Journal of Universal Computer Science, vol. 14, no. 16 (2008), 2680-2698
submitted: 9/7/07, accepted: 15/10/07, appeared: 28/8/08 © J.UCS

Designing Collaborative User Interfaces for Ubiquitous Applications Using CIAM: The AULA Case Study

Maximiliano Paredes
(Department of Computer Science Languages and Systems
Rey Juan Carlos University, Móstoles, Spain
maximiliano.paredes@urjc.es)

Ana I. Molina
(Department of Information Technologies and Systems
University of Castilla - La Mancha, Ciudad Real, Spain
anaIsabel.molina@uclm.es)

Miguel A. Redondo
(Department of Information Technologies and Systems
University of Castilla - La Mancha,, Ciudad Real, Spain
miguel.redondo@uclm.es)

Manuel Ortega
(Department of Information Technologies and Systems
University of Castilla - La Mancha,, Ciudad Real, Spain
manuel.ortega@uclm.es)

Abstract: In this article we explain how we apply the CIAM methodology based on the CIAN notation in order to generate user interfaces in collaborative applications. CIAM has been applied successfully in the development of desktop applications, such as Domosim-TPC, demonstrating its effectiveness in the definition of user interfaces for collaborative applications where a shared context is required. We present the AULA system modeled by means of CIAM. 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.

Keywords: CSCW, Human – Computer Interaction, User Interfaces, Ubiquitous Computing, Mobile Computing
Categories: H.5.2, H.5.3, K.3.1

1 Introduction

In the last years a great amount of collaborative applications have been developed. On the other hand only a few applications have been developed according to the paradigms of ubiquitous or mobile computing. Most of them were carried out in the same manner as other applications were 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 development and the perception of the context of the application.

From our point of view we need appropriate frameworks and tools (Methodologies, CASE tools, etc.) to help in the analysis and design processes of these complex applications in the same way as we already have them for the development of applications without these characteristics. In addition, we need to follow a process that considers the methods belonging to the Software Engineering methodologies but also we have to take into account the experience learnt from the Computer – Human Interaction approaches.

AULA is a collaborative and ubiquitous application intended to be an aid in language learning by means of a methodology called MECA. AULA has been developed to be used with PDAs inside and outside a language classroom. For this reason the synchronization of contents is required when the PDAs are online.

The learning process begins as follows: the students create a document structured in aspects and ideas. They use an editor with facilities for collaborative edition in an argumentative discussion process. The system has also different collaborative tools, such as a chat or an electronic mail, structured in *aspects* and *ideas*, and a set of linguistic tools to help in the edition of the texts.

In this article we explain how we apply the CIAM methodology based on the CIAN notation in order to generate user interfaces in collaborative applications. We apply this framework to a particular case: a foreign language learning system called AULA. For this, the paper is organized in the following way: section 2 introduces the CIAM methodological approach for designing interactive groupwork applications, presenting a brief explanation of its stages and the issues that can be specified in each. Section 3 explains the individual and collaborative writing model for foreign language learning. In section 4 a methodology called MECA, used for modelling collaborative writing learning, is presented. Section 5 explains the application of CIAN for modeling the system AULA (that implements the methodology MECA). Finally the conclusions extracted from this work are presented, and the future works we plan to follow are described.

2 CIAM Methodology

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 work in-group 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 [Dillenbourg, 95].

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 [Molina, 06].

- 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 [Molina, 07].
- A *notation*, called CIAN (*Collaborative Interactive Applications Notation*), that allows expressing the peculiarities of the interactive groupware systems.

In the figure 1 we can see the stages of the CIAM proposal. In each of them several collaborative and interactive systems issues are specified. The *Sociogram Development* stage allows to specify the social context in which the work in-group will be developed (roles, actors, work teams…). The following two stages allow the specification of the tasks of greater level of abstraction to be performed by the group (*Responsibilities Modeling*) and the temporal and data dependencies that exist among them (*Inter-Action Modeling*). In the *Work in-Group Tasks Modeling* the collaborative and cooperative tasks identified in previous stages are specified in a differentiated way and with a greater level of detail. The collaborative tasks specification is based on the *shared context* definition [Ellis, 91]. In the *Interaction Modeling* stage the interactive tasks to be supported by the Application User Interface to develop are specified. For this we use the CTT notation [Paternò, 04]. An interactive task tree will be created for each individual task or individual responsibility and for each work in-group task. In the case of collaborative tasks the interaction model is obtained from the shared context definition.

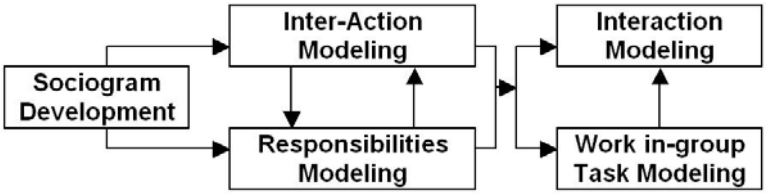


Figure 1: Stages in the CIAM methodological proposal

The models created in each of the stages of the proposal are specified using a set of graphical elements that are summarized in figure 2. On the top left (2.a) of the figure we can see the icons that represent the organization members (roles, actors, software agents, etc). On the bottom left (2.a) and the top centre (2.b) areas we can see the icons for representing the nodes that forms the Inter-Action Model and for indicating the several tasks and interdependences types. On the right area (2.c) we can see the icons used for representing an interaction task model in CTT notation. We have enriched this notation by means of the use of three new icons to express visualization features and blockade of the objects that compose the shared context in a collaborative task. A more detailed description of the notation CIAN can be found in [Molina, 06].

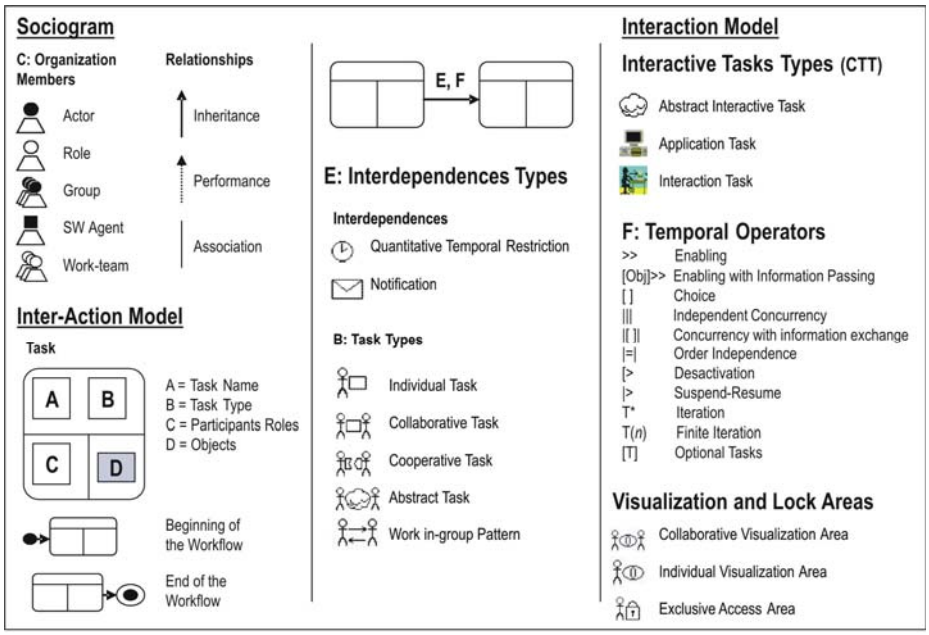


Figure 2: CIAN Notation

3 Individual and Collaborative Writing Model for Foreign Languages Learning

Writing individual process is composed of three sub-processes [Hayes, 80]:

- Planning. Author knows the writing subject and the context (writing tools, expression techniques, etc.). The author set goals and plans and define a w*riting project.* This *writing project* will direct the process.
- Coding. Author produces the text. This generation of text is based on the writing project and on the author's knowledge.
- Reviewing. Author reads the encoded text and improves the quality of the text produced.

This process is difficult in a collaborative writing environment. The collaborative writing is a task where a group of authors (co-authors) produce written documents. The co-authors share and discuss different ideas and have the same objective: creating a text. Sharples et al. [Sharples, 93] emphasize the collaborative writing complexity and identify some important issues, summarized in:

- Division of tasks and the work coordination strategy. The division of work and the coordination strategy are very important. There are three strategy types: Sequential, Reciprocal and Parallel.

- Work groups and communication. The members of the group can have different views. This discrepancy generates conflicts. We need to manage these conflicts adequately.
- External representation issues. We need structured representation of context. This structure represents the writing task status: generated text, objects, ideas, scheduling, etc. Therefore, aspects related with performance and control of versions are important.

Finally, the collaborative writing is composed of two stages: pre-process of writing (creation of group, planning, etc.) and process of writing (generation of text and discussion).

Collaborative writing is a pedagogical tool. In this approach, the students should write a text composition. In the classical scenario the students have a notebook and a pen and there is a blackboard in the classroom. The teacher writes the composition's title on the blackboard. Besides, the teacher specifies related information, for example the deadline to complete the task. Then, a text generation process is begun. The students individually write text on their notebooks. Therefore, each student generates composition fragments. Later, a discussion process is made. At this point, the students propose their fragments of composition. They write their text on the blackboard and the group discusses about acceptance or rejection of this text. At this point, the students have new ideas and they propose improvements or alternatives, fine-tuning the original proposal. Sometimes the group can decide to eliminate the proposal. Other times, the author of the proposal modifies his/her fragment of text and later the author proposes this change. The students know their mates' comments. They do all this work using the blackboard.

In this process, the student develops active and passive abilities [García, 04]. The students accept an active role in this process. The teacher observes the discussion and argumentation process. Sometimes, the teacher can participate along the process, clarifying and explaining some questions or issues. Therefore the teacher is a mentor in the discussion process. This process finishes when the students get to an agreement. In the last step of the composition, the teacher plays an active role. The teacher discusses the written solution (it is written on the blackboard) and highlights mistakes, improvements, extensions, etc. and the students write down. In this process, the student develops active and passive abilities. In particular, the students develop active abilities when they prepare their contributions and develop passive abilities when they read other contributions.

4 Applying MECA to design a learning model

MECA is a methodology for modelling the learning of collaborative writing [Paredes, 06]. The main aim of MECA is to help to efficiently structure the collaborative writing learning. The methodology identifies stages, agents and components generated by the agents. The MECA proposes six stages organized in three main categories:

- Pre-process. The learning activity is defined.
- Writing. The composition is performed.

- Analysis. Assessment of the activities performed by the students.

Next we are going to describe our learning scenario (a collaborative composition) guided by MECA. We need to consider some aspects of ubiquitous computing. We are going to emphasize the requirements of this scenario. The students use a PDA, which can be used anywhere at any time, while the teacher uses a PC. The students and the teacher use an electronic whiteboard and a PDA in the classroom. Therefore, some constraints are considered in this computer environment: time, space, multi-devices, and multi-user restrictions. MECA structures the lesson in the following main stages:

A) In the first stage the teacher should do the pre-process activity. This activity is accomplished before the beginning of the lesson and consists in defining the title and the type of composition (letter, report, request, etc.). Besides, the teacher should define the working group. MECA proposes small working groups (from 2 to 6 students). Later, the teacher should schedule the work session. This scheduling involves determining the date and length of the session. Finally, the teacher sends this information to the group of students (using electronic mail or face to face).

B) In the second stage, the students begin to work when they receive the information (the title and the type of composition). This process is individual and this process happens before beginning the session. The student has a PDA which has software tools in off-line mode. The student writes text on his PDA using text edition tools (sections and paragraphs of the composition). MECA proposes organizing this text in two types of information: aspects (they are titles of sections of the composition) and ideas (the paragraphs integrating an aspect). This task is usually performed outside the classroom and the PDA facilitates these actions.

C) In the third stage a session begins inside the classroom. The students propose their fragments of text to the classmates (beginning of the session) and the proposals are visualized on the whiteboard. These proposals are aspects and ideas written by the students (described in the above item). At this point, we need to start an information synchronization of the PDAs. Next, the discussion process begins. At this point the students discuss, propose, modify and argue their contributions. Now the students use on-line tools by means of their PDAs (for example, they use text edition utilities). The students agree or refuse proposals and the discussion process finishes. Later, the students should order the accepted proposals and the teacher assign the students' roles.

D) In the last stage the teacher reviews the composition generated by the students. MECA proposes quantitative and qualitative analysis methods. This analysis evidences work done by each student and shows collaborative process' conclusions, indicating work done inside and outside the classroom. The teacher assesses work and explains errors, improvements, etc. At this point, the composition activity finishes.

In conclusions, we identified context-awareness information (time, space and devices). We have to define synchronization strategies. Besides, we found context-sensitive actions, for example text edition. This can be collaborative (inside-the-classroom edition) or individual (outside-the-classroom edition).

We have applied MECA to language learning courses, in particular English as a Foreign Language (EFL), and we have implemented the AULA platform for this purpose. A detailed description of the system can be found in [Paredes, 03].

5 AULA Modeling using the CIAN notation

In this section the application of the CIAM methodological proposal and the use of CIAN notation, for the modeling of the AULA system are presented.

5.1 Social Structure in AULA

In the first stage of the CIAM methodology the social structure of the organization in which the groupware system to be designed will be implanted is modeled (the so-called *sociogram* of the organization is created). For this the different actors and roles of the system as well as their grouping (in groups and work teams) are identified. Figure 3 shows the sociogram of the AULA system. Two roles are identified: *student* and *teacher*. Both are specializations of a generic *user*, who interacts with the application. The Teacher role is in charge of defining the composition topic and making the planning of the work session. Also he/she will be the one in charge of facilitating and guiding the composition process. The *teacher* and *student* roles can form a work team in the context of some of the tasks supported by AULA (for example, in the process of discussion of the proposals). The work teams are formed by a set of *students* (minimum 2 and maximum 6) and a *teacher*. By means of the use of cardinalities these restrictions can be expressed. CIAN allows specifying the number of actors who can carry out a certain role. In this case an indefinite number of *students* and *teachers* is admitted, but at least an actor must exist for each role.

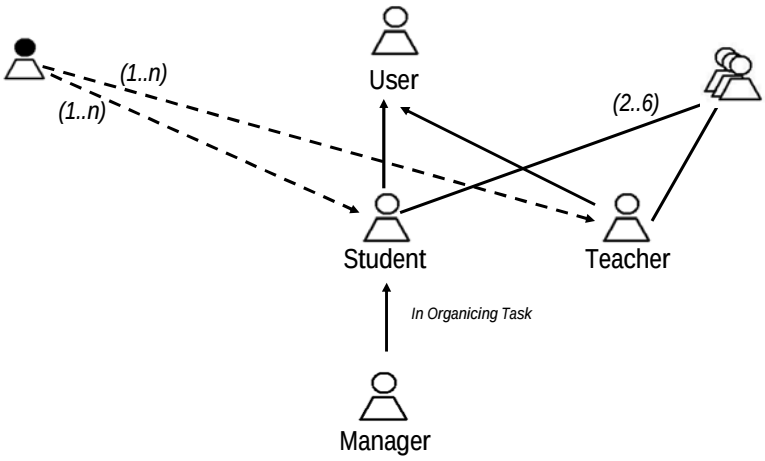


Figure 3: Sociogram of AULA

The notation also allows specifying role specializations in the context of certain tasks. In the diagram we can see as the *Student* role is specialized in the *Manager* role

in the context of the *Organizing Task*. When the discussion process has finished a student assumes the Manager role. This student is the one in charge of readjusting the composition. The system must activate this role when the student accesses the content organization tool.

5.2 Responsibilities Modeling

In this phase of the methodology the tasks of a greater level of abstraction to be supported by the system are identified. In AULA we identify the tasks shown in Table 1. This specification technique, which we have called *Participation Table*, allows the user to relate tasks and roles, as well as to specify the task type (individual, collaborative and cooperative).

Roles / Tasks	Student	Teacher	Type
Defining Topic		**X**	
Creating Activity		**X**	
Composing Personal Work	**X**		
Discussing Contents	**X**	**X**	
Organizing Contents	**X**		
Analyzing Solution	**X**	**X**	

Table 1: Participation Table

The tasks supported by AULA are the following ones:

- *Defining Topic*. (corresponding with the first stage of MECA; see paragraph A of section 4). The teacher role is the one in charge of performing this task. This task consists of defining the composition subject. This is an individual task. The teacher must define the following items: the *title* of the composition, the *composition type* (informal, administrative request, letter to a friend, etc), the *temporality* (duration of the work session) and some extra information or *observations.*
- *Creating Activity*. The work session is planned. In this task the teacher must specify certain information. Thus, for example, the teacher will indicate the date of beginning and end of the work session in the classroom. Also in this task the students group is defined and the decision policy of the group is chosen. When a proposal exceeds a certain number of *ok* votes (% of votes), the proposal is accepted. In this task this percentage of votes is specified. In the context of this task additional information related to the session can be

specified (classroom identifier, recommendations, date and hour of the chat sessions, etc.). This task is of an individual nature.

- *Composing Personal Work*. This task corresponds with the second stage of MECA; see paragraph B, section 4. The task called Composing Personal Work is performed by the student. The students, individually, write text fragments in their PDA. These text fragments will be proposed by the student in the classroom (in the work session). The student can do this task anywhere because the student has a mobile device.
- *Discussing Content*. This task is of a collaborative nature. This task is performed by the student and the teacher in the classroom (at the beginning of the session). The student proposes his/her text fragments (third stage of MECA; see section 4, paragraph C). At this point a discussion process begins: the students discuss and propose changes, alternatives, improvements, etc. The group will accept some proposals and will reject others. The teacher is the one in charge of facilitating this process.
- *Organizing Content*. This task is performed by a student. It is a task of an individual character. The student orders the text fragments accepted by the group. This task is taken over by the Manager role (see Figure 3).
- *Analyzing Solution*. (corresponding with the fourth stage of MECA; see paragraph D, section 4). This task is performed by the teacher and the students, in a collaborative way. First, the teacher reviews the text of the composition proposed by the students. Then, the teacher evaluates the knowledge of the students. Finally the students and the teacher review the composition and the teacher identifies errors, proposes improvements, etc.

Next, the responsibilities models of AULA are shown. Tables 2 and 3 show the *Responsibilities Models* of the teacher and student roles, respectively. Table 2 indicates the tasks assumed in an individual way by the teacher, as well as the tasks of work in-group in which he/she participates. The Responsibilities Model shows, for each task, its type, the manipulated objects (as well as the access modifiers to such) and the task pre-requirements (of execution and information). For example (Table 2), in the *Discussing Content* task the teacher must access the object *Proposal*. This task begins when the *Composing Personal Work* task finishes (this condition is expressed in the *Pre-requirements* column - *Task*). This task needs the object *Proposal* (we indicate this in the *Pre-requirements* column – Data) (Table 2). The *Defining Topic* task is the first task to be executed in the work-in group flow specified (it is indicated by means of the pre-requirement *INI*). By means of the information specified in the table we can see, for example, as the teacher role is the person in charge of creating the *Topic*, *Group*, *Activity* and *Experience* objects. The teacher also consults the *Proposal* and *Solution* objects, which are objects created by the students. The student is the one in charge of creating, in addition to these two objects, the *Feasible Solution* object.

Responsibility	Task Type	Object in Domain Model	Pre-requirements	
			Task	Data
Defining Topic		C: Topic	INI: Initial Task	
Creating Activity		L: Topic C: Group C: Activity	Defining Topic	Topic
Discussing Contents		L: Proposal	Composing Personal Work	Proposal
Analyzing Contents		L: Proposal L: Solution C/L: Experience	Organizing Contents	Solution

Table 2: Responsibilities Model of the Teacher role

Responsibility	Task Type	Object in Domain Model	Pre-requirements	
			Task	Data
Composing Personal Work		L: Topic C: Proposal	Creating Activity	Topic
Discussing Contents		L: Proposal C: Feasible Solution	Composing Personal Work	Proposal
Organizing Contents		L: Feasible Solution C: Solution	Discussing Contents	Feasible Solution
Analyzing Contents		L: Proposal L: Solution	Organizing Contents	Solution

Table 3: Responsibilities Model of the Student role

5.3 Inter-Action Modeling

The *Inter-Action Model* shows the order of accomplishment of the tasks. This model shows the roles, accessed and generated objects, and the main tools used for supporting the work in-group (for example, decision making tools, conversation, etc). Figure 4 shows the inter-action model supported by the AULA system. The three states shown in this model correspond with the three main stages in MECA (Process, Writing and Analysis). We use abstract tasks to handle the complexity of the created model. The learning activity begins in the *Definition* task (first node of the inter-action model). In this task the teacher creates the objects *Topic*, *Group* and *Activity*. The *Definition* task is an abstract task, formed by the tasks of a smaller level of abstraction, *Defining Topic* and *Creating Activity*. Next, the *Composition* task (second node) begins. This task begins when the *Definition* task has finalized and the *Topic* object has been created. The students read the title of the composition (the *Topic* object) and create the composition (the *Solution* object). The students use two types of auxiliary tools: tools related to the objective of the task (*Book, e-Dictionary* and *Composition Editor*) and social tools (*Chat, email, Decision making*, etc.). This task ends when the students reach a consensus by majority (? *MajorityAgreement* >=

Topic.Consensus). The teacher also takes part in the *Composition* task, facilitating its accomplishment. We have defined this task as an abstract task, since later its operation will be in detail. The last task is the *Analysis* task (third node). The students and the teacher study the proposals and the created solution (objects *Proposal* and *Solution* respectively). In addition, the teacher evaluates the work performed (he/she creates the *Experience* object). The learning activity ends at this point.

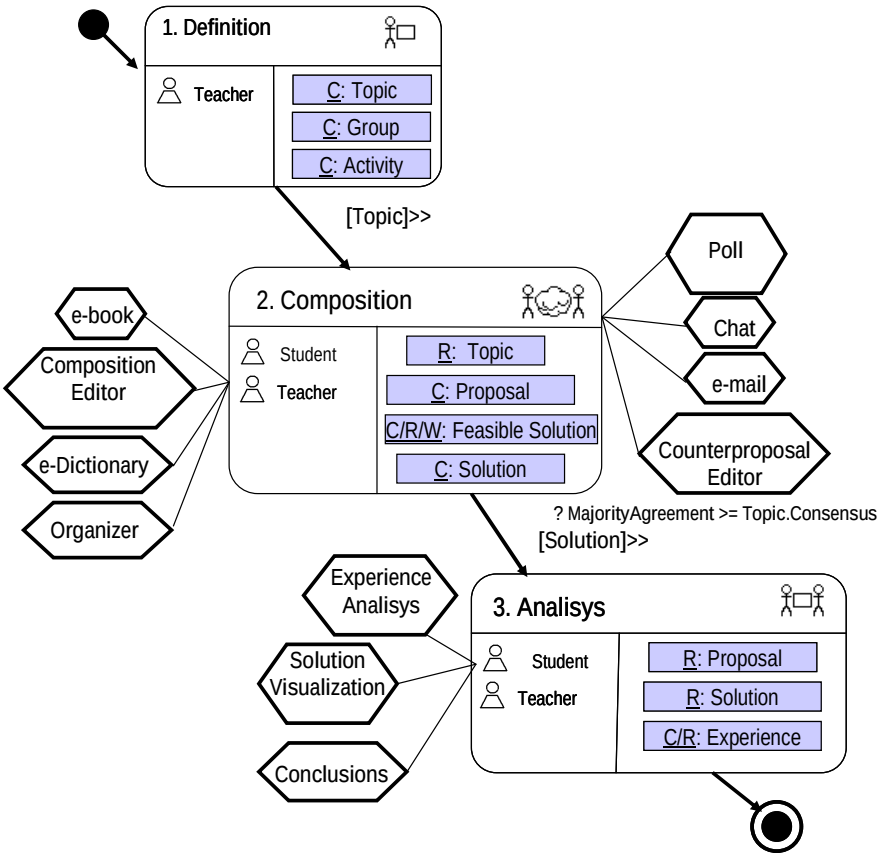


Figure 4: AULA Inter-Action Model (abstract level)

As we have previously mentioned, the abstract *Composition* task is the most complex task. Figure 5 shows this task in detail. The abstract *Composition* task is formed by two individual tasks and a collaborative task. The process begins when the students write their text fragments (the *Composing Personal Work* task). The students read the composition title (the *Topic* object) and write text (creating the new *Proposal* object). Later, a session begins in the classroom (*Discussing Content* task). This session must begin on a specific date and at a specified time (this is indicated by means of the expression *t=Topic.date* & *Topic.hour* in the input flow of the task). The

students create a final draft (the *Feasible Solution* object). This task finishes when most of the students decide what solution will be considered as the solution of the group (*MajorityAgreement >= Topic.Consensus*). Later, the task *Organizing Content* begins. A student (*Manager* role) organizes the text of the draft (*Feasible Solution*) and creates the final composition (*Solution*). Figure 6 shows the abstract *Discussing Content* task in detail. The three subtasks shown in the figure model the activities corresponding with the third stage of MECA (section 4, subsection C).The first subtask is the *Proposals Reading* task. The group reads the proposals and creates a first version of the draft (*Feasible Solution*). Then, the students and the teacher review and discuss this draft (by means of a discussion process) and the students create a new version (*Contributions Discussion* task). Next, the *Agreement* task begins. In this task the students must vote the draft. If the students reach an agreement the process finishes. On the contrary, if the students disagree, the draft must be modified, returning the execution to the *Contributions Discussion* task.

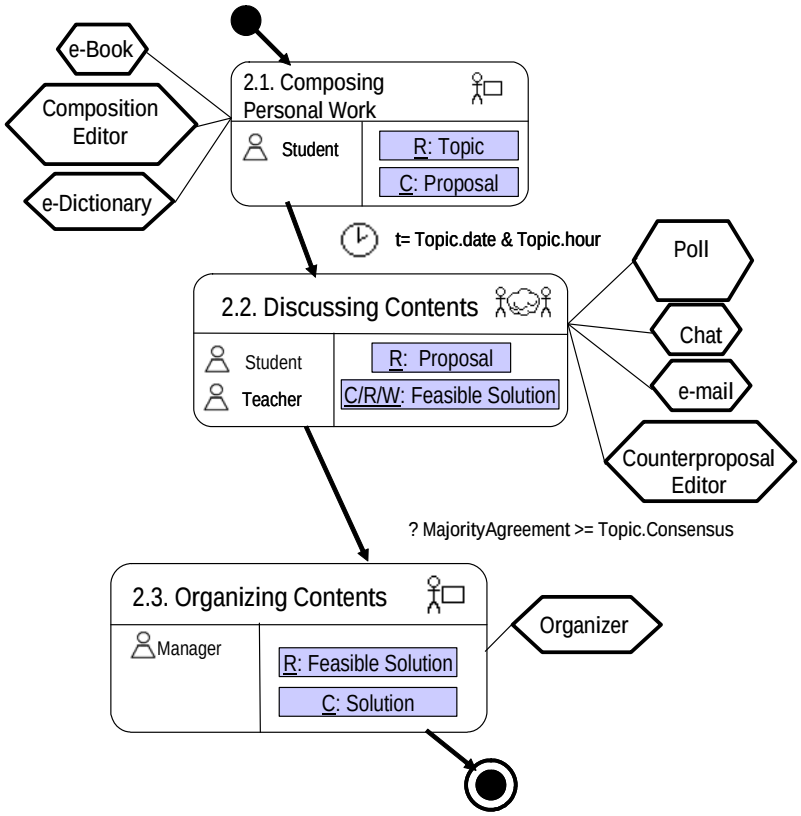


Figure 5: Detailed model of the abstract Composition task

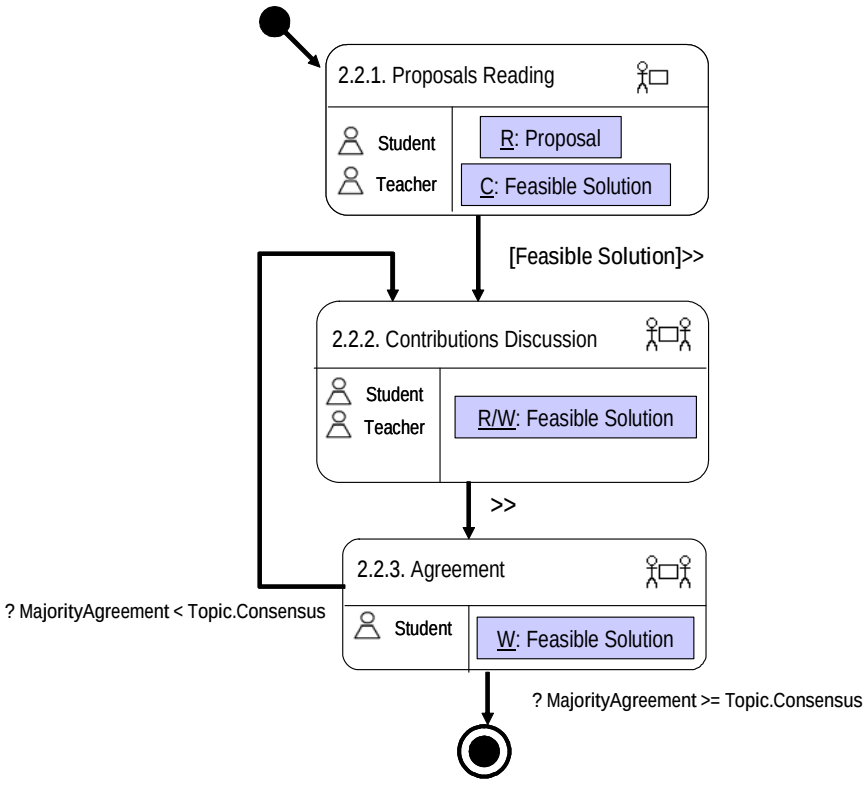


Figure 6: Discussing Contents Inter-Action

Figure 7 shows the appearance of the specification of the collaborative task called *Contributions Discussion*. Modeling collaborative tasks implies to know the roles involved in their execution and the objects of the data model that are manipulated in a shared way. The area on the left of the figure shows the roles involved in the task (student and teacher), the objects manipulated (*Feasible Solution*) and the access mode to these objects (reading and/or writing). The central area shows the objects of the data model manipulated that constitute the shared context. For specifying the shared context we use UML notation to which we add some icons to express visualization features and blockade of the objects that compose the shared context (see figure 2.c). In the shared context specification area we have the *Topic*, *Aspect*, *Idea* and *Arguments* objects. One *Topic* is composed of several *Aspects* and an *Aspect* is composed of several *Ideas*. The aspects and ideas can have arguments. These objects are acceded in an exclusive mode.

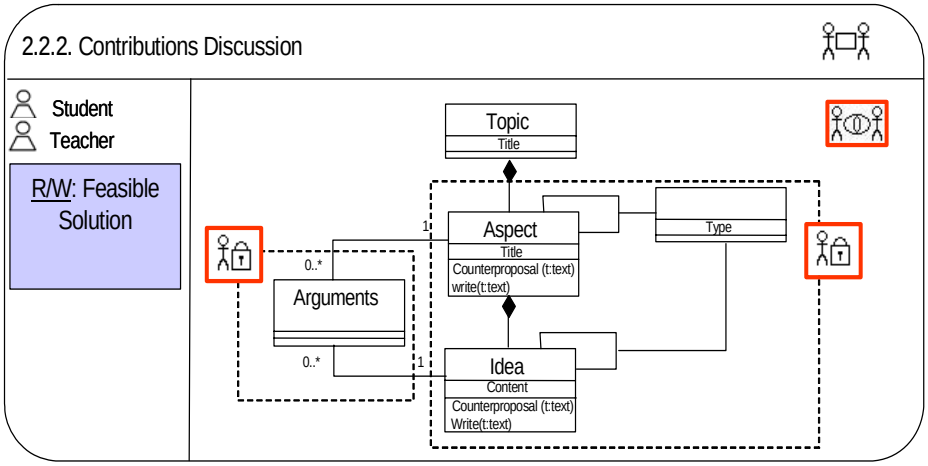


Figure 7: Contributions Discussion

5.4 Generation of User Interface

In this phase the designer is centered 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 (in the initial phase and in the refinement of the cooperative tasks). To create these models, the designer should identify the tasks that are initiated by the user when he 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, in order to model the application dialog, the temporal order among the tasks must be specified.

To model the interaction, a broadly diffused notation exists in the community of Computer Human-Interaction. This language is CTT [Paternò, 04] that we have previously mentioned. Using this language, the models built present a tree-form hierarchical structure that allows representing several levels of abstraction. Using CTT we can reach high levels of detail in the interaction model. This facilitates the realization of the final design of the user interfaces. In addition, 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 directly generated. For this, the definition of the shared context and the information relative to visualization and lock issues are used.

We have enriched the CTT notation with three new icons that represents three visualization areas (see figure 2). These icons are used as roots of the subtrees in the interaction tree in CTT notation separately: (a) the subtree that represents the interaction with the shared context that is common for all the members of a group involved in a multiuser task (*collaborative visualization*), (b) the interaction of individual nature for each member in the group (*individual visualization*) and (c) the subtree that specifies the dialog with the area of the shared context that can only be accessed exclusively by one member of the group at a time. Using our extension of

CTT we can identify additional information about the areas that compose the collaborative user interface. Thus, this extension has a higher-level semantics, which organizes and expresses in a better way specific aspects of collaborative applications. In figure 8 we have highlighted the visualization areas considered in the collaborative tasks studied.

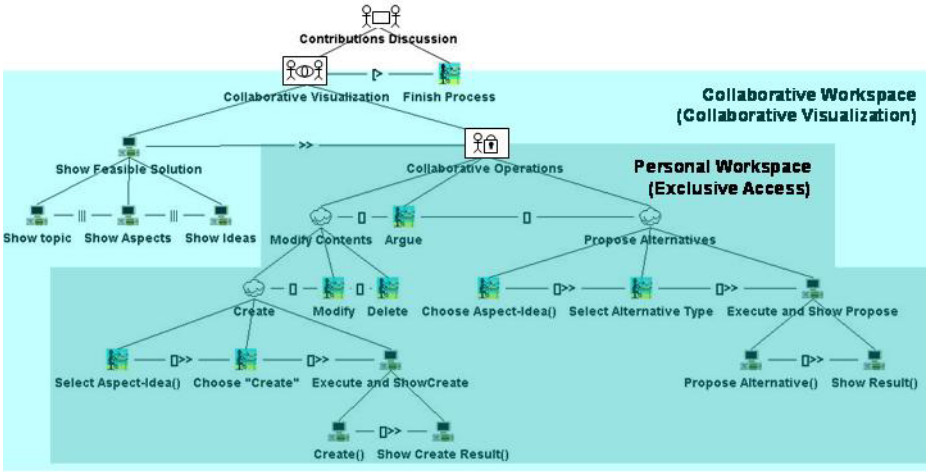


Figure 8: Modeling the Contributions Discussion task with an enhanced CTT Tree.

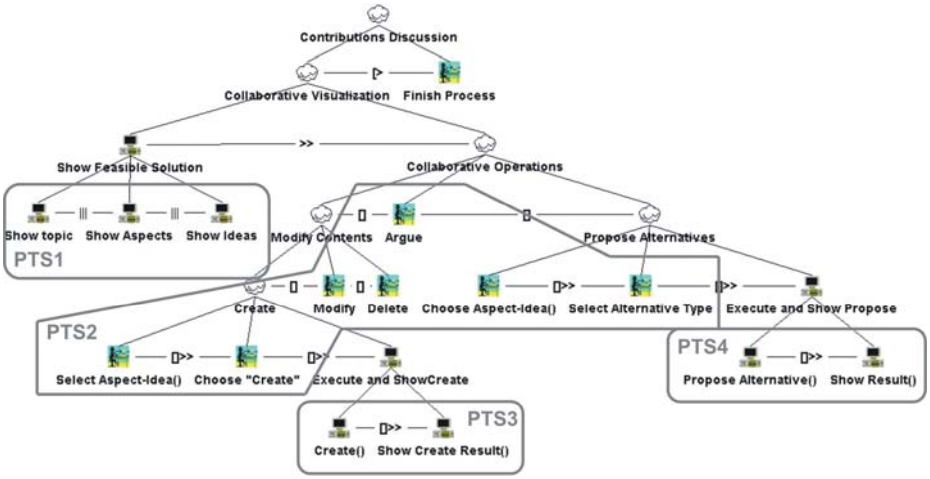
From this model the User Interface can be obtained in a semiautomatic way using proposals as [Luyten, 04, Mori, 04]. Figure 10 shows a possible User Interface for supporting the collaborative task studied. This is obtained using the techniques proposed by Paternò, in particular, the calculation of the Presentation Task Sets (PTS) and the application TERESA [Berti, 04]. Using PTS we can obtain the tasks set that must be enabled (and, in the context of the user interface design, visualized) on time. Using the information expressed by the PTS, the type of interaction and application tasks, the features of the object manipulated and some guidelines for preserving usability, we can select the most appropriate user interface techniques (*widgets*) for supporting the interaction tasks model.

The PTS obtained in this example are the following:

PTS1: {Show Topic, Show Aspects, Show Ideas, Finish Process}
PTS2: {Select Aspect-Idea(), Modify, Delete, Argue, Chose Aspect-Idea(), Finish Process}
PTS3: {Choose “Create”, Finish Process}
PTS4: {Create(), Finish Process}
PTS5: {Show Create Result(), Finish Process}
PTS6: {Select Alternative Type, Finish Process}
PTS7: {Propose Alternative(), Finish Process}
PTS8: {Show Result(), Finish Process}

The number of rules in this initial set can be reduced using some heuristics. This is specially useful when we want to create user interfaces for small devices. In this example we can obtain the minimum set of rules shown in the figure 9. From these

PTS we can obtain the initial set of interactors (abstract interaction objects) associated with the basic tasks.



PTS1: {Show Topic, Show Aspects, Show Ideas, *Finish Process*}
PTS2: {Select Aspect-Idea(), Modify, Delete, Argue, Chose Aspect-Idea(), Choose "Create", Select Alternative Type, *Finish Process* }
PTS3: {Create(), Show Create Result(), *Finish Process*}
PTS4: {Propose Alternative(), Show Result(), *Finish Process*}

Figure 9: Lower number of tasks sets in the example

In figure 10 we can see the appearance of the concrete user interface for supporting the collaborative task *Modify Contents* (see figure 8). We centre our attention on the *Create Content* subtask. This subtask is modeled by the *Create* subtree (node *Create*). In figure 8 we can see that *Create* is composed for these sequential tasks: 1) selecting the type of object: aspect or idea (node *Select Aspect Idea*), 2) choosing the operation *Create* and writing contents of *Aspect* object or *Idea* object (node *Choose Create*) and 3) performing the operation and showing result (node *Execute and ShowCreate*). In figure 10 we can see these tasks marked as 1) Select Aspect-Idea, 2) Choose "Create" and 3) Execute and Show. In figure 10 we can also see the interactors used for each PTS obtained and the splitting of the user interface in personal and collaborative workspaces.

Figure 10: AULA User Interface

6 Discussion

Analyzing the models built we can observe that CIAM and CIAN allow the representation and modeling of the composition scenario created with AULA (see Figure 3). This model shows the tasks to do, the roles involved and the main and most significant objects handled and the way to do this. A precise definition of the roles and responsibilities for each task is also shown. Although not demonstrated as computable, this model is based on a conceptual framework described by means of ontologies, which facilitates its interpretation [Molina, 06].

The application of CIAM to the modeling of tasks supported by the AULA system has been demonstrated, dealing in depth with the modeling of a collaborative task. In particular, the Composition task presents a greater wealth of requirements relative to work-in-group/learning-in-group situations. The modeling of the collaboration starts with the previous description of tasks, roles and responsibilities and then describes how to share the context and how to access the objects of the shared context (see Figure 7). Thus, areas of individual visualization, of collaborative visualization and of exclusive edition are defined.

It is remarkable how CIAN systematically guides the extraction of the interaction model from the model of the shared context. This model of interaction is expressed in CTT (figure 8) and works as an entry point to use it with development tools of user interfaces based on models as is the case of TERESA [Berti, 2004] or the Dygimes framework [Luyten, 04]. This process is described in depth in [Molina, 2007]. Thus, a

user interface is obtained starting from an interaction model that gathers the requirements to support collaborative tasks.

In the specification of requirements for the AULA functionality there are other aspects related to the mobile computing paradigm. CIAM does not offer mechanisms involving these requirements. For example, there are objects data of the system created in different contexts (online and offline). These objects need tasks synchronization. CIAM does not model these aspects at the data level or at the task level. There are specific tasks for different devices. For example: tasks of argumentation on the interactive whiteboard (teacher and students), text editing tasks on PDAs (students) and tasks of planning of the composition on a PC (teacher). CIAN does not allow the user to associate tasks to devices nor can it describe and characterize tasks that are made with PDA devices in contexts of mobile computing. The use of these devices with this approach has had important implications: tasks that can be made with or without a connection to a network, the need to have synchronization tasks of information generated offline, physical proximity of the students, tasks that are done face-to-face in the classroom or which can be done at a distance, etc. In summary, we focus on the modeling of aspects related to contexts such as computation, user and physical, time and device contexts. The last one, in particular, can indeed be included modifying the model of interaction directly expressed in CTT. Nevertheless, this is not done in a guided and systematic way as the rest of the steps that CIAM proposes.

7 Conclusions

In this article we have presented how to apply a methodological approach (CIAM) for the development 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 an interaction model that can directly be used by a MBUID tool [Myers, 1995] to get implementations of end user interfaces. 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). Nevertheless, it does not use semantics to describe the features of mobile computing, specially required for the modeling of the context. This has arisen from its application in the design and development process of the user interface that supports one of the tasks of collaborative learning in the language learning system called AULA. This system is thought to support collaborative tasks with PDA mobile devices. 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 specially some parameters that allow modeling the context.

References

[Berti, 04] Berti, S., Mori, G., Paternò, F., Santoro, C. (2004). TERESA: A Transformation-Based Environment for Designing Multi-Device Interactive Applications. Demo Paper in

Proceedings of CHI 2004, CHI '04 extended abstracts on Human factors in Computing Systems, ACM Press, pp.793-794, 24-29 April 2004, Wien, Austria.

[Dillenbourg, 95] Dillenbourg, P., Baker, M., Blaye, A., O'Malley, C. (1995). The Evolution of Research on Collaborative Learning. London.

[Ellis, 91] Ellis, C., Gibbs, S. & Rein, G. (1991) Groupware: Some Issues and Experiences. Communications of the ACM. Vol. 34, no. 1, pp. 39-58, January 1991.

[García, 04] García, L., & Novella, M. A. (2004). Preparación de Materiales Didácticos Auditivos. Ponencias Seminarios de Diagnóstico Locales de la CECU. In http://www.congreso.unam.mx/ponsemloc/ponencias/873.html 22/12/04.

[Hayes, 80] Hayes, J.R., & Flower, L.S. (1980). Identifying the organization of writing process. In Gregg y Steinberg (Eds.), Cognitive Processes in Writing. LEA.

[Luyten, 04] Luyten, K. (2004). Dynamic User Interface Generation for Mobile and Embedded Systems with Model-Based User Interface Development, PhD. thesis, Universiteit Limburg.

[Molina, 06] Molina, A.I., Redondo, M.A., Ortega, M., (2006) A Conceptual and Methodological Framework for Modeling Interactive Groupware Applications. The 12th International Workshop on Groupware (CRIWG 2006). Y.A. Dimitriadis et al. (Eds.) Groupware: Design, Implementation, and Use: CRIWG 2006, LNCS 4154, pp. 413 – 420, 2006.

[Molina, 07] Molina, A.I., Redondo, M.A., Ortega, M. (2007) CIAM: A methodology for groupware user interface development. Journal of Universal Computer Science. Know-Center. Acepted for publication.

[Mori, 04] Mori, G., Paternò, F. et al. (2004) Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions, IEEE Transactions on Software Engineering, 507-520.

[Myers, 95] Myers, B. A. (1995). "User Interface Software Tools." ACM Transactions on Computer-Human Interaction 2 1: 64-103.

[Paredes, 03] Paredes, M., Sánchez-Villalón, P.P., Ortega, M., Redondo M.A., Bravo, C., & Bravo, J. (2003). Semi-automatic Assessment Process in a Ubiquitous environment for Language Learning. In J.M. Cueva, B.M. González, L. Joyanes, J.E. Labra y M.P. Paule (Eds), Web Engineering. LNCS 2722, 255-258. Springer Verlag.

[Paredes, 06] Paredes, M. (2006). AULA: un Sistema Ubicuo de Enseñanza-Aprendizaje Colaborativo. PhD. Departamento de Tecnologías y Sistemas de Información. Universidad de Castilla – La Mancha.

[Paternò, 04] Paternò, F. (2004). 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.

[Sharples, 93] Sharples, M., Goodlet, J.S., Beck, E.E., Wood, C.C., Easterbrook, S.M., & Plowman, L. (1993). Research Issues in the Study of Computer Supported collaborative writing. En M. Sharples (ed.), Computer Supported Collaborative Writing (pp. 9-28). Springer-Verlag.