

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



# Evolution of Collaborative Learning Environments based on Desktop Computer to Mobile Computing: A Model-Based Approach

Ana I. Molina, William J. Giraldo, Francisco Jurado,  
Miguel A. Redondo and Manuel Ortega  
*University of Castilla-La Mancha  
Spain*

## 1. Introduction

In the last years a great amount of collaborative applications have been developed. These applications can be framed in the paradigms of *Computer Supported Cooperative Work* (CSCW) or *Computer Supported Collaborative Learning* (CSCL) according to their specific purpose (group work or learning in-group). On the other hand some of them have been developed according to the paradigm of 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 learning 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, in relation to 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, specification techniques, CASE tools, etc.) to help in the analysis and design processes of these complex applications. However the current approaches do not offer an integrating and efficient solution that tackles jointly mobility, learning and group work issues. With the aim of obtaining an appropriate support for the development of multi-platform groupware applications we propose to use a model-based approach for the development of interactive groupware applications. In particular we propose to use a proposal called CIAM (Molina et al., 2009). 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 (Eisenstein et al, 2001). In this chapter we show the application of this method to a case study. We will take as a starting point a collaborative e-learning environment called Domosim-TPC (Redondo & Bravo, 2006) and show the application of our method.

This chapter is structured as follows. Section 2 presents the related works in the field of modeling of interactive and collaborative issues, as well as proposals for automation and model-based generation of user interfaces. In the following section, our model-based evolution process is presented. 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 a CSCL system, called Domosim-TPC, towards PDA support and we will draw some conclusions.

## 2. Related Works

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, as for example UIML (Abrams, 1999). 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. Other alternative is the use of a model-based design of UI (Paternò, 1999), 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 (Foley et al., 1991), HTA (Annett & Duncan, 1967), CTT (Paternò, 2004), etc.). The logical decomposition of tasks is reflected in the selection, consistency and grouping of elements in the UI obtained. The new context of use implies reconfigurations of the UI that are beyond the traditional UI changes (Eisenstein et al., 2001), 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. The technique of automatically selecting an appropriate interactor while considering screen resolution constraints has already been investigated and shown to be feasible (Eisenstein & Puerta, 2000).

The process of generating a user interface in a model-based system can be seen as that of finding a concrete specification given an abstract one (well-known as the "*mapping problem*") (Limbourg & Vanderdonckt, 2004). When we deal with the evolution of an application (in particular, the user interface) towards devices with limited interaction capacities (mobiles phones, PDAs,...) or new interaction modes (tactile screen, use of pen for pointing), the *mapping problem* is more complex, and especially, so is a possible generalization. The transformation process must take into account device restrictions, but without constraining the application usability in the adaptation process to the new context. Selecting the best interaction element (or widget), as well as the arrangement of them in the screen area, is becoming the main problem to be solved. The user interface development is a very subjective process, which depends a lot on the designer's experience, creativity and aesthetic opinions. Complete generalization of this process is complex and, in fact, there is not much success in this sense. Normally, developed applications are very closed in specific domains. Several systems have attempted to automatically generate user interfaces in a model-based environment (UIDE (Foley et al., 1991), Mecano (Puerta, 1996), Trident (Vanderdonckt &

Bodart, 1993). The idea of these systems was to try to automate as much as possible the interface generation process from a task model. However, these systems usually delegate to the designer the responsibility of selecting the interface final appearance. The system assists about possible alternatives applicable to a certain case.

Besides, there are several proposals that have tackled the *problematic of the conceptual modeling of work in-group applications*. These proposals come from different fields: (1) the Human-Computer Interaction community (the CTT notation (Paternò, 2004) and the GTA framework (van Welie & van der Veer, 2003) are some examples); (2) the CSCW systems and, in particular, *workflow* systems, as for example the APM notation (Carlsen, 1998); and (3) the Software Engineering (as for example the COMO-UML (Garrido, 2003). The study of these proposals has allowed us to detect the following limitations: (a) the need for *theoretical and computational models* that allow specifying the activities in group supported by a computer in a suitable way; (b) the lack of notations that allow MODELING the existing *differences between cooperative and collaborative tasks* accurately, according to Dillenbourg's remarks<sup>1</sup> (Dillenbourg et al, 1995); and (c) the scarce variety of *notations* that support in a joint way interactive and work in-group aspects. These limitations cause the semantics of the specifications of collaborative applications to be incomplete and confirm the lack of a methodological framework supported by a coherent set of notations for modeling and designing collaborative tools of interactive nature.

### 3. From desktop support to mobile support: A Model-Based Evolution Process

With the aim of obtaining an appropriate *support for the development of multi-platform CSCL applications* we propose to use a *model-based approach* for the development of interactive groupware applications, called CIAM. 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 collaborative learning systems based on desktop metaphor towards mobile support. Also we are interested in the identification of 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.

#### 3.1 CIAM: A methodological approach for modeling interactive groupware applications

In this section the CIAM (*Collaborative Interactive Application Methodology*) proposal is briefly presented. CIAM is a methodological approach for the development of CSCW applications that takes into account the modeling of group work and interaction issues. Unlike other existing proposals in the fields of conceptual modeling of CSCW systems and modeling of issues related with the Human-Computer Interaction, CIAM considers the joint modeling of both issues, as well as the differentiation between the concepts cooperation and collaboration. A more detailed description can be looked up in (Molina et al, 2009).

---

<sup>1</sup> These differences affect the division of tasks, the roles participation in the tasks and the product obtained as a result of a joint activity.

The most important elements of CIAM are the following ones: (1) A *methodological approach* that defines the set of phases that compose the proposal, as well as the set of specification techniques to use in each of them (figure 1); and (2) A *notation*, called CIAN (*Collaborative Interactive Application Notation*) that allows expressing the peculiarities of the interactive groupware systems by means of a conceptual modeling (figure 2).

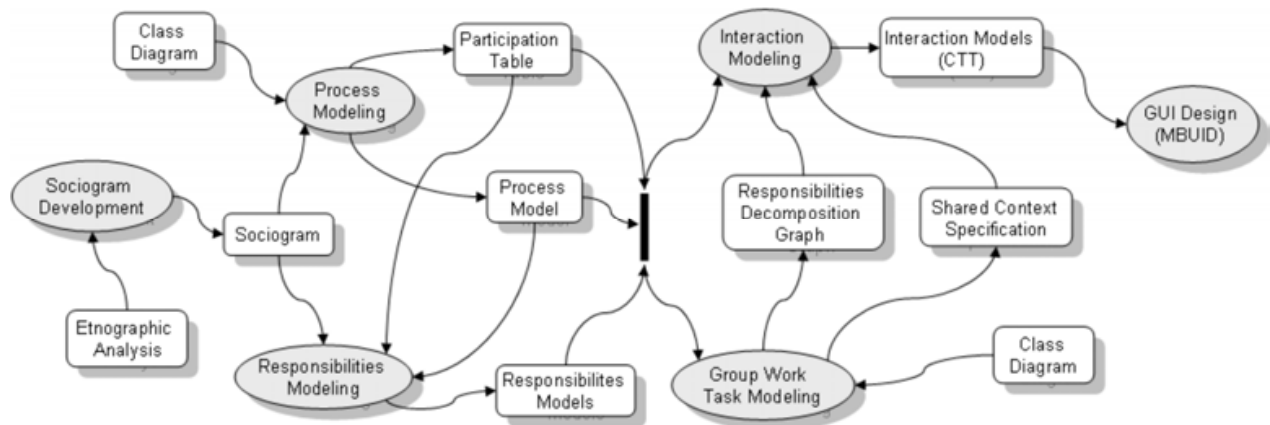


Fig. 1. The CIAM proposal and the products obtained in each of its stages.

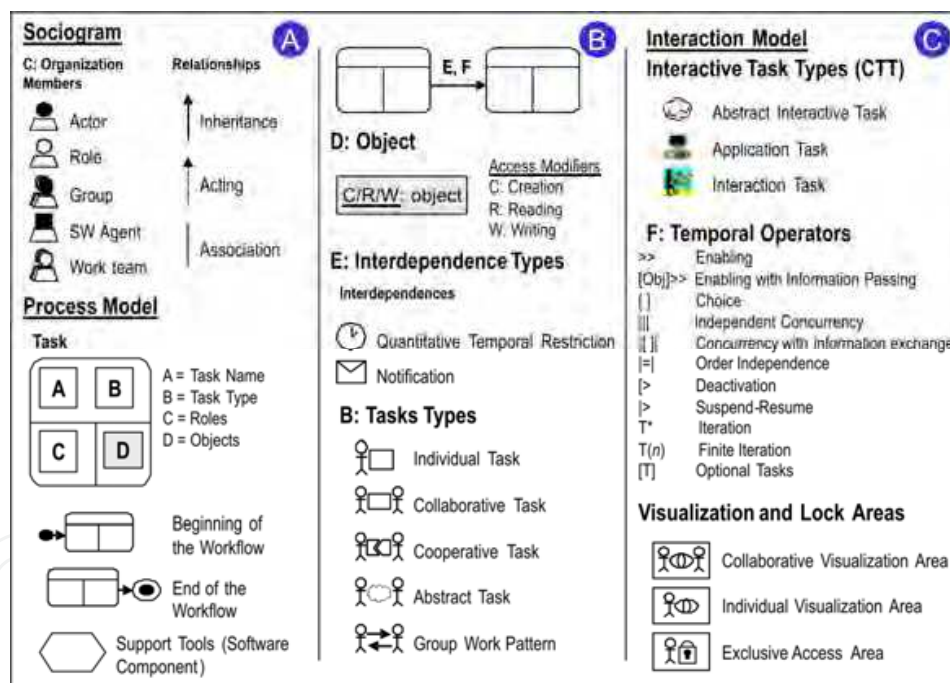


Fig. 2. CIAN Notation

The stages in CIAM and their objectives are enumerated next:

- In the *Sociogram Development* stage the organizational structure is modeled, as well as the relationships that exist between its members. The members that form the organization are in one of the following categories: *roles*, *actors*, *software agents*; or groupings of the previous ones, giving rise to *groups* or *work-teams*. Elements in this diagram can be interconnected by means of three kinds of basic relationships: *inheritance*, *performance* and *association*.

- In the following two stages (*Responsibilities Modeling* and *Process 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. Once the main tasks that characterize the group work and the responsibilities for each role have been defined, it is created the *process model*. This model uses a diagram (a graph) that allows relating all the information defined (the roles, the objects generated and accessed, and the main tools used for supporting the work in-group) by means of the two previous techniques. It shows the order of accomplishment of the tasks.
- In the *Group Work Task Modeling* stage the group tasks identified in the previous stages are described in more detail. Two different kinds of tasks are distinguished and modeled in a distinctive way: the *collaborative tasks* and the *cooperative tasks*. The *cooperative task modeling* uses the called *Responsibilities Decomposition Graph*. The notation used for cooperative task modeling is similar to the one used in the stage of creation of the *process model*. This way, we maintain coherence in the notations. The nodes of the graph represent individual tasks in which a single role is involved. The *collaborative tasks* specification is based on the *shared context* definition (Ellis et al, 1991), that is, the set of objects that are visible to the users and the actions that can be performed.
- Finally, in the *Interaction Modeling* stage the interactive aspects of the application are modeled. 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. As for the collaborative tasks, CIAM allows obtaining the interactive model directly from the *shared context* definition. The *interaction model* obtained facilitates the obtaining of the final UI.

## 4. A case study: The learning of Domotics

### 4.1. The starting system: Domosim-TPC

In Spain the new regulation for *Technical Training* (“Formación Profesional” in Spanish) 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. 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 (Redondo & Bravo, 2006). Using the DomoSim-

TPC system (figure 3), 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.

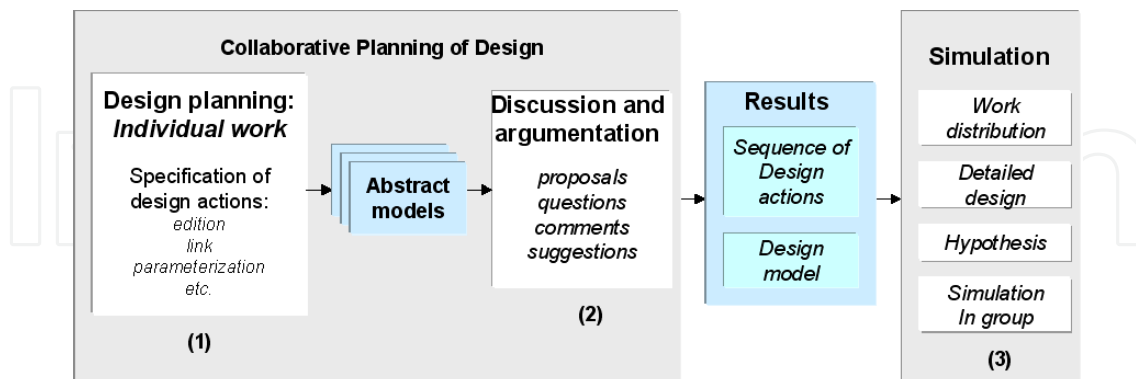


Fig. 3. Stages and tasks of domotical design learning carried out in an experience with DomoSim-TPC.

The *collaborative planning of domotical design* is divided in several phases, supported in three different workspaces in DomoSim-TPC:

- *Individual Planning stage* (figure 4). 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.

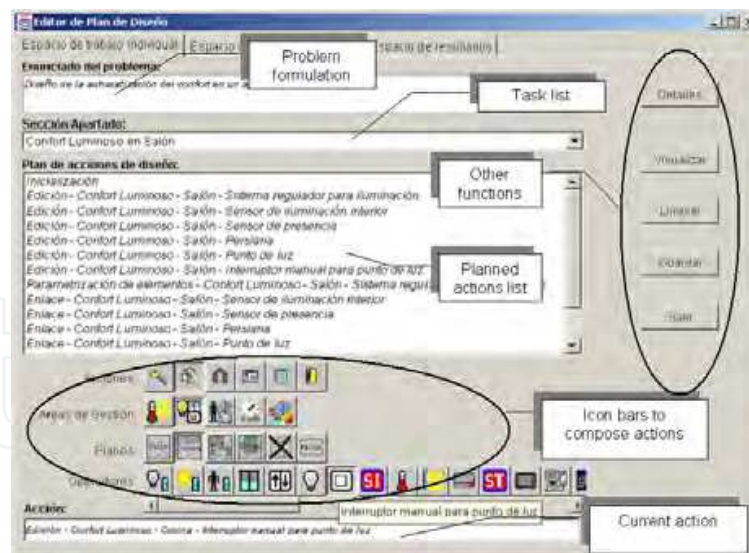


Fig. 4. Plan Editor in DomoSim-TPC

- *Discussion Stage* (figure 5). The discussion process is a social task in which participants in an activity reflect about the model built, exchange ideas, propose a resolution mechanism, acquire new knowledge and purify their contributions. From this process a proposal (model) is obtained, reflecting the viewpoint of each participant.

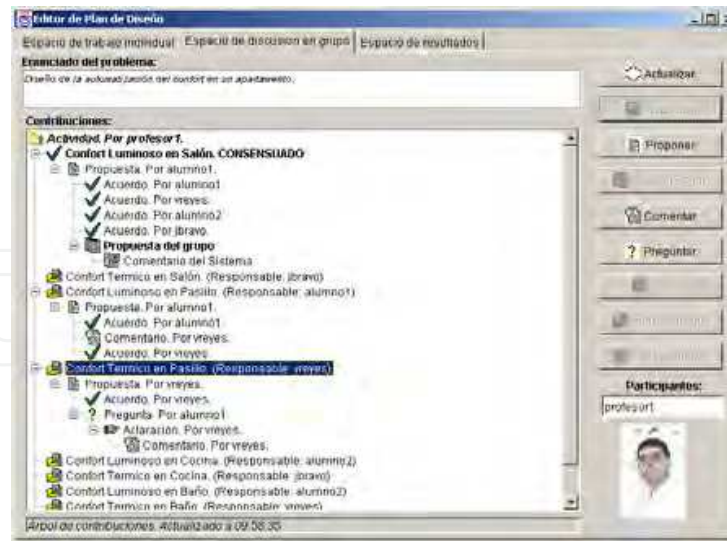


Fig. 5. Discussion workspace in Domosim-TPC

- Result stage.* Finally, the space of results 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 of the development process necessary for their achievement (this is the main goal of a *groupware* tool). Alike the discussion space, the results space has a tree-shaped hierarchical structure, used to organize the information relative to the results obtained in the activity. The application allows visualizing the results in a textual and a graphical way.

We are interested in the effective use of 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.

#### 4.2. Applying CIAM for adapting collaborative tasks in Domosim-TPC to mobile computing

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 proposed 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 CIAM notation. The models, created in the context of CIAM, allow adapting the user interface of Domosim-TPC to mobile computing support. Also, these conceptual models are used for automating and generating this evolution process.

First we are going to explain the creation of the *sociogram* and the *process models* associated with the system Domosim-TPC. In figure In figures 6 and 7 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 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.

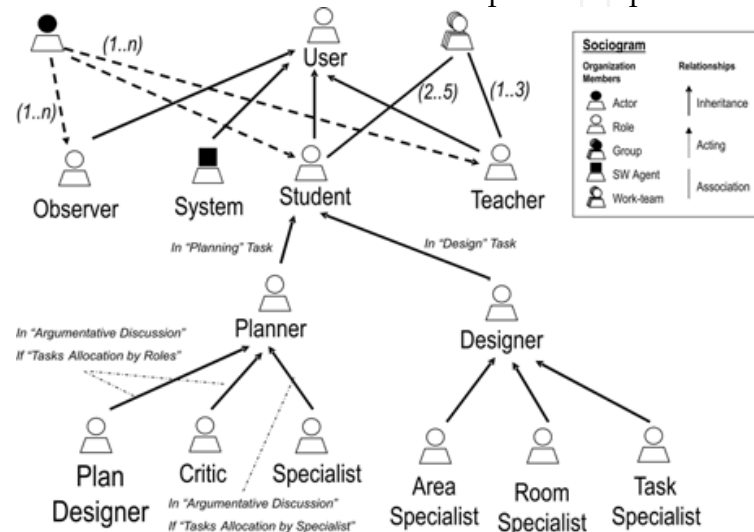


Fig. 6. Sociogram of Domosim-TPC

In the following two stages (*Responsibilities Modeling* and *Process Modeling* stages) the workflow that defines the work in-group developed in the organization is described. Figure 7 shows the *process model* associated to the system taken as example. In this example we use a symbol to express *Abstract Task* (7.d), that is, group work tasks that can be decomposed into others in a lower level of abstraction and of different kinds (7.g and 7.h). Collaborative and Cooperative tasks must specify the roles involved in its execution (7.a), whereas in the individual tasks only a role must appear. For all the tasks the objects manipulated and their access modifiers are indicated (7.b). For each task we can specify the so-called *Domain Independent Support Tool* (7.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 (7.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 (7.f) and a condition that must be checked (7.e).

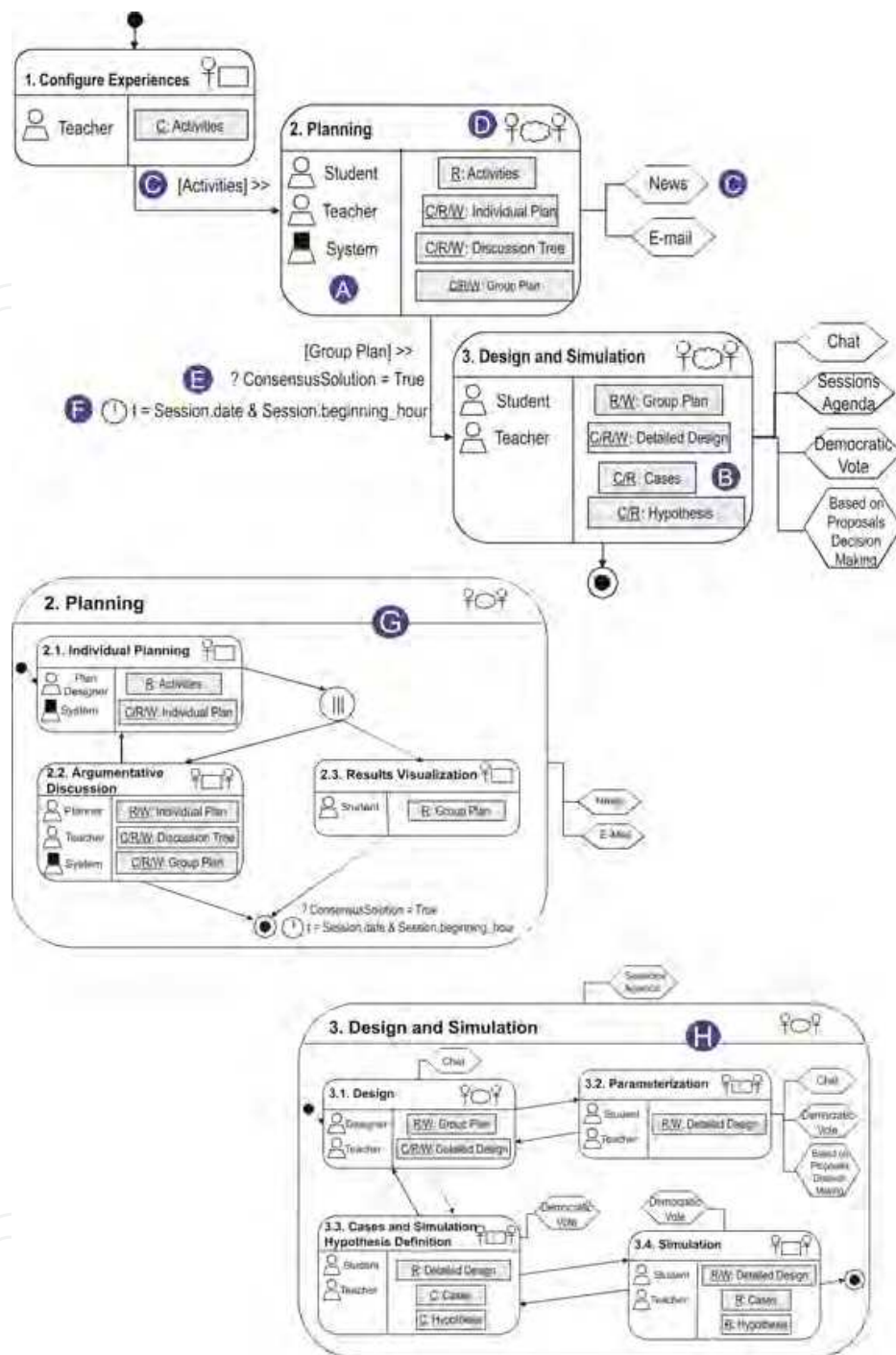


Fig. 7. Demosim-TPC Process Model

Figure 8 shows the *Interaction Model* (task model in CTT notation) for individual task “Individual Planning” (task 2.1 in figure 7). To obtain the version for PDA of the individual workspace, temporal 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 8 shows the names (in uppercase) of both objects. They are part of the name of the tasks that manipulate them.

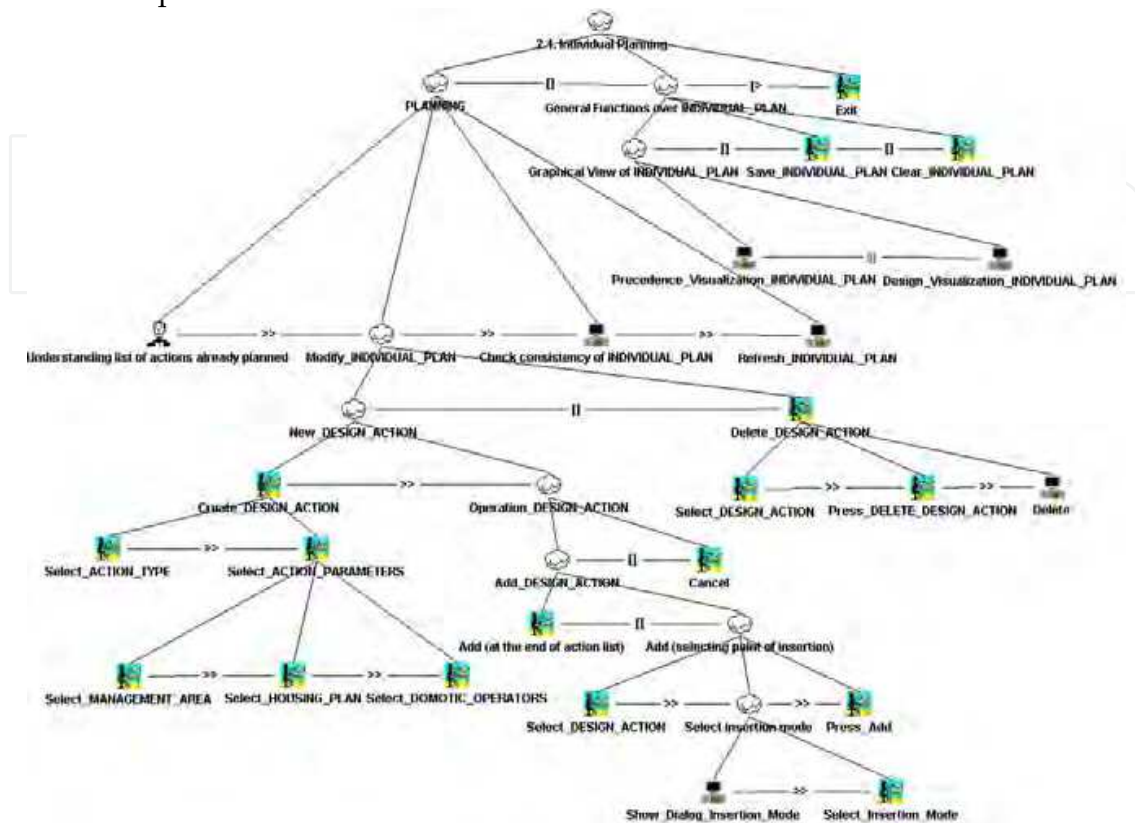


Fig. 8. Modeling of abstract task “2.1 Individual Planning”.

Diagram in figure 8 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 this subset of tasks is shown in figure 9 (a). In addition, the individual plan editor handles *DESIGN\_ACTION* objects. In the diagram shown in figure 8 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 9 (b). This dialog box appears whenever a new design action is created.

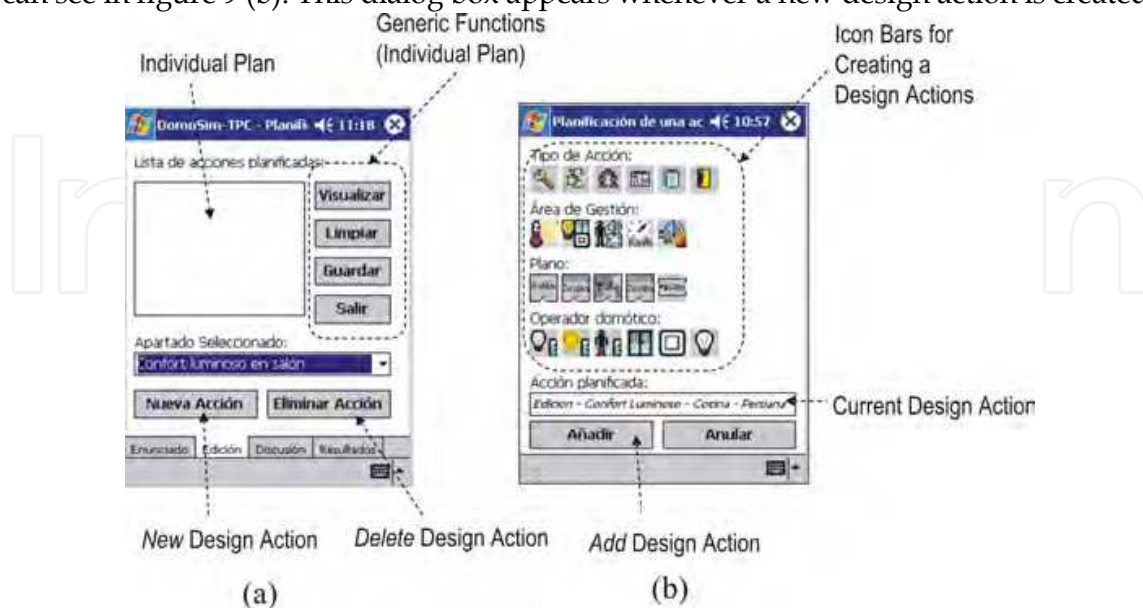


Fig. 9. PDA version of the interface to individual plan edition. (a) Interface that allows showing and performing actions on the *INDIVIDUAL\_PLAN*. (b) Dialog box that allows the creation of a new *INDIVIDUAL\_ACTION*.

### 4.3. The obtained system: Domosim-MOB

The system Domosim-Mob has been obtained as result of applying the aforementioned method for generating the PDA User Interface version of the Domosim-TPC system. It is important to point out that the functionality supported in PDA devices is a supplement of the original system. Domosim-Mob is an extension developed to add mobility to the learning process (*“learning anywhere and at anytime”*). Whilst mobility in learning processes favours the involvement of students in the task in question, the use of these devices does, however, mean paying the price of visualization limitations. In figure 10 we can see the appearance of the user interfaces for supporting the *result stage*.

The tasks supported in Domosim-Mob are similar to Domosim-TPC tasks, but the access to them has been improved, as a consequence of there being greater flexibility. However using mobile devices, beside the features characteristic of the discussion process, creates additional problems due to the fact that it offers the possibility of participating in the discussion without the need for the device to be connected to the server: that is the synchronization with the rest of the members in the group. One-to-one synchronization, in other words, synchronization of a mobile device with the system, is quite well resolved. However, in our case a synchronization of order  $n$  to  $m$  is necessary. Our approach to solving the problem is by applying a variation of the Schiper-Eggli-Sandoz algorithm for the exchange and ordering of messages in distributed systems respecting a causal order based on a model of logic clocks. The diagram in figure 11 shows the global structure of the Domosim-Mob application. The new context of use entails the appearance of several work modes in the application:

- *Connected work session* (Figure 11-1). The mobile application accesses the information server, although not at the same time as other participants of the group. The information associated with the group process and the history of the group is recorded and filed at a central location. Information exchange is based on XML files.
- *Disconnected work session (off-line)* (Figure 11-2), in which the user works with local information available in the mobile device. Students can work in isolation. They reflect on and construct the solution of the problem, which is stored in the PDA.
- *Synchronization session* (Figure 11-3). This one appears as a result of the previous work mode. Students send updated information to the server (individual contributions in the discussion process and their answer to the proposed problem) with the aim of providing and storing consistent information about the collaborative experience. The server distributes this information to the rest of the members of the teams. In order to guarantee confidentiality when accessing the server, an authentication process is necessary.

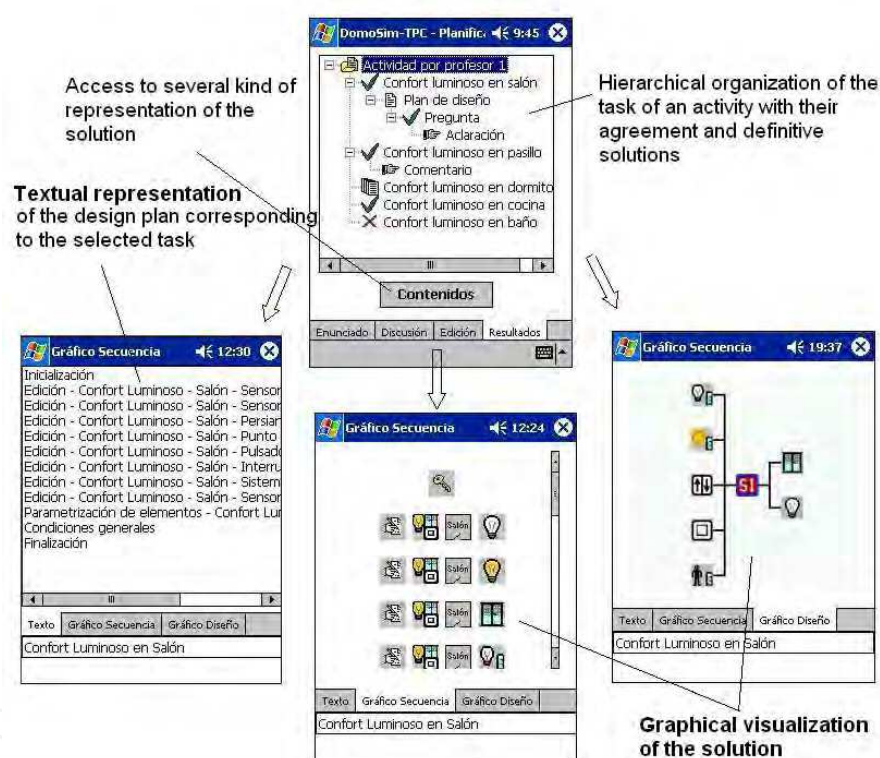


Fig. 10. PDA version of the interface to result visualization

## 5. Conclusions and Future Works

The main goal of the work presented 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. There are a lot of applications based on desktop metaphor that can be adapted to mobile computing support. We propose to use a model-based approach for the development and evolution of the graphical user interface of CSCL applications. Using a model-based approach allows the production of device-independent specifications of starting systems in this evolution process. In particular we propose to apply

the specification techniques proposed in the context of CIAM. CIAM is a methodological approach that guides the designer following different phases from modeling to reaching interaction models of groupware systems. 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 a CSCL system, called Domosim-TPC. We have obtained a mobile version of this system, called Domosim-Mob.

The results in the application of this method 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.

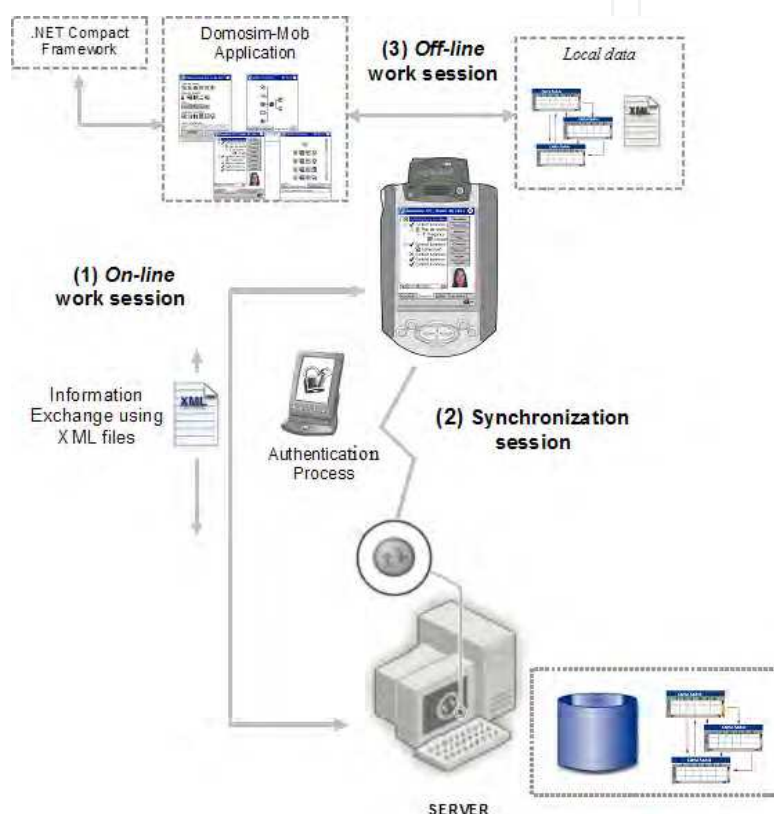


Fig. 11. General structure of Domosim-Mob

## 6. Acknowledgments

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

## 7. References

Abrams, M.; Phanourious, C.; Batongbacal, A.L.; Williams, S. & Shuster, J. (1999) UIML: An appliance-independent XML user interface language. In A. Mendelson, editor, *Proceedings of 8th International World-Wide Web Conference WWW'8* (Toronto, May 11-14, 1999), Amsterdam, Elsevier Science Publishers.

- Annett, J. & Duncan, K.D. (1967). Task Analysis and Training Design, *Occupational Psychology*, 41, pp. 211-221.
- Carlsen, S.: (1998) Action Port Model: A Mixed Paradigm Conceptual Workflow Modeling Language. *3rd IFCIS International Conference on Cooperative Information Systems*.
- Dillenbourg, P.; et al., (1995) The Evolution of Research on Collaborative Learning, ed. P. Reimann, Spada, H. Vol. Learning in humans and machines. *Towards an interdisciplinary learning science*. pp. 189- 211.
- Eisenstein, J. & Puerta, A. (2000). Adaptation in Automated User-Interface Design. *Proc. of IUI'2000* (New Orleans, 9-12 January 2000), ACM Press, New York, pp. 74-81.
- Eisenstein, J.; Vanderdonckt, J. & A. Puerta. (2001) Applying model-based techniques to the development of user interfaces for mobile computers. in *ACM Conference on Intelligent User Interfaces IUI'2001*. Alburqueque.
- Ellis, C. A.; Gibbs, S. J. & Rein, G. L.. (1991). Groupware: Some Issues and Experiences, *Communications of ACM* 34(1) 38-58.
- Foley, J.; et al., (1991). UIDE-An Intelligent User Interface Design Environment, in *Intelligent User Interfaces*, J. Sullivan and S. Tyler, Editors. Addison-Wesley. pp. 339-384.
- Garrido, J. L. (2003). AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas, *PhD-Thesis*, Departamento de Lenguajes y Sistemas Informáticos, Granada, Universidad de Granada. Spain.
- Limbourg, Q. & Vanderdonckt, J.(2004). Addressing the Mapping Problem in User Interface Design with UsiXML. *Proceedings of the 3rd annual conference on Task models and diagrams (TAMODIA 2004)*. Czech Republic.
- Molina, A.I.; Redondo M.A. & Ortega, M. (2009). A methodological approach for user interface development of collaborative applications: A case study, *Science of Computer Programming* 74 (9), pp. 754-776
- Paternò, F. (1999). *Model-Based Design and Evaluation on Interactive Applications*. Springer Vergal, ISBN 1-85233-155-0.
- Paternò, F., (2004) ConcurTaskTrees: An Engineered Notation for Task Models., in *The Handbook Of Task Analysis For HCI*, D. Diaper and N.A. Stanton, Editors. LEA, Mahwah, NJ., p. 483-501.
- Puerta, A.R. (1996). The MECANO Project: Comprehensive and Integrated Support for Model-Based Interface Development". in *CADUI96: Computer-Aided Design of User In-terfaces*. Numur, Belgium.
- Redondo, M.A. & Bravo, C. (2006) DomoSim-TPC: Collaborative Problem Solving to Support the Learning of Domotical Design. *Computer Applications in Engineering Education*. Ed. John Wiley & Sons, vol. 4, N°1, 9-19.
- Vanderdonckt, J.M. & Bodart, F. (1993) Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. In *InterCHI'93*: ACM Press.
- van Welie, M. & van der Veer, G. C. (2003). Groupware Task Analysis. *Handbook of Cognitive Task Design*. E. Hollnagel. LEA., NJ, pp. 447-476.



## **E-learning Experiences and Future**

Edited by Safeeullah Soomro

ISBN 978-953-307-092-6

Hard cover, 452 pages

**Publisher** InTech

**Published online** 01, April, 2010

**Published in print edition** April, 2010

This book is consisting of 24 chapters which are focusing on the basic and applied research regarding e-learning systems. Authors made efforts to provide theoretical as well as practical approaches to solve open problems through their elite research work. This book increases knowledge in the following topics such as e-learning, e-Government, Data mining in e-learning based systems, LMS systems, security in e-learning based systems, surveys regarding teachers to use e-learning systems, analysis of intelligent agents using e-learning, assessment methods for e-learning and barriers to use of effective e-learning systems in education. Basically this book is an open platform for creative discussion for future e-learning based systems which are essential to understand for the students, researchers, academic personals and industry related people to enhance their capabilities to capture new ideas and provides valuable solution to an international community.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ana I. Molina, William J. Giraldo, Francisco Jurado, Miguel A. Redondo and Manuel Ortega (2010). Evolution of Collaborative Learning Environments Based on Desktop Computer to Mobile Computing: A Model-Based Approach, E-learning Experiences and Future, Safeeullah Soomro (Ed.), ISBN: 978-953-307-092-6, InTech, Available from: <http://www.intechopen.com/books/e-learning-experiences-and-future/evolution-of-collaborative-learning-environments-based-on-desktop-computer-to-mobile-computing-a-mod>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen