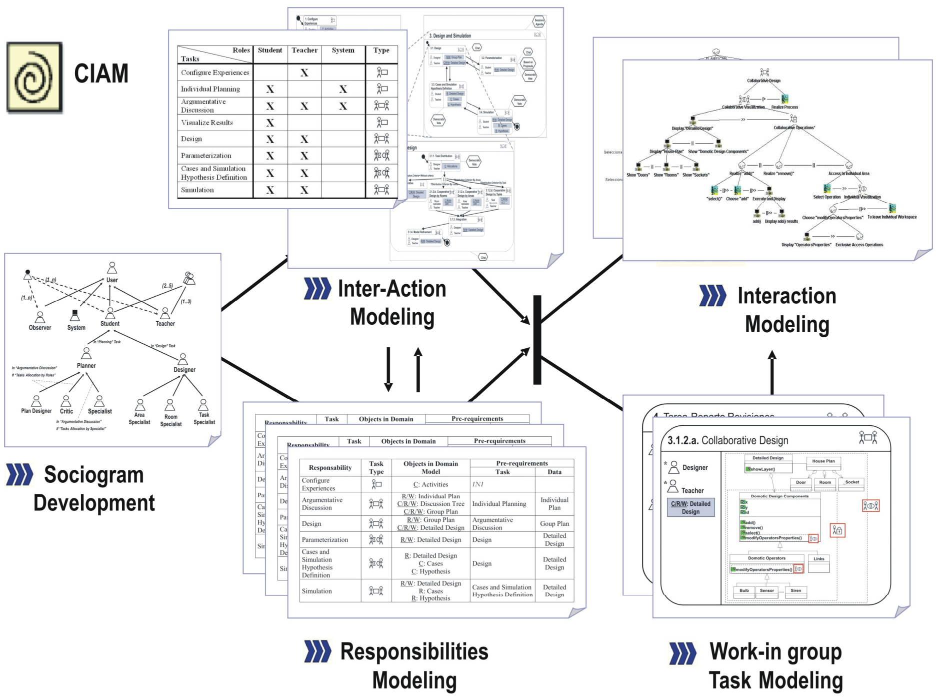

CIAM
Inter-Action Modeling
Interaction Modeling
Sociogram Development
Responsibilities Modeling
Work-in group Task Modeling
3.1.2.a. Collaborative Design

Fig. 9. Overall process followed by the CIAM proposal until the obtaining the interaction models

5 Conclusions and Future Work

In this article we have presented how to apply a methodological approach (CIAM) for the development and evolution of user interfaces to support collaborative and interactive activities that can be developed in a context of mobile computing. CIAM guides the designer following different phases from modeling to reaching interaction models. CIAM is based on the CIAN notation that allows users to accurately describe the features of a collaboration process (roles, responsibilities, tasks, shared context, etc).

We have shown the application of CIAM in the design and evolution process of the user interface that supports the task of collaborative planning of design in the context of the system Domosim-TPC.

Therefore, this has been a case of study really appropriate to discover the CIAM potential in helping to the development of collaborative user interfaces. Also it has been useful to show the necessities of extension of CIAM in order to add certain features of the mobile computing paradigm and especially some parameters that allow modeling the context. The results in the application of this Methodology show the necessity to include those aspects closely related with context modeling and the synchronization of contents; that is why we make an outline of the way to take into account these characteristics as a future work.

Acknowledgments. Work supported by the Castilla–La Mancha University and Junta de Comunidades de Castilla–La Mancha in the project mGUIDE (PBC08-0006-512) and M-CUIDE (TC20080552).

References

1. Eisenstein, J., Vanderdonckt, J., Puerta, A.: Applying model-based techniques to the development of user interfaces for mobile computers. In: ACM Conference on Intelligent User Interfaces IUI 2001, Alburqueque (2001)
2. Puerta, A., Eisenstein, J.: Towards a General a Computational Framework for Model-Based Interface Development Systems. In: IUI 1999: International Conference of Intelligent User Interfaces, Los Angeles (1999)
3. Molina, A.I., Redondo, M.A., Ortega, M.: CIAM: A methodology for the development of groupware user interfaces. Journal of Universal Computer Science (JUCS) (Designing the Human-Computer Interaction: Trends and Challenges) (2007)
4. Redondo, M.A., Bravo, C.: DomoSim-TPC: Collaborative Problem Solving to Support the Learning of Domotical Design. In: Computer Applications in Engineering Education, vol. 4(1), pp. 9–19. John Wiley & Sons, Chichester (2006)
5. Dillenbourg, P., et al.: The Evolution of Research on Collaborative Learning. In: Reimann, P., Spada, H. (eds.) Learning in humans and machines. Towards an interdisciplinary learning science, London, pp. 189–211 (1995)
6. Molina, A.I., Redondo, M.A., Ortega, M.: A conceptual and methodological framework for modeling interactive groupware applications. In: Dimitriadis, Y.A., Zigurs, I., Gómez-Sánchez, E. (eds.) CRIWG 2006. LNCS, vol. 4154. Springer, Heidelberg (2006)
7. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. In: Proceedings of the Conference on Computer Supported Cooperative Work CSCW 1992, Toronto, Canada. ACM Press, New York (1992a)
8. Paternò, F.: ConcurTaskTrees: An Engineered Notation for Task Models. In: Diaper, D., Stanton, N.A. (eds.) The Handbook Of Task Analysis For HCI, LEA, Mahwah, NJ, pp. 483–501 (2004)
9. Grudin, J.: The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. In: 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2000) (2000)
10. Paternò, F.: CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. IEEE Transanctions on Software Engineering 28(9) (2002)
11. Luyten, K.: Dynamic User Interface Generation for Mobile and Embedded Systems with Model-Based User Interface Development, PhD. thesis, Universiteit Limburg (2004)