Journal of Universal Computer Science, vol. 14, no. 9 (2008), 1435-1446
submitted: 30/4/07, accepted: 19/2/08, appeared: 1/5/08 © J.UCS

CIAM: A Methodology for the Development of Groupware User Interfaces

Ana I. Molina
(Escuela Superior de Informática, Castilla – La Mancha University, Ciudad Real, Spain
AnaIsabel.Molina@uclm.es)

Miguel A. Redondo
(Escuela Superior de Informática, Castilla – La Mancha University, Ciudad Real, Spain
Miguel.Redondo@uclm.es)

Manuel Ortega
(Escuela Superior de Informática, Castilla – La Mancha University, Ciudad Real, Spain
Manuel.Ortega@uclm.es)

Ulrich Hoppe
(Faculty of Engineering, University of Duisburg-Essen, Germany
hoppe@collide.info)

Abstract: The design of the groupware systems is a progressively extended task, which is difficult to tackle. There are not proposals to support the joint modeling of collaborative and interactive issues of this kind of systems, that is, proposals that allow designing the presentation layer of these applications. In order to solve this lack we propose a methodological approach, based on a set of notations of both a graphical and a textual nature.

Keywords: groupware design, interaction design, model based design
Categories: D.2.1. D.2.2, H.5.2, H.5.3

1 Introduction

The development of CSCW (*Computer Supported Collaborative Work*) systems is not a trivial task, among other reasons, because of the multidisciplinarity of these systems [Grudin 92]. There are issues such as the support to modeling of cooperative procedures and spaces for the sharing of information. These issues become requirements to be considered during the development of this kind of systems. A detailed study of the existing alternatives in this field has allowed us to detect the need of modeling collaborative issues, and particularly, the lack of proposals that support a joint modeling of collaborative and interactive issues [Johnson 04]. These problems justify the need of a methodological framework supported by a coherent set of notations that give support to the design of collaborative tools of interactive nature. This situation has led us to propose the definition of a notation (that we have called CIAN, *Collaborative Interactive Applications Notation*) that allows the expression of collaborative and cooperative tasks in a distinctive way clarifying its differences, already pointed out by Dillenbourg [Dillenbourg 95]. The distinction among these kinds of tasks affect the division of the tasks, the participation of the different roles in the execution of the tasks and the product obtained as a result of the joint activity. This notation will be used to complete a methodological framework for the design of systems to support work in-group.

In this paper our methodological proposal is presented. This is based on the use of specific notations, to design interactive groupware applications. In section 2 a review of the main contributions in this field is shown, indicating the strong and weak points of each one. In section 3 our proposal is presented, explaining its several stages, and the issues that are specified in each one. Section 4 shows a case of use, which also presents the different notations used in each stage. Finally the conclusions extracted as a result of this work are outlined.

2 Related Works

There are several proposals that have tackled the problematic of the conceptual modeling of work in-group applications. These proposals come from the Human-Computer Interaction (HCI) community, from Software Engineering (SE) and from CSCW systems and, in particular, *workflow* systems.

Among the main contributions within the HCI field, we point out the CTT notation created by Fabio Paternò [Paternò 04], the GTA framework [van Welie 03], the CUAN notation [Pinelle 03] and TKS [Jonson 03]. The main disadvantage of these proposals is that they present difficulties to integrate themselves within Software Engineering methodologies and to be related in a natural way with its notations and software development processes. In most cases the support for modeling CSCW systems is based on the extension of existing notations for modeling individual interactive applications, by means of the incorporation of three new concepts:

- The use of a new kind of tasks in the models (*the cooperative task*)
- The possibility to indicate which roles do which tasks.
- The separation in models: cooperative and individual (one model for each role involved).

Within the field of CSCW and the specification of *workflow* applications, we point out the APM notation [Carlsen 98], taken as reference by other more recent notations [Trætteberg 02]. These proposals include concepts related with the interaction between the members of the organization, and the main activities that must be supported by the system. These proposals emphasize the consideration of the resources (data or tools) handled in the several processes or activities under development.

As for the approaches derived or completely fitted within the Software Engineering we find the framework i* [Yu 95], which includes a notation for objective-oriented modeling. This notation is used in disciplines like the requirements engineering or organizational processes modeling. Also we have studied the support for collaborative processes modeling using the UML standard notation or UML variations [Eriksson 00; de Cesare 06]. We want to point out the COMO-UML notation for modeling of cooperative issues [Garrido 03]. One of its main contributions is the inclusion of dynamic issues modeling, as well as a model of the organization to be supported by the system. This proposal allows the specification of a greater number of aspects than the rest of proposals that approach this subject. The proposals that come from the Software Engineering benefit from the extensive use of these techniques. Also these notations present a better connection with SE notations and processes. In this way the link with the processing and persistent-storage parts of the applications to develop is much more natural. Nevertheless, they forget the purely interactive aspects of these applications.

The study of these proposals has allowed us to detect the following limitations:

- The need for *theoretical and computational models* that allow specifying the activities in group supported by a computer in a suitable way.
- There are not any *notations* that allow modeling the existing *difference between cooperative and collaborative tasks* accurately.
- There are not *notations* that approach in a joint way interactive and work in-group aspects.

These limitations cause the semantics of the specifications of collaborative applications to be incomplete.

3 Methodological Approach

In this section we present the several stages that compose our methodological proposal. This approach is called CIAM (*Collaborative Interactive Applications Methodology*). This proposal implies the adoption of different viewpoints when tackling the creation of the conceptual models that describes this kind of systems. The first stages tackle a group-centered modeling, going on in subsequent stages to a more process-centered modeling (cooperative, collaborative and of coordination), approaching, as we go deeper into the abstraction level, towards a more user-centered modeling, in which interactive tasks are modeled, that is, the dialogue that occurs between an individual user and the application. The two first stages of modeling allow the definition of the context in which the interactive model will be created, and serve as starting point for the last one. The information specified 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.

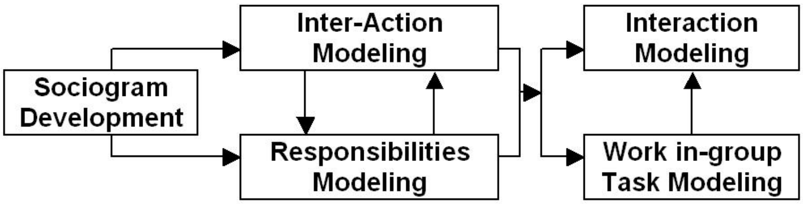


Figure 1: CIAM methodological proposal stages.

The stages which this proposal consists of (see Figure 1) and their objective are enumerated next:

1. *Sociogram Development*. In this 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*. The elements of these diagrams might be interconnected by means of three kinds of basic relationships (*inheritance*, *performance* and *association*).
2. *Inter-Action Modeling*. In this phase, the main tasks (or processes) which define the work in-group developed in the organization defined in the previous stage are described.
3. *Responsibilities Modeling.* In this phase, attention is paid to the individual perspective of each role of the organization, adding to its shared responsibilities those that are exclusive to it. The information specified in this stage is supplementary to the one in the previous stage, being necessary for both models to be coherent.
4. *Work-in group Tasks Modeling*. In this phase, the group tasks identified in the previous stage are described with a greater level of detail. We distinguish two different kinds of tasks, which need to be modeled in a distinctive way: *cooperative tasks* and *collaborative tasks*.
5. *Interaction Modeling.* In this last stage, the purely 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.

In the following section we present an application example of this methodological proposal and each of its stages is commented.

4 Application Example

In this section a complete example of application of our proposal for the design and specification of the presentation layer of groupware systems is presented. For each of the previously presented stages, we will show the models obtained and we will explain the elements included in the proposed notation. Next the problem statement is formulated:

Problem Statement

We tried to develop a system of a virtual administration office that supports the processing of the Degree Ending Project (DEP) in a University School. The definition of the process is the following:

The student and the director draw up a first draft plan in a joint way and propose the composition of the examining board. Then the student presents the draft plan, as well as the instance of evaluation request and the proposal of examining board in the administration office of the center. The student must wait for the evaluation of the Academic Committee. This evaluation could be positive, in which case the composition of the examining board will be appointed. This resolution must be notified to the director and the student. Also the members of the examining board will receive the corresponding notification. Once the draft plan approved, three months must pass at least so that the DEP can be defended. If the Academic Committee rejects the draft plan, it can suggest the corresponding changes. Next the student will be able to carry out the project, after registering it before the examination. Once the student has finished the proyect, he must present four copies of it in the Administration Office of the Center, with the authorization of the director of the project. Next, and when the three necessary months have passed, the president of the examining board will announce the examination. Once celebrated, the project must be marked. This mark could be positive (C: passed, B: good, A: excellent or honour degree), which finishes the process, or negative (D: failed), in which case it will be necessary to review the project. In this case, it is not necessary to pass through the process of presentation and evaluation of the draft plan again.

If we want to obtain the user interface which supports this virtual administration office starting from a model of interactive tasks such as, for example, CTT (which is one of the most widespread and even taken as a starting point in processes of automatic generation of user interfaces, [Luyten 03; Mori 04]), there are some issues that cannot be modeled correctly. For example, it is difficult to model the conditional iterations, or the decisions on which a certain condition occurs depend. Also the passage of time or the notifications are not contemplated by this notation. The purely collaborative tasks (in which several roles can take part simultaneously) cannot be modeled in CTT either, since it is only possible to model cooperative tasks.

The models that are created in each stage of our proposal are described in the following subsections.

4.1 Stage 1: Sociogram Development

In this stage the structure of the organization is defined. The diagram includes relationships of *conditional inheritance* that imply that a role can be specialized (and, therefore, have greater number of responsibilities) under a certain condition. This situation does not occur in this example. The *cardinality* of each role is specified by means of an *acting* relationship between roles and actors (the discontinuous arrow in Figure 2). Also the *association* of roles that work together in some of the activies of the process can be specified (giving rise to the formation of workteams).

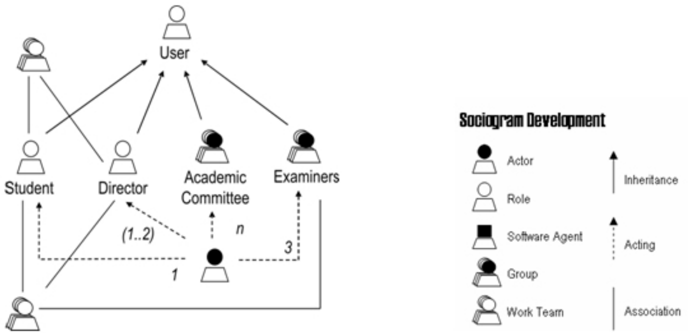


Figure 2: (a) Sociogram than represents the organization described in the example. (b) Notation symbols for describing the sociogram.

4.2 Stages 2 and 3: Responsibilities Modeling and Inter-Action Modeling

The two following phases can be approached indistinctly in any order, since the information contributed in the models generated in each is supplemented with the information that is specified in the other one, existing mutual feedback, with modifications and refinements. Even, they have to be revised in a cyclical way. In the *Inter-Action Modeling* stage (that shows the interaction/collaboration that takes place between the group members), creating the so-called *participation table* is of great help (see Table 1). Using this specification technique, of a textual nature, allows the designer to have one first idea about the division of the work at the highest level in abstraction. This table is composed of as many lines as tasks of greater level of abstraction have been identified by the designer and of so many columns as roles have been identified in the previous stage. A cell (T_i, R_i) will be marked when the role R_i is implied in the accomplishment of the task T_i. Once the appropriate cells marked, the last column is filled. This allows classifying the tasks identified in three categories, making use of a different icon for each type. These categories are in Table 2.

Roles **Tasks**	**Student**	**Director**	**Academic Committee**	**Examiners**	**Type**
Draft Plan Writing	X	X			웃□웃
Examining Board Proposal	X	X			웃□웃
Request	X				웃□
Academic Committee Valuation			X		웃□
Suggest Changes			X		웃□
DEP Development	X				웃□
Post-DEP Procedure	X	X		X	웃⧉웃
Examination	X			X	웃⧉웃

Table 1: Participation Table elaborated from the example statement.

Task Types	**Icon**
Individual Tasks	웃□
Cooperative Tasks	웃⧉웃
Collaborative Tasks	웃□웃

Table 2: Types of tasks handled by notation used in the methodology stages.

Once the participation table has been constructed, we will center on the *Responsibilities Model* definition. Information expressed by means of the previous techniques works as the basis for the definition of the responsibilities model associated to each of the roles of the system. Taking a reading by columns (by roles) of the previous table, we complete the tasks that each role must carry out, adding those that are of an individual nature and are not wrapped in the group work processes of the organization. This way, we can create a listing of responsibilities by each role detected, indicating for each one their nature (individual task, task carried out collaboratively or cooperatively). For each task the objects manipulated are specified, including the *access modifiers* to these objects (*R*, Reading; *W*, Writing; *C*, Creation; and any combination of these). Also, for each task the pre-requirements, which allow a satisfactory execution, are defined. The pre-requirements make reference to the tasks that should be completed before the current task, as well as the object/s in the data model must have been created by some role in the system previously. This way, we can establish dependencies of temporal (order) execution among the main processes, as well as the dependencies of the data to present. As an example, Table 3 shows the responsibilities model of the *Student* role.

Responsibility	**Task Type**	**Object in Domain Model**	**Pre-requirement**	
			Task	**Data**
Draft Plan Writing	웃□웃	C/R/W: Draft Plan	INI	
Examining Board Proposal	웃□웃	R/W: Draft Plan	Draft Plan Writing	Draft Plan
Request	웃□	R/W: Draft Plan	- Draft Plan Writing - Examining Board Proposal	Draft Plan
DEP Development	웃□	C/R/W: DEP	Academic Committee Valuation	Draft Plan
Post-DEP Procedure	웃⧉웃	R/W: DEP	DEP Development	DEP
Examination	웃⧉웃	R/W: DEP	Post-DEP Procedure	DEP

Table 3: Responsibilities Model of the Student Role.

Once the main tasks that characterize the work in group and the responsibilities for each role have been defined, we will create the inter-action model. This model allows specifying the complete operation of the group process that can be cooperative, collaborative or mixed. This model uses a diagram that allows relating all the information defined by means of the two previous techniques. This diagram is represented by means of a graph whose nodes are the activities that compose the work in-group and whose arcs indicate relationships between these activities (of order, data dependency, condition, notification, passage of time, etc). Each node includes the task name, its type, the roles involved in its execution and the objects manipulated. Each object is prefixed by its access modifiers (*C* indicates creation; *R*, reading and *W*, writing).

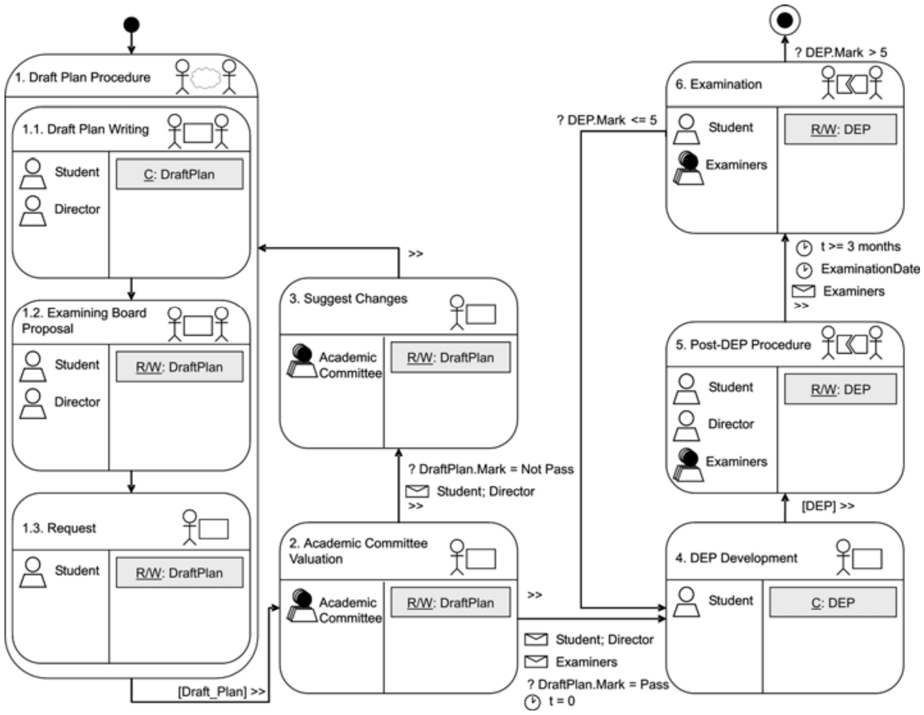


Figure 3: Inter-Action Model associated to the example

Figure 3 shows the inter-action model associated to the system taken as an example. The model includes nine main activities that compose the work in-group. As we had already identified (in Table 1), four of them are of individual execution, two of them are of a cooperative nature, with a clear division in subtasks that will be indicated in later stages of refinement (those labeled as 5 and 6). Finally, there are two tasks of a collaborative nature (labeled as 1.1 and 1.2). For all the tasks the objects manipulated are indicated, preceded by the corresponding access modifiers. In this case the work flow is sequential, although two conditional flows exist, being the temporal operator >> the more frequently used. The temporal operators which can be used in the inter-action model are the ones provided by CTT [Paternò 04]. The notation also allows specifying notifications, and as it happens when finalizing the task *Academic Committee Valuation.* Next to the icon that indicates notification we can specify the role/s that will be notified. It is also possible to specify the passage of time, by means of the corresponding icon. When it is explicitly necessary to indicate the passage of information from an activity to another, we can show the name of the object to transfer in brackets preceding the temporal operator used.

4.3 Stage 4: Work-in group Tasks Modeling

In this phase the level of detail, with which the previously identified group tasks (collaborative or cooperative) are specified, increases. It is important to highlight the necessity to model the cooperative tasks and the collaborative ones in a differentiated way. The outstanding information in each one varies. Considering the definition of Dillenbourg [Dillenbourg 95] this distinction is translated in two important aspects: the *division of tasks* (in individual tasks in the case of cooperation) and in the manipulated *objects* (which are shared in the case of the collaboration). As an example, we will show in detail the specification of a task of each type. In particular, we will show the modeling of cooperative task *Post-DEP Procedure* and the collaborative task of *Draft Plan Writing*. Figure 4.a shows the detailed **modeling of** the **cooperative task**. On the left we can see the roles involved and the objects manipulated. On the right appears the *responsibilities decomposition graph.* The notation used is similar to the one used in the stage of the creation of the *inter-action model*. This way, we maintain coherence in the notations. The nodes of the graph must represent individual tasks in which a single role appears involved. In this level the objects manipulated in the cooperative task are specified, associating them to the roles involved. **Modeling collaborative tasks** implies to know the roles involved in its execution and the objects of the data model that are manipulated in a shared way. For this, the specification of this kind of tasks is based on the definition of the shared context (this is, the set of objects that are visible to the set of users and the actions that can be performed on them). Figure 4.b shows the appearance of the specification of the collaborative task *Draft Plan Writing*. As in the cooperative tasks specification, the area on the left shows the roles involved, the objects manipulated and the access mode to these objects (reading and/or writing). The area on the right shows the objects of the data model manipulated constituting the shared context. For specifying the shared context we use the UML notation to which we add some icons to express visualization features (to the group or particular individuals) and the blockade of the objects that compose the shared context (see table 4). It can happen that a model does not include an area of individual visualization, as occurs in this example. This indicates that we are in a situation in which all the members that collaborate see exactly the same objects. If, in addition, they all see exactly the information in the same way, we would be specifying a situation in which the visualization is governed by the technique of strict WISIWYS. Adding an area of individual visualization entails relaxing the WISIWYS, possibly by means of the separation of work spaces (with a public space and a private area). Another of the aspects that is defined when we specify a collaborative task is the way in which its finalization is agreed. As for the *finalization policy of the collaborative task* in this example we have decided that it will be a *completion with an individual responsibility*. The *Director Role* will be the person in charge of ending the *Draft Plan Writing* task (it is represented graphically by means of an asterisk * plus the icon of this role).

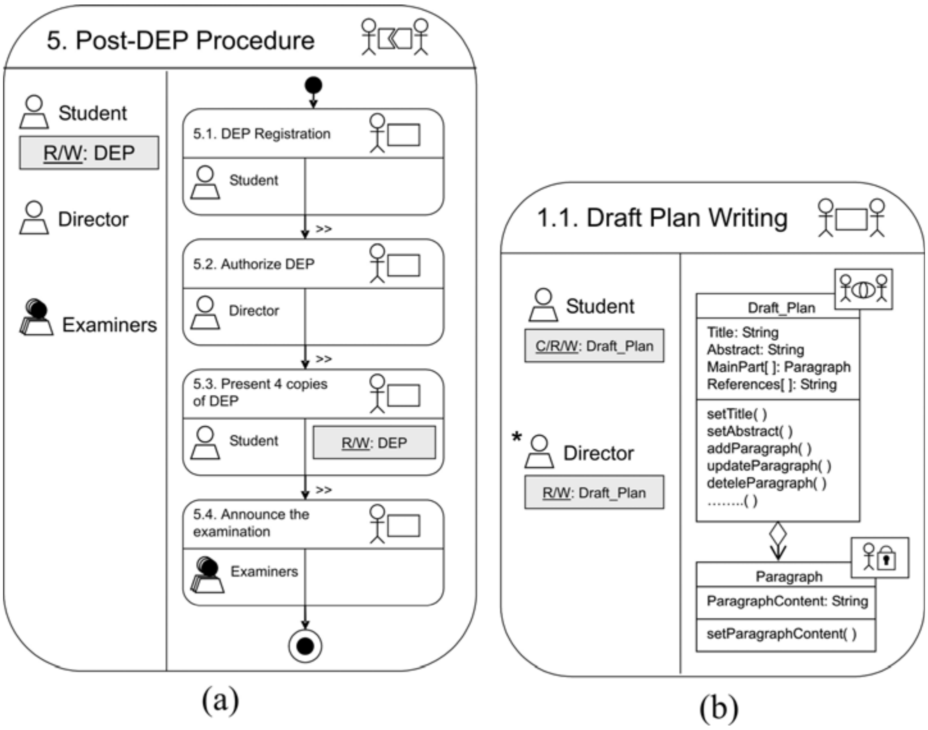


Figure 4: (a) Modeling of the cooperative task Post-DEP Proceduce. (b) Model of the collaborative task Draft Plan Writing.

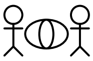
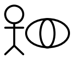
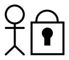
Icon	Definition
	Area of the shared context of collaborative visualization
	Area of the shared context of individual visualization
	Segment of the shared context of access of exclusive modification

Table 4: Icons for representing visualization features and exclusive access to the shared context.

4.4 Stage 5: Interaction Modeling

In this stage the designer centers on specifying the dialog that can take place among the users (individual users) and the applications (user interfaces) that mediate in the collaborative process defined. Thus, for each individual task it is necessary to create an interaction model. To create these models, the designer should identify the tasks that are initiated by the user when he/she interacts with the application (*interactive tasks*) and those that are carried out by the computer (internal processes or information visualization; that is, *application tasks*). In addition, for modeling the application dialog, the temporal order among the tasks must be specified. To model the interaction, a notation exists broadly disseminated in the HCI community. This language is CTT [Paternò 04], which we have already previously mentioned. Using this language, the models built present a hierarchical structure, in tree form that allows representing several levels of abstraction. Using CTT we can reach high levels of detail in the interaction model. This facilitates the obtaining of the final design of the user interfaces. In the particular case of collaborative tasks and using the models that we have constructed in previous phases, the CTT tree that models the interaction can be generated directly. For this, the definition of the shared context and the information relative to visualization and lock issues are used. Using CTT models in this last stage help us obtain the IU. This notation has been used as starting point in several proposals for obtaining and generating the UI in a semiautomatic way [Luyten 03; Mori 04].

5 Conclusions

Consulting the research dealing with the design of the interactive aspects of the applications and the work in group, we detected the lack of a proposal that allows *combining interaction, collaboration and information sharing aspects*. This paper presents a methodological approach, called CIAM, to solve this lack, as well as an application example. CIAM guides the designer when it comes to modeling the system, starting with specifications of higher level of abstraction which is diminishing when advancing in the process and it approaches the design of the final UI. The proposed notations give support to the differentiated modeling of cooperative and collaborative tasks. CIAM gives a more complete support, with greater capacity to represent semantics than the rest of the proposals that deal with the design of the presentation layer in CSCW systems.

Acknowledgements

This work has been supported by the Castilla – La Mancha University and the Junta de Comunidades de Castilla – La Mancha in the project GAMTest (PCI-05-005).

References

[Carlsen 98] Carlsen, S. Action Port Model: A Mixed Paradigm Conceptual Workflow Modeling Language. 3rd IFCIS International Conference on Cooperative Information Systems, 1998.

[de Cesare 06] de Cesare, S. and A. Serrano. Collaborative Modeling Using UML and Business Process Simulation. 39th Hawaii International Conference on System Sciences 2006 (HICSS'06), IEEE, 2006.

[Dillenbourg 95] Dillenbourg, P., M. Baker, et al. The Evolution of Research on Collaborative Learning. London, 1995.

[Eriksson 00] Eriksson, H. E. and M. Penker. Business Modeling with UML. Business Patterns at Work, 2000.

[Garrido 03] Garrido, J. L. AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas. Departamento de Lenguajes y Sistemas Informáticos. Phd. Thesis, Granada, Universidad de Granada, 2003.

[Grudin 92] Grudin, J. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. Groupware: Software for ComputerSupported Cooperative Work. Marca D. and G. Bock. Los Alamitos, CA, IEEE Press: 552-560, 1992.

[Johnson 03] Johnson, H. and J. Hyde. "Towards Modeling Individual and Collaborative Construction of Jigsaws Using Task Knowledge Structures (TKS)." ACM Transactions on Computer-Human Interacion 10(4): 339-387, 2003.

[Johnson 04] Johnson, P. Interactions, collaborations and breakdowns. 3th Task Models and Diqgrams for user interface design (TAMODIA 2004), Prague, Czech Republic, 2004.

[Luyten 03] Luyten, K., B. Creemers, et al. Multi-device Layout Management for Mobile Computing Devices, Expertise Centre for Digital Media – Limburgs Universitair Centrum, Belgium, 2003.

[Mori 04] Mori G., Paternò F., et al. "Design and Development of Multidevice User Interfaces through Multiple LogicalDescriptions." IEEE Transactions on Software Engineering August 2004: 507-520, 2004.

[Paternò 04] Paternò, F. ConcurTaskTrees: An Engineered Notation for Task Models. The Handbook Of Task Analysis For HCI. D. Diaper and N. A. Stanton. LEA, Mahwah, NJ.,: 483-501, 2004.

[Pinelle 03] Pinelle, D., C. Gutwin, et al. "Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration." ACM Transactions on Computer-Human Interacion 10(4): 281-311, 2003.

[Trætteberg 02] Trætteberg, H. Model-based User Interface Design. Dept. of Computer and Information Sciences, Norwegian University of Science and Technology, 2002.

[van Welie 03] van Welie, M. and G. C. van der Veer. Groupware Task Analysis. Handbook Of Cognitive Task Design. E. Hollnagel. LEA., NJ: 447-476, 2003.

[Yu 95] Yu, E. Modeling Strategic Relationships for Process Reengineering, University of Toronto, 1995.