A Model-Driven Approach for the Development of CSCL Tools that Considers Pedagogical Usability

Ana Isabel Molina
Universidad de Castilla-La Mancha
Escuela Superior de Informática
Departamento de Tecnologías y Sistemas de Información
(+34) 926 29 53 00
anaisabel.molina@uclm.es

Jesús Gallardo
Universidad de Zaragoza
E.U. Politécnica de Teruel
Departamento de Informática e Ingeniería de Sistemas
(+34) 978 645 387
jesus.gallardo@unizar.es

Christian Xavier Navarro
Universidad Autónoma de Baja California
Facultad de Ingeniería, Arquitectura y Diseño
cnavarro@uabc.edu.mx

Miguel Ángel Redondo
Universidad de Castilla-La Mancha
Escuela Superior de Informática
Departamento de Tecnologías y Sistemas de Información
(+34) 926 29 53 00
miguel.redondo@uclm.es

ABSTRACT
The application of the collaboration paradigm in software for teaching has resulted of a great help to increase motivation and participation of students. However, the development of such software is not an easy task. Model-driven development can be a help in this sense, provided that the peculiarities of collaborative learning systems are taken into account. In this paper, we introduce a model-driven development method for collaborative learning systems that gives support to group graphical modeling. The method is based on the use of models by different roles all over the development, and it also considers pedagogical usability factors to guarantee that the generated systems have into account the factors that are typical in the learning field. In order to have a measure of the usefulness of the method, we have applied it to create a series of collaborative modeling tools. These systems and the method have been evaluated by teachers/professors of different fields, who have stated a favorable opinion regarding the proposed approach.

CCS Concepts
· Software and its engineering → Model-driven software engineering

· Applied computing → Collaborative learning

· Human-centered computing → Usability testing

Keywords
Pedagogical usability, CSCL, Collaborative system design, Model-driven development.

1. INTRODUCTION
One of the objectives of the Computer Supported Cooperative Learning (CSCL) [13] is to exploit the advantages provided by groupware systems in the field of eLearning [5].

The application of CSCL paradigm favors the motivation and involvement of learners, and the exchange of ideas, knowledge and points of view, stimulating the creativity. On the other hand, its use allows developing skills such as decision-making in a group, argumentation, or the ability to communicate and transmit knowledge and opinions; all of them are necessary skills for the professional future of the learners.

Considering the spectrum of possible CSCL systems, in this work we will focus on systems that support activities of graphical modeling in groups. This type of systems is useful in disciplines where graphical notations or visual languages are frequently used. In the scope of Computer Science, numerous notations can be taught through such tools. For example, we could mention UML diagrams or network topologies. In other fields, there are also notations that can be learned using such systems. For example, digital circuits or concept maps fit perfectly into this approach.

However, the development of collaborative systems, in general, and the CSCL systems, in particular, is not a simple task [11]. There are aspects such as the support to model collaborative procedures, the roles supported by the system or the existence of spaces for sharing information, that they become in key design and implement requirements.

In recent years, our interest has focused on providing a methodological support, aligned with the principles of MDD (Model Driven Development), for the development of these applications [23].

As a consequence, the CIAM [18] methodology was developed. This methodology proposes a series of notations [22] and stages to design collaborative systems that meet usability requirements (i.e., a design based on models of users and tasks), as well as the principles of groupware usability [1, 29] (role modeling, incorporating coordination and communication tools, access control mechanisms to share context, etc.).

Nevertheless, CIAM only supports the phases of analysis and design of such systems, even though allows automatically generate the presentation layer of applications, i.e., the user interfaces. This process is supported by CIAT-GUI tool [21]. However, aspects such as the implementation of communication mechanism (specially, synchronous mechanism), the access control to shared resources, the support to coordination (by mean chats or decision support systems), the management of sessions or the inclusion of awareness elements (tele-pointers, user identification by color, etc.) are not technologically supported by the CIAM approach.

This lack could be solved by integrating the CIAM approach with the SpacEclipse framework [7], which allows semi-automatic generation of synchronous collaborative systems for modeling.

This framework is based in Graphical Modeling Framework (GMF) of the Eclipse platform. In a simple way, its use allows designers to generate a collaborative modeling tool adapted to any domain or type of diagram. This requires the definition of the elements or nodes that make up the models to create as well as the type of relationships or connections that may exist between them. Similarly, it is necessary the configuration of other collaboration features (to be included in the user interface), such as the type of communication and coordination mechanism, the awareness elements, etc. The result obtained is a collaborative synchronous modeling tool specialized for the domain of application chosen and conveniently specified and modeled. Currently, there exist tools like Eugenia [30] that make a similar job for using GMF in a simple way. However, our SpacEclipse proposal was developed when such tools were not still published and GMF had to be used with all its complexity. In addition, SpacEclipse is oriented towards a specific kind of modeling tools, which are graphical editing tools, so it can be more specific and not so generic.

Thus, by integrating CIAM (to support the analysis and design phases) with the SpacEclipse framework (to support the implementation phase), full support for the development of groupware modeling tools may be obtained. A first approach to integrate these two methods is described in [8].

However, if we expect to support the development of collaborative learning systems (CSCL) it is necessary to consider its educational and pedagogical dimension. The pedagogical aspects should be taken into account during the design of this type of interactive applications. Some authors address the treatment of these aspects by introducing the Pedagogical Usability term [10, 14, 25]. This concept would be related with all aspects of usability that positively influence the teaching/learning process.

Nevertheless, for the integration of the pedagogical usability it is necessary to characterize what are the criteria and/or dimensions that define it. For this objective, we take as reference the MoLEF proposal (Mobile Learning Evaluation Framework) [24]. MoLEF is a framework for evaluating the usability of mobile learning systems and deals with the technological usability (for mobile interfaces) and pedagogical usability (in a general point of view). In this work, we will use the part that allows validating the pedagogical usability.

Therefore, in this research we address a new evolution of the integration of CIAM with SpacEclipse in order to consider the criteria and guidelines for pedagogical usability that are proposed in MoLEF. Thus, the final goal is to propose a model-driven process for development of CSCL tools. This process will be task-oriented and will take into account aspects of pedagogical usability.

Finally, we present a case of study that shows how this approach has been applied to generate some collaborative modeling tools (for students of Computer Science degree) and a first validation of the method and of products that can be generated.

2. RELATED WORK

The development of CSCL systems is a complex task [35]. In this sense, several points of view for their development can be taken: *ad hoc* development, the use of patterns and/or components, or the use of model-driven approaches. In this work, we are using the model-driven paradigm to develop CSCL models. Therefore, we propose to apply the models proposed by CIAM in the first steps of the development, and to use the software components supported by SpacEclipse in further steps.

The principles of model-driven development have been applied, in the discipline of Computer-Human Interaction (CHI), mainly for the design and development of the user interface [32], giving birth to the area of *Model-Based User Interface Development* (MBUID) [15]. The modeling of group work has also been an area of interest in the field of CHI. There exist some proposals that have faced that challenge [20]. Among them, we propose the use of CIAM [18, 22], which faces the modeling of the interactive and group work factors. However, CIAM does not take into account the peculiarities of the design of collaborative learning systems.

Concerning the design and development of CSCL systems, there exist many proposals in that sense, but few of them adopt a model-driven approach [31, 34]. Most of them consider as a conceptual model of reference the one supported by the standard for instructional model IMS-LD [26], as well as a pattern and component based design [2] or scripts for teaching [12]. Although IMS-LD includes concepts that allow specifying group behavior, such as roles or notifications, it does not consider other factors to take into account when designing systems of collaborative learning [17], and even less if they are of a synchronous nature. What is more, basing the design of CSCL systems in scripts has its own limitations [4]. While many of those contributions follow a pedagogical approach, they use to be focused on specific teaching scenarios, such as the one of Problem-Based Learning (PBL) [16] or do not really generate fully functional CSCL systems [33].

Thus, the proposal we outline in this paper is distinguished by being a model-driven method that allows the automatic generation of a CSCL application in a way in which collaborative, interactive and pedagogical design factors are considered all over the development process. All these considerations have an effect in the final product, causing that it covers all those dimensions.

3. METHODOLOGICAL PROPOSAL FOR THE MODEL-DRIVEN DEVELOPMENT OF COLLABORATIVE LEARNING TOOLS

In this section, we describe how we have carried out the integration of the three initial proposals: the CIAM methodology, the technological support provided by SpacEclipse and lastly, the steps of evaluation and guidelines provided by the MoLEF framework.

3.1. A model-driven method for developing collaborative tools: CIAM+SpacEclipse integration

The first point that is going to be faced is to analyze the complementarity of both proposals, CIAM and SpacEclipse. Both of proposals are aligned with the model-driven paradigm. In Table 1, factors that are supported by CIAM and SpacEclipse are shown so that it gets clear the support they share and the one they lack.

As it can be seen in the comparison, both proposals are complemented in most factors. Thus, CIAM supports the phase of requirements specification of the groupware system by means of graphical notations, but it only supports the automatic generation of the presentation layer of the application.

Regarding technological issues, the integration is possible as both proposals include the same model-driven approach and share most technologies. This makes easy to integrate the meta-models of both proposals, which have many concepts in common.

The main problem for this integration is that SpacEclipse is oriented towards a very specific kind of applications, which are the ones for developing diagrams and models, whilst CIAM aims to give support to any kind of group work task. Therefore, our integration proposal will only give support to the creation of synchronous modeling CSCL applications in its first version.

Table 1. Analysis of the complementarity of the CIAM and SpacEclipse proposals

	CIAM	SpacEclipse
Kind of systems it supports	Collaborative systems	Collaborative modeling systems
Specification techniques	CIAN notation, CTT	EMF[1]
Support for requirements analysis and techniques for requirements specification	Yes	No
Support for the definition of the application domain	Partial	Yes
Support for task definition	Yes	No
Support for the modeling of interactive issues	Yes	Only for collaborative modeling tasks
Support for automatic code generation	Only for the user interface; supported by the CIAT-GUI tool [21]	Yes
Incorporation of elements for awareness support in the generated GUI	No	Yes
Incorporation of elements for communication and coordination	No	Yes (communication tools, decision making tools, etc.)
Support for the definition and configuration of workspaces and sessions	No	Yes

3.2. MoLEF: Framework for pedagogical usability

As we mentioned in the introduction, although the integration of CIAM and SpacEclipse allows generating fully functional collaborative modeling tools, there is no way to guarantee that such systems cover pedagogical issues when generating CSCL systems. In order to cover that, we have used the MoLEF framework [24]. MoLEF allows us to give support to the evaluation of *m-learning* applications. In order to achieve that, it defines a series of dimensions and sub-dimension for the evaluation of the usability factors that an application of this kind should support. Therefore, MoLEF can be used for both evaluating and designing *m-learning* systems.

Between the higher level dimensions, the framework differentiates between the design and evaluation of *technological usability* (the one of the user interface, centered on mobile computation factors) and *pedagogical usability* (the one which includes criteria related with learning factors). In our integration proposal, we will just consider the latter, as it can be applied to any learning application, not only to *m-learning* applications.

The MoLEF framework defines the *pedagogical usability* dimension by dividing it in five sub-dimensions: *content*, *multimedia*, *tasks or activities*, *social interaction* and *personalization*. As it can be seen, the social dimension of the learning process, which is also faced by CIAM, is considered by MoLEF. Each sub-dimension includes some elements to consider when designing and evaluating a learning application.

[1] Eclipse Modeling Framework: www.eclipse.org/modeling/emf/

As an example, we show in Table 2 the definition of the elements related to *tasks or activities* and *social interaction*.

Table 2. Elements for evaluation in two sub-dimensions of the MoLEF framework

Tasks or activities	
Aligning with objectives	Tasks or activities must have a strong connection to the objectives.
Sequencing	Tasks must allow students to integrate new information with prior learning to generate knowledge.
Problem-based learning	Tasks should require students to compare and classify information, make deductions, and promote creativity.
Authenticity	The task should reflect real-world practice, relevant to professional practice, generating interest and engagement in students. They must support transference of skillsbeyond the learning environment and critical thinking.
Interactivity	Tasks should engage students in problems to solve, that take advantage of state of the art mobile design (field investigations, taking pictures, videos, augmented reality, QR codes).
Adequacy	Tasks should be congruent with the content and capabilities of the target audience.
Self-evaluation	Software should allow opportunities wherever appropriate for self-assessment that advance students' achievement.
Social interaction	
Dialogue	The m-learning application should allow students to communicate with their classmates or teachers (chat, notice board or social networks).
Collaboration	The mobile learning environment should allow students to do group work with their classmates.
Discussion	The mobile learning environment should provide opportunities to support learning through interaction, discussion and other collaborative activities.
Sharing	The m-learning application should allow students to share photos, videos or any other documents related to their work.

In Figure 1, we detail all sub-dimensions and elements that describe *pedagogical usability* in MoLEF.

One of the main contributions of MoLEF, together with defining specifically each element as a guideline for the design of learning systems, is that it includes a mechanism for their evaluation: the CECAM questionnaire (*Cuestionario de Evaluación de la Calidad de Aplicaciones M-learning: questionnaire for the evaluation of the quality of m-learning applications*), which is made up of 56 items. The questions that it includes can be used as heuristics for the design of *m-learning* systems or as an evaluation checklist. 29 out of the 56 items in the questionnaire refer to *pedagogical usability*. CECAM has undergone a refinement process in which its validity and its reliability have been analyzed. Thus, it can be considered a quality and reliable mechanism. Moreover, a software tool (Figure 2) has been developed for its application and the subsequent analysis of the results obtained.

In Figure 2 it can be seen a screenshot of the analysis module of the tool that supports the application of the CECAL questionnaire. In this module, the results of the application of the questionnaire can be analyzed in three ways: with the numeric values, with a bar graph and with a star graph.

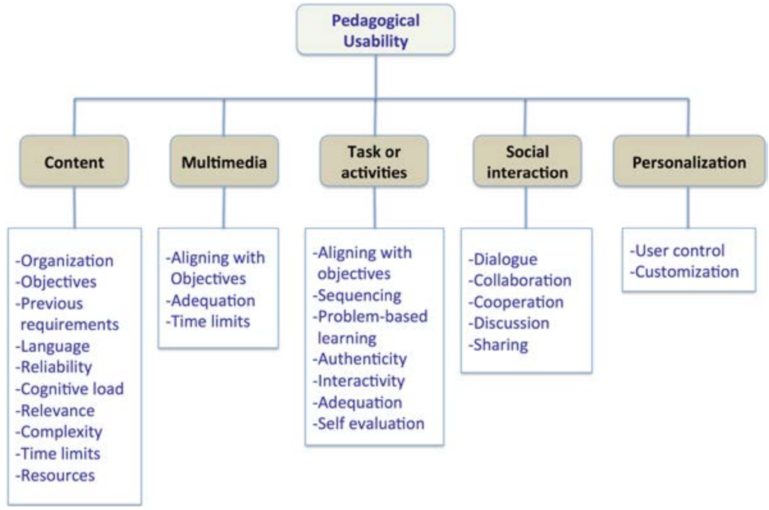
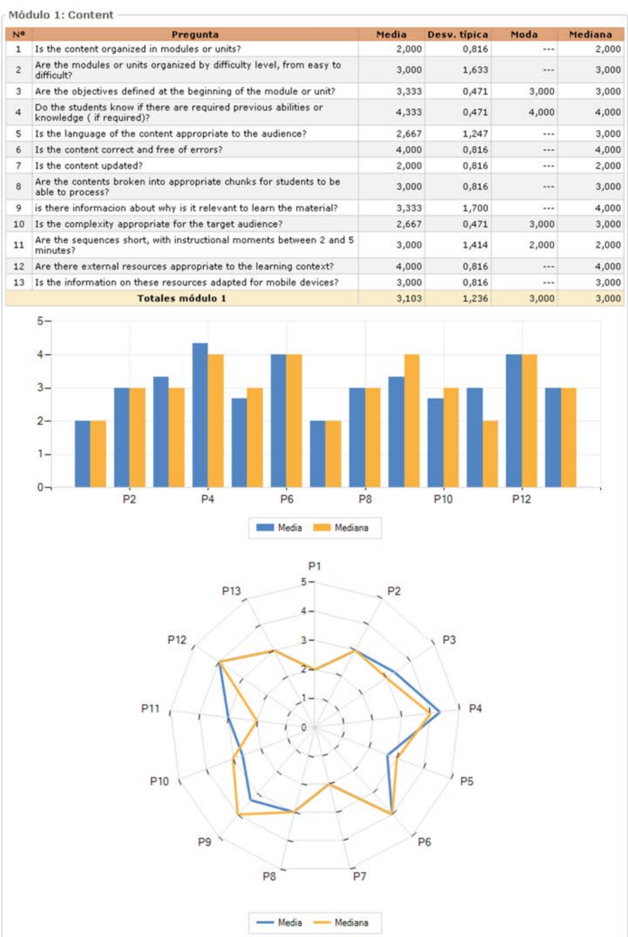


Figure 1. Elements of *pedagogical usability* in the MoLEF framework.



Módulo 1: Content

Nº	Pregunta	Media	Desv. típica	Moda	Mediana
1	Is the content organized in modules or units?	2,000	0,816	---	2,000
2	Are the modules or units organized by difficulty level, from easy to difficult?	3,000	1,633	---	3,000
3	Are the objectives defined at the beginning of the module or unit?	3,333	0,471	3,000	3,000
4	Do the students know if there are required previous abilities or knowledge (if required)?	4,333	0,471	4,000	4,000
5	Is the language of the content appropriate to the audience?	2,667	1,247	---	3,000
6	Is the content correct and free of errors?	4,000	0,816	---	4,000
7	Is the content updated?	2,000	0,816	---	2,000
8	Are the contents broken into appropriate chunks for students to be able to process?	3,000	0,816	---	3,000
9	is there informacion about why is it relevant to learn the material?	3,333	1,700	---	4,000
10	Is the complexity appropriate for the target audience?	2,667	0,471	3,000	3,000
11	Are the sequences short, with instructional moments between 2 and 5 minutes?	3,000	1,414	2,000	2,000
12	Are there external resources appropriate to the learning context?	4,000	0,816	---	4,000
13	Is the information on these resources adapted for mobile devices?	3,000	0,816	---	3,000
	Totales módulo 1	3,103	1,236	3,000	3,000

Figure 2. Screenshot of the tool that allow applying the CECAL questionnaire.

3.3. A new model-driven method for the development of CSCL tools

The methodological framework of the method is made up of a series of phases to be followed when it is applied, the roles that users may play in those phases and the models used. Several users can take part in the development method at the same time, so they will handle those models depending on the role they play. Thus, users taking part in the development method may play any of these roles:

- The *teacher* is the person who states the need to have a collaborative CSCL tool available. He has lots of experience in the domain over which the tool that he is going to teach will work.
- The *software engineer* may participate in all phases in which software development tools are manipulated. He or she will have knowledge of the development method, its basis and its notations.
- The *student* will use the collaborative CSCL tools. Such tools are generated in the scope of the teaching/learning process of a certain subject. Usually, students get organized in groups are work in class sessions.

In Figure 3, the global diagram of integration of proposals that has been mentioned is shown. Next, each one of the phases that integrate the method is explained in detail.

Phase 1: **Organization Specification.** This phase is carried out by the *teacher*, and it is composed by two sub-phases:

- **Sociogram Development**. This phase comes from the CIAM methodological proposal. In it, the *sociogram* is generated in order to depict the structure of the organization to which the collaborative system will give support, as well as the relationship among its members. Thus, *actors*, *roles*, *groups*, *work teams* and *software agents* will be defined. Elements in these diagrams might be interconnected by means of three kinds of relationships: (a) *inheritance* relationship (for specifying responsibilities inheritance between roles); (b) *acting* relationship (between actors and roles) and (c) *association* relationship, for specifying situations in which some roles collaborate to carry out a joint task. In the case of CSCL applications, the main roles will be the ones of *teacher* and *student*. They may be specialized in sub-roles depending on their contribution to the group work.

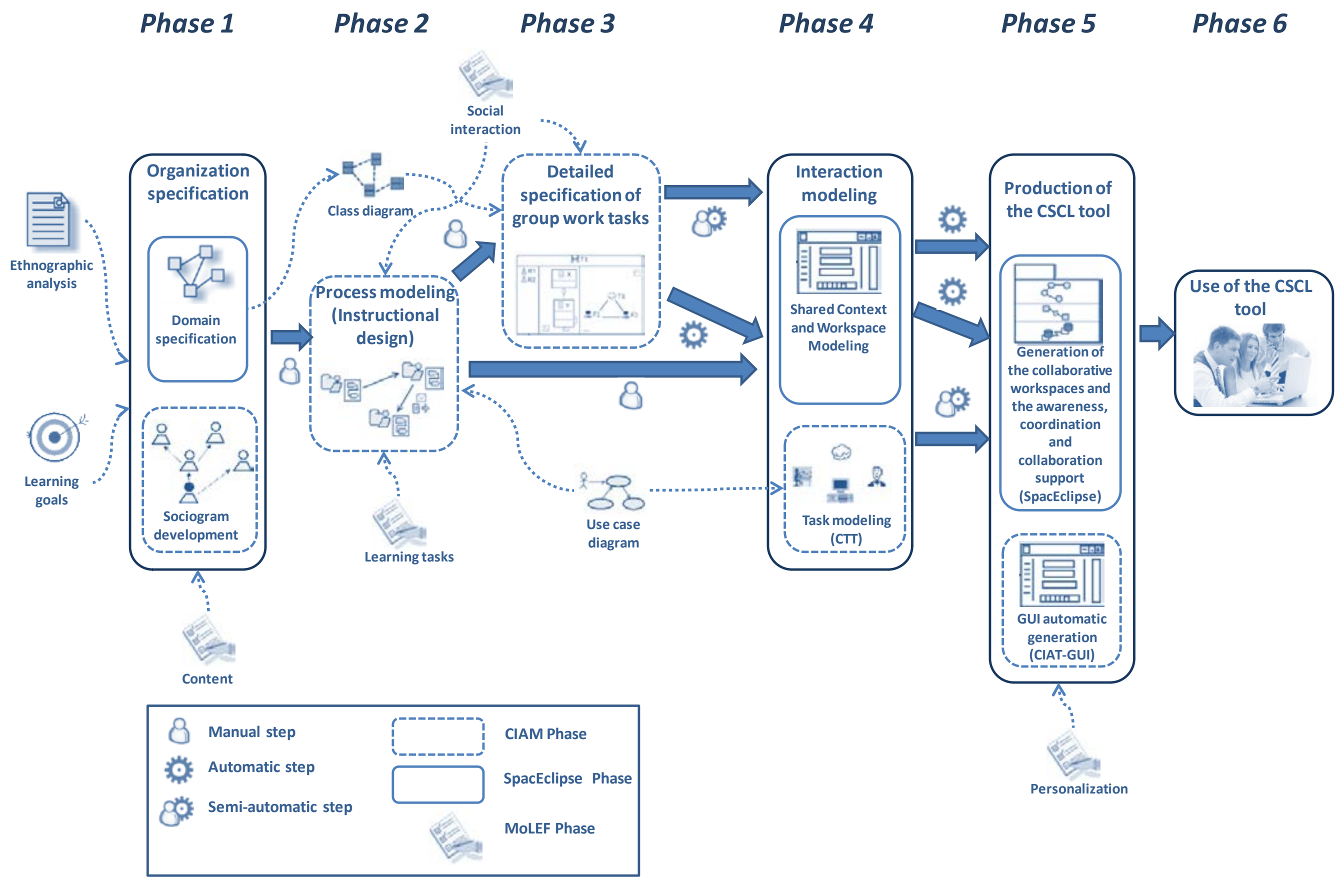


Figure 3. Methodological framework for the integration of the CIAM, SpacEclipse and MoLEF proposals.

- **Domain Specification.** In this step, the main elements and concepts of the application domain are identified and defined, just as it is shown in the SpacEclipse method. The teaching task will consist on the students designing and creating a model or diagram in a collaborative way.

In this phase, when the responsibilities of the roles are defined and the initial specification of the domain of the teaching task is specified, is when the first checklist of pedagogical factors is applied: the one related to the content. Two models are generated in this step: the sociogram and the domain model, which usually will consist of UML class diagrams. Users playing the *teacher* and *software engineer* roles may have access to those models in order to generate and modify them.

Phase 2: **Process Modeling (Instructional Design)**. In this stage, the main tasks defining the group work developed in the organization are described. A *collaborative process* consists of a set of tasks carried out in a certain order, taking into consideration certain data or temporal restrictions among them. For each *task*, the *roles* involved, the *data manipulated* and the *product* obtained in the task, are specified. For the data specified in the context of a task, the access modifiers to the objects are defined, which can be *reading*, *writing* or *creation*. Each task must be classified in one of the following categories: *group work task* or *individual task.*

The tasks in the process will be interconnected by means of several kinds of relationships: *temporal dependencies* (order relationship), *data dependencies* (when tasks need data manipulated by previous tasks) and *notification dependencies* (when it is necessary for a certain event to occur so that the workflow continues). The model including all this information will be the deliverable of this phase.

In this phase of specification of the instructional design, or sequencing of the learning activities, the section of the CECAM questionnaire that would be applied would be the one that allows validating the pedagogical factors of the activities, which is called *learning activities*.

Phase 3: **Detailed Specification of Group Work Tasks**. In this stage, the main collaborative tasks identified in the previous stage are described in detail, as it is originally done in the CIAM methodology. We classify the collaborative tasks in two main types: (a) *Tasks for supporting communication and coordination factors* (decision-making tasks, work distribution tasks, asynchronous and synchronous communication support tasks, etc); and (b) *Tasks for supporting collaborative creation of shared artifacts* (which can be of a different nature: textual, graphical, etc.). In this proposal we support the obtainment of tools for supporting tasks of the first type (supported by SpacEclipse), and, among the tasks in the second type, we support tasks for supporting collaborative visualization of shared information (supported by CIAM method) and tasks for supporting collaborative visualization and editing of graphical information (supported by the SpacEclipse method).

For specifying tasks for supporting collaborative access to shared information, we use the models provided by CIAM, in particular, those for *collaborative task modeling*. Collaborative task modeling requires the specification of the *roles* involved in its execution, as well as the objects of the data model manipulated and shared by the work team, that is, the specification of the *shared context* [5]. The shared context is defined as the set of *objects* that is visible to users, as well as the *actions* that can be executed on those objects.

Both in the previous phase (*process modeling*) as in the current one, the checklist in MoLEF about the support to *social interaction* factors would be applied.

Phase 4: **Interaction Modeling**. In this phase, the *teacher* and the *software engineer* specify the *human-computer interaction* issues, that is, the models related to the most external part of the collaborative application: its *graphical user interface*. For modeling the interaction issues, we propose several specification techniques, joining what was proposed in the methods being integrated:

- **Task Modeling (CTT)**. For specifying interaction issues of individual and collaborative tasks, we propose the use of *task models*. The *task models* are logical descriptions of the tasks that users must carry out in order to achieve their objectives while interacting with the application [27]. We propose the use of the *ConcurTaskTrees* (CTT) notation [28] for task modeling. In the case of the *individual tasks*, the CTT model has to be built, but, in the case of the *collaborative tasks*, the CTT interaction model can be extracted directly from the *shared context* definition. The algorithm used to extract the CTT model is described in [19].
- **Shared Context and Workspace Modeling**. Once processes, roles and tasks are identified and modeled, in the case of collaborative tasks in which there is a *shared context*, a more detailed specification of these shared artifacts is required. In addition, the *workspace* issues and the *awareness* and *collaboration support elements* to be included in the final tool must be specified. A set of widgets and support tools that are supported by the technological framework may be included in the workspace, such as a chat, a session panel, a floor control tool, a radar view, tele-pointers, etc. The *teacher* and the *software engineer* are responsible for modeling these factors. They must build the set of meta-models needed for the subsequent automatic generation of the final tool.

Phase 5: **Production of the CSCL tool**. Once the shared context and the workspaces definition have been formalized, some automatic steps take place resulting in the generation of the final collaborative tool. In this phase, the graphical user interface (GUI) and the collaborative tools are generated applying a set of M2M and M2T transformation processes. This phase consists of the following two sub-phases:

- **GUI Automatic Generation (CIAT-GUI).** In the case of individual tasks and collaborative tasks without shared context, the GUI is semi-automatically obtained by applying the method described in [21]. This is a model-based user interface development (MBUID) method that allows final GUIs to be obtained from declarative models (a *task model* in CTT notation and a *domain model* in UML notation). This method is supported by a tool called CIAT-GUI implemented using MDE technologies such as EMF, GMF, ATL and MofScript.
- **Workspace Generation in Collaborative Tasks with Shared Context (SpacEclipse).** In the case of collaborative tasks with shared context, a set of M2M transformations are applied. The models required by GMF in order to generate a graphical editor are generated from the *shared context* specification. These transformations were developed by us using the ATL language [6]. Next, a set of M2T transformations allows the final tools for supporting visualization of shared context to be generated and, in the case of graphical shared artifacts, for collaborative edition of models to be supported. The specific M2T transformations being carried out are an extension of the original GMF transformations that make up the final tool [6].

Once the final user interface of the application has been obtained, the factors related to *personalization* supported by the application can be evaluated.

Phase 6**: Use of the CSCL System**. Once the tool has been generated, *students* can work collaboratively using the CSCL system.

As it has been previously stated, the integration with SpacEclipse implies that, up to this moment, the kind of activity that is supported by the framework is a synchronous collaborative modeling one. In future works our aim is to include some other kinds of collaborative tasks, as group edition of text information or similar ones. The fact that the domain is restricted to this kind of tasks also makes that some dimensions in the MoLEF framework cannot be applied up to this moment, as is the case of the *multimedia* dimension.

4. A FIRST VALIDATION OF THE APPROACH

In order to test the usefulness and versatility of the method described, we have applied it to several domains. In particular, it has been applied to domains related with subjects in the Degree of Computer Science and Engineering. During their studies, students face several subjects in which they need to create specification of a graphical nature, that is, diagrams or models. This is the case of subjects such as Software Engineering, Network Design or Computer-Human Interaction, among other ones. Many works have to be carried out arranged in groups, but the existing tools do not support synchronous collaborative modeling. Thus, the use of such applications can be of great interest in this field. Moreover, students will also work in groups in their professional future, as they may carry out group tasks or take part in projects arranged in work groups. Therefore, we find useful that students of Computer Science and Engineering acquire competences such as negotiation, coordination and group diagram creation. These factors are related with the *Authenticity* element in the MoLEF framework.

Thus, we have applied the method to create several instances in various domains. In particular, we have instantiated it for the collaborative edition of UML diagrams, CTT diagrams and network design (Figure 4).

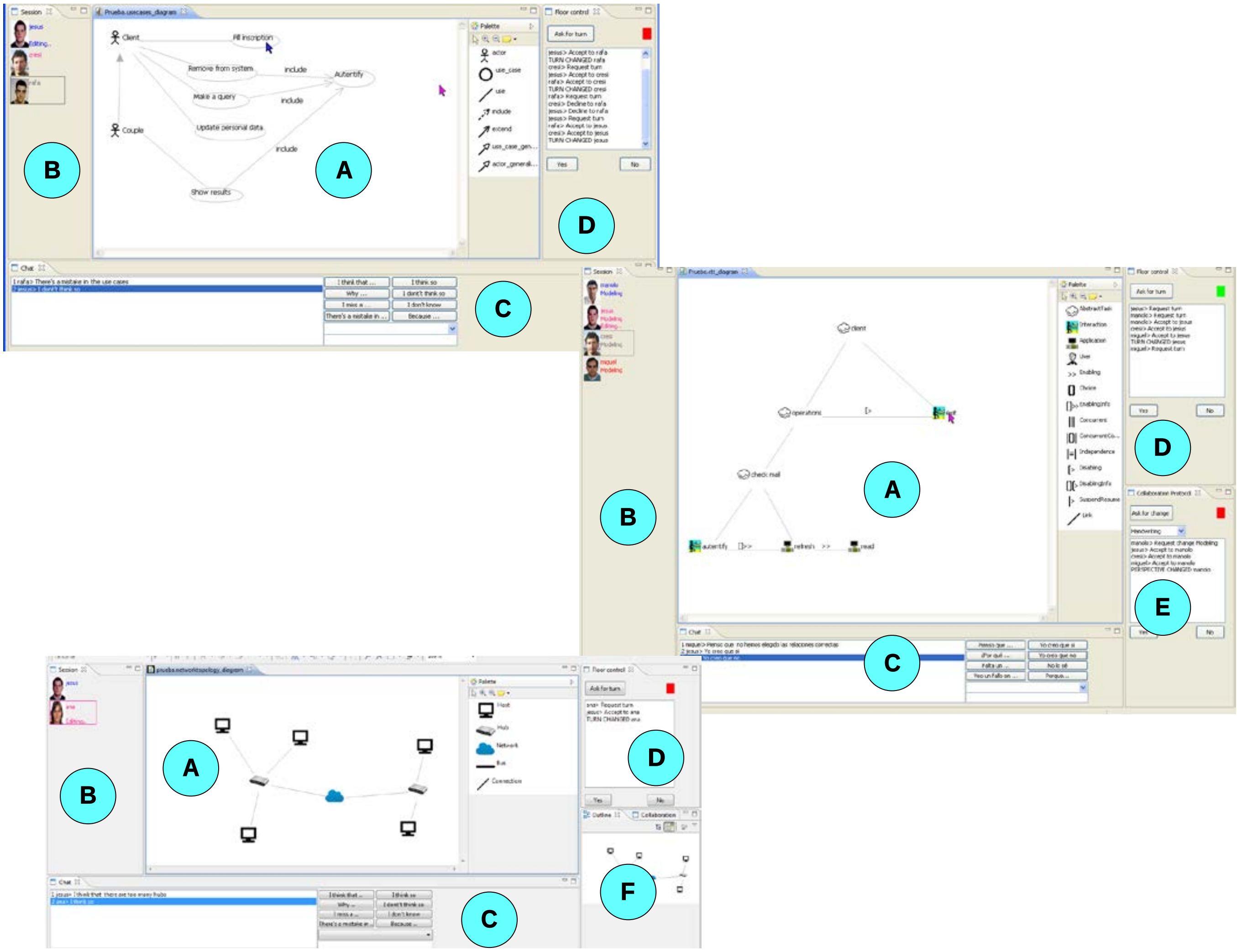


Figure 4. Instances of tools generated by applying the method for several domains

In Figure 4, we can also see how the different widgets that make up the user interface of each tool. In the figure, we have identified the following widgets: (A) graphical editor, (B) session panel, (C) chat, (D), turn taking tool, (E) collaboration protocol tool, and (F) outline. It can be seen how not all tools own the same widgets, as such configuration is specified in one of the models that are defined in Phase 4.

Once the instances of the tool were created, we carried out a first evaluation with professors of several subjects in which it is usual to carry out modeling tasks. The goal of this first evaluation was to catch their opinion about the method proposed and the tools generated by means of it. Therefore, we carried out a study with some professors about the global approach, as well as about the use of this kind of visual modeling collaborative tools in their teaching tasks. A full description of the study can be found in [9].

In the evaluation, 10 professors took part. They belonged to the *Escuela Superior de Informática* in Ciudad Real (University of Castilla-La Mancha) and the *Escuela Universitaria Politécnica* in Teruel (University of Zaragoza). Professors received a brief seminar about the tools and the development approach and then they answered a questionnaire about the tools generated by the development method. Professors had to indicate in a Likert scale from one to five their degree of agreement with a series of statements.

In the second part of the questionnaire, we included some statements about the *perceived ease of use*, *perceived usefulness* and *intention to use* of the participants in relation to the generated tools. The questions were adapted from the *Technology Acceptance Model* (TAM) [3], which is often used to know the subjective opinion of user about an artifact or technology. We also included a question that allowed evaluating the communication mechanisms provided by the tool. Lastly, two open-answer questions allowed collecting some other comments related to the tool and the development approach.

Most participants considered the generated tools useful for their subjects, with an average value of 4,7 (σ = 0,5) in the evaluation scale. This was the same score obtained by the global approach (μ = 4,7; σ = 0,5), although some professors considered that the number of domain in which the approach and the tool can be used is not so high (μ= 4,1; σ = 0,7).

Concerning the answers given about *subjective perception* (TAM evaluation framework), the tool was considered as very easy to use (μ = 4,6; σ = 0,5) and learn (μ = 4,7; σ = 0,5). Although the professors considered the tool useful to improve the competence of group work of the students (μ = 4,5; σ = 0,7), they did not consider it so useful for their teaching task (μ = 3,8; σ = 0,8). Even so, they showed a receptive attitude over its use, as most of them would use it in his classes (μ = 3,9; σ = 0,3) of would recommend it to other professors for it use (μ = 4,2; σ = 0,6).

Regarding the last questions, most participants valued the approach, mainly its versatility and its adaptation to several domains. They also made some remarks towards the improvement of the communication and coordination mechanisms that are supported, and they proposed the possibility of simulation or code generation from the models developed. This would imply a lot of work, as that functionality is very close to the application domain. Thus, we have not included that line as a priority one in our future work. Lastly, we found some comment about the higher usefulness that the tools may have in the first steps of the teaching of the specific modeling domains.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a model-driven approach for the development of graphical modeling CSCL tools. The method has been developed from the authors' previous experience in model-driven development methods for groupware. The main novelty of this work is the integration that has been carried out in order to obtain a method that considers technological factors as well as those about pedagogical usability. This way, the method is applied to CSCL tools, not to any CSCW tool. The development method implies several users playing different roles working over different models during the application models of the method. Models used include Ecore models that are used in the scope of the Eclipse Modeling Framework and other conceptual models such as CTT models for task modeling or UML class diagrams for domain modeling.

Pedagogical and teaching factors are covered in the method by considering the evaluation guidelines proposed by a pedagogical usability framework (MoLEF) during the different phases of the method. The global approach that the method proposes has been evaluated by means of some questionnaires fulfilled by university professors. The evaluation generated some good results.

As a first line of future work, we intend to go deeper in the development of the method, providing it with a full technological support that makes up its technological framework. Therefore, our goal is to get the CSCL tools implemented with as less programming effort as possible. In addition, pedagogical issues will be integrated in the final tools in a better way when this technological support is finished.

In the same way, we aim to continue with the evaluation of the approach by carrying out studies that are more exhaustive and putting it into practice in real teaching environments. In fact, the validation that we have described in this work is a very preliminary one. Further evaluations will include comparisons between tools developed using the method and without using it, and comparisons between tools generated using different characteristics of the method.

Another line of future work, in order to go deeper in the CSCL integration, is to study a possible integration with the Tin Can API, also known as eXperience API[2]. In addition, we will work over the conceptual framework of the approach in order to consider improvements such as the addition of *families* of domains, which could suppose a kind of inheritance of elements, properties and relationships among the domains.

[2] https://tincanapi.com/

6. ACKNOWLEDGEMENTS

This work has been partially supported by the PPII-2014-021-P project, funded by the *Junta de Comunidades de Castilla-La Mancha* (Spain) and by the TIN2015-66731-C2-2-R project, funded by the *Ministerio de Economía y Competitividad* (Spain).

7. REFERENCES

[1] Baker, K., Greenberg, S. and Gutwin, C. 2001. Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. In: *Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'01)*, 123-140.

[2] Calvo, A. and Turani, A. 2010. E-Learning Frameworks = (Design Patterns + Software Components). In: *E-learning, design patterns and pattern languages*. Sense Publishers. 49-64.

[3] Davis, F.D. 1993. User acceptance of information technology: System characteristics, user perceptions and behavioural impacts. *International Journal of Man–Machine Studies* 38, 3, 475–487.

[4] Dillenbourg, P. 2002. Over-scripting CSCL: the risks of blending collaborative learning with instructional design. In: *Three worlds of CSCL. Can we support CSCL?* Open Universiteit Nederland, Heerlen, 61–91.

[5] Ellis, C., Gibbs, S. and Rein, G. 1991. Groupware: Some Issues and Experiences. *Communications of ACM* 34, 1, 39-58.

[6] Gallardo, J., Bravo, C. and Redondo, M.A. 2011. A domain definition and transformation approach for the generation of GMF-based modeling tools. 3rd International Workshop on Model-Based Software and Data Integration (MBSDI 2011). June 6th, 2011. Birmingham, UK

[7] Gallardo, J., Bravo, C. and Redondo, M.A. 2012. A model-driven development method for collaborative modeling tools. *Journal of Network and Computer Applications* 35,1086-1105.

[8] Gallardo, J., Molina, A.I,, Bravo, C. and Redondo, M.A. 2013. A model-driven and task-oriented method for the development of collaborative systems. *J. Network and Computer Applications* 36, 6, 1551-1565.

[9] Gallardo, J. and Molina, A.I. 2014. Experiencia de uso de una herramienta de modelado colaborativo síncrono multi-dominio en el Grado en Ingeniería Informática. In: *Actas de las XX Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI 2014)*. 67-74.

[10] Gebera, O.W.T. 2012. Criterios de valoración sobre la usabilidad pedagógica en la formación continua docente. *Razón y palabra* 80, 48-22.

[11] Grudin, J. 1992. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. En *Groupware: Software for Computer Supported Cooperative Work*. IEEE Press: Los Alamitos, CA. 552--560.

[12] Hernández, D., Asensio, J.I., Dimitriadis, Y. and Villasclaras, E.D. 2010. Pattern languages for generating CSCL scripts: from a conceptual model to the design of a real situation. In: *E-learning, design patterns and pattern languages*. Sense Publishers. 49– 64.

[13] Koschmann, T. 1996. *CSCL: Theory and Practice of an emerging paradigm*. Lawrence Erlbaum Associates, Mahwah, NJ.

[14] Mayes, J. T. and Fowler, C. J. 1999. Learning technology and usability: a framework for understanding courseware. *Interacting with computers* 11, 5, 485-497.

[15] Meixner, G. Paternò, F. and Vanderdonckt, J. 2011. Past, present and future of model-based user interface development. *i-com* 3/2011.

[16] Miao, Y., Ally, M., Samaka, M. and Tsinakos, A.A. 2014. Towards pedagogy-driven learning design: a case study of problem-based learning design. In: *Advances in Web-Based Learning–ICWL 2014*. Switzerland: Springer International Publishing. 179–189.

[17] Molina, A.I., Jurado, F., Giraldo, W.J., Redondo, M.A. and Ortega, M. 2008. Specifying Scripts and Collaborative Tasks in CSCL Environment Using IMS-LD and CIAN. En *ICALT 2008*. 775-777

[18] Molina, A.I., Redondo, M.A., Ortega, M. and Hoppe, U. 2008. CIAM: A Methodology for the Development of Groupware User Interfaces. *J. UCS* 14, 9, 1435-1446.

[19] Molina, A.I., Redondo, M.A. and Ortega, M. 2009. A methodological approach for user interface development of collaborative applications: A case study. *Sci. Comput. Program.* 74, 9, 754-776.

[20] Molina, A.I., Redondo, M.A. and Ortega, M. 2009. A Review of Notations for Conceptual Modeling of Groupware Systems. In: *New Trends on Human-Computer Interaction: Research, Development, New Tools and Methods.* Springer-Verlag.

[21] Molina, A.I., Giraldo, W.J., Gallardo, J., Redondo, M.A., Ortega, M. and García, G. 2012. CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems. *Advances in Engineering Software* 52, 10-29.

[22] Molina, A.I., Gallardo, J., Redondo, M.A., Ortega, M. and Giraldo, W.J. 2013. Metamodel-driven definition of a visual modeling language for specifying interactive groupware applications: An empirical study. *Journal of Systems and Software* 86, 7, 1772-1789.

[23] Molina, A.I., Giraldo, W.J., Ortega, M., Redondo, M.A. and Collazos, C.A. 2014. Model-driven development of interactive groupware systems: Integration into the software development process. *Sci. Comput. Program.* 89, 320-349.

[24] Navarro, C.X., Molina, A.I., Redondo, M.A., Juárez-Ramírez, R. 2016. Framework to Evaluate M-Learning Systems: A Technological and Pedagogical Approach. *IEEE-RITA* 11, 1, 33-40.

[25] Nokelainen, P. 2006. An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. *Educational Technology & Society* 9, 2, 178-197.

[26] Paquette, G., Léonard, M., Ludgren-Cayrol, K., Mihaila, S., and Gareau, D. 2006. Learning design based on graphical modelling. *Educational Technology & Society* 9, 1, 97–112

[27] Paternò, F. 1999. *Model-based design and evaluation on interactive applications*. London: Springer Verlag.

[28] Paternò, F. 2004. ConcurTaskTrees: an engineered notation for task models. In: Diaper D, Stanton NA, editors. *The handbook of task analysis for HCI*. Mahwah, NJ: LEA (Lawrence Erlbaum Associates); pp. 483–501.

[29] Pinelle, D. and Gutwin, C. 2002. Groupware walkthrough: Adding context to groupware usability evaluation. In: *Proceedings of the 2002 SIGCHI Conference on Human Factors in Computing Systems*. ACM Press. 455–462.

[30] Rose, L.M., Kolovos, D.S., and Paige, R.F. 2012. EuGENia live: a flexible graphical modelling tool. In: *Proceedings of the 2012 Extreme Modeling Workshop (XM '12)*, Davide Di Ruscio, Alfonso Pierantonio, and Juan de Lara (Eds.). ACM, New York, NY, USA, 15-20.

[31] Sallaberry, C., Nodenot, T., Laforcade, T. and Marquesuzaa, C. 2005. Model driven development of cooperative Problem-Based Learning Situations. Implementing tools for teachers and learners from pedagogical models. In: *Proceedings of the 38th Hawaii International Conference on System Sciences*.

[32] Vanderdonckt, J. 2008. Model-Driven Engineering of User Interfaces: Promises, Successes, and Failures. In: *Proc. of 5th Annual Romanian Conf. on Human-Computer Interaction ROCHI'2008* ,Matrix ROM, Bucarest. 1-10.

[33] Wang, D., Miao, Y., Hoppe, U., and Samaka, M. 2014. A domain-specific modeling language approach to support various forms of online PBL. In: *Proc. of the 14th IEEE International Conference on Advanced Learning Technologies - ICALT2014*.

[34] Wang, D., Samaka, M., Miao, Y., Ali, Z. and Hoppe, U. 2016. A model-driven PBL application to support the authoring, delivery, and execution of PBL processes. *Research and Practice in Technology Enhanced Learning* 11,6.

[35] Wang, Q. 2009. Design and evaluation of a collaborative learning environment. *Computers & Education* 53, 1138–1146.