Journal of Universal Computer Science, vol. 14, no. 9 (2008), 1463-1479
submitted: 30/4/07, accepted: 19/2/08, appeared: 1/5/08 © J.UCS

Defining Tasks, Domains and Conversational Acts in CSCW Systems: the SPACE-DESIGN Case Study

Rafael Duque
(School of Computer Science and Engineering
Department of Information Technologies and Systems
University of Castilla – La Mancha, Ciudad Real, Spain
Rafael.Duque@uclm.es)

Jesús Gallardo
(School of Computer Science and Engineering
Department of Information Technologies and Systems
University of Castilla – La Mancha, Ciudad Real, Spain
Jesus.Gallardo@uclm.es)

Crescencio Bravo
(School of Computer Science and Engineering
Department of Information Technologies and Systems
University of Castilla – La Mancha, Ciudad Real, Spain
Crescencio.Bravo@uclm.es)

António José Mendes
(Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática
Universidade de Coimbra, Coimbra, Portugal
toze@dei.uc.pt)

Abstract: Most of the current academic and professional work requires collaboration between the members of a working group. Groupware tools play a prevailing role in supporting this collaborative work, often from different locations and at the same time. The research field of CSCW (Computer-Supported Cooperative Work) studies how to design effective groupware tools. To increase their potential, groupware systems must be flexible and have the capacity to adapt themselves to multiple tasks and situations. In order to provide answers to these challenges, in this article we propose the use of meta-models and XML-based languages to specify the most important characteristics of a groupware modeling system, such as the application domain, the requirements of the tasks to be carried out, how communication takes place and the regulation of the shared workspace. These models and techniques have been used to develop a specific groupware system called SPACE-DESIGN (SPecification and Automatic Construction of collaborative Environments of DESIGN), a CSCW tool with support for synchronous distributed collaborative work that adapts and re-configures itself as a result of processing the domain specification, the task, the communication and the system working norms.

Keywords: CSCW, groupware**,** model-driven development
Categories: D.2.2, H.4.3, H.5.3

1 Introduction

The effectiveness of group work is an essential requirement in order to successfully complete most of the current professional and academic activities. Over the last ten years, many proposals have been made and many groupware systems have been developed to support collaborative work [Poltrock 98]. Some aspects such as awareness, synchronization and shared workspaces are new concepts that groupware introduces. These new elements and the inherent complexity of the design and evaluation of collaborative applications, due to the fact that social protocols and group activities must be taken into account [Grudin, 93], make groupware development a difficult task.

In this article, we explore a number of mechanisms that can be used to characterize and specify the most important aspects of a groupware system, such as the application domain, the tasks to be carried out and the support for communication and coordination. Our final aim is to reduce the complexity and difficulty of developing groupware. Our approach consists firstly in specifying some components of such systems and then generating the system in a model-driven process [Karsai 03]. The system will be able to deal with diverse scopes of design defined by means of a re-configuration process. Some researchers have tried to follow a similar approach (e.g the AMENITIES methodology [Garrido 07]); however they lacked tools for implementing their proposal.

Within the high number of existing groupware systems, we are going to deal with distributed synchronous systems, in which diverse users work at the same time from different locations, and we focus on collaborative modeling systems. In these systems, several users typically collaborate in the construction of a design or artefact, working on a shared workspace according to the whiteboard metaphor. This design usually follows the specification of a goal or task. To achieve this goal, users participate in design sessions in which they have a set of objects and relationships available to be placed on the shared workspace, making up a design. This approach can be applied to a group activity, when the problem is a real situation to solve in the scope of a company or organization, as well as to a collaborative e-learning activity, when the system implements a learning method based on problem solving.

The mechanisms or components of the groupware system we study in this article are the most representative ones in these kinds of systems, i.e., the application domain, the way in which the users of a working group communicate, the policies for turn-taking in the use of the shared workspace (floor control), the definition of the tasks to be carried out and the awareness mechanisms. In order to validate our model-driven development proposal for groupware, the SPACE-DESIGN (SPecification and Automatic Construction of collaborative Environments of DESIGN) tool was implemented [Gallardo 07b]. SPACE-DESIGN is a groupware system with support for synchronous distributed collaborative work that allows users to build designs in domains that are well-defined in a way that is external to the system.

There are diverse approaches for building synchronous collaborative modeling systems, which can be domain-specific or claim to be domain-independent. In the latter approach, what the systems really do is to include a higher number of application domains. Some examples of domain-specific systems are X-CHIPS [Wang 00], which works with hypermedia, and Co-Lab [van Joolingen 05], which deals with system dynamics. As far as the systems that include more than one design domain are concerned, CoolModes [Wichmann 06] and Synergo [Avouris 04] are worth mentioning. CoolModes includes a shared workspace that adopts the electronic whiteboard metaphor, in addition to a set of palettes that the system denominates plug-ins and that contain the elements that can be placed on the whiteboard and the relationships that can connect these elements. The power of the system is that the constructed model can be simulated, since the plug-ins have this functionality built in. On the other hand, Synergo also has several fixed palettes with design components that can be connected to each other. One of the strong points of this system is its analysis tool that processes the actions that take place during the work sessions.

As a first conclusion derived from the analysis of the aforementioned systems, it can be seen that the existing domain-independent collaborative modeling systems do not have as much flexibility as it could be expected. The design palettes, which are the most effective representation of the variety of domains, are hard coded in most of these systems, not allowing end users to extend their functionality through the definition of new domains. A second conclusion is that domain-specific systems have many more awareness, communication and coordination mechanisms than domain-independent ones. The effort required to obtain domain independence seems to be the cause for some loss of functionality in other aspects of the system, such as awareness.

Our approach avoids the problems of having to re-design the system for each new application domain. Our meta-modeling approach overcomes this problem and allows us to work with the same system for different domains. Let us think, for example, about a software development company that wants their employees to discuss on a flow chart or on a class diagram. With our approach, it could be done using the same system, adapting it on each occasion to each specific domain.

In the following section, we introduce the possibilities that meta-models offer for the definition of domains external to the systems aimed to support them. To illustrate this, the definition of the domain of use case diagrams in UML in the SPACE-DESIGN system is discussed. In the third section, the support that this system offers to define the tasks that must be carried out is studied. The fourth section focuses on the study of the mechanisms that are used to define structured communication. The fifth section studies how SPACE-DESIGN regulates the use of the shared workspaces. In the sixth section, a description of the tools that SPACE-DESIGN offers for synchronous distributed collaborative support is presented. The seventh section discusses an evaluation of SPACE-DESIGN made by experts in several application domains and by users of the system. Finally, the eighth section presents the conclusions that are drawn from the proposals underlying our work and their implementation in the SPACE-DESIGN system; also, further research is outlined.

2 Domain modeling

SPACE-DESIGN is domain-independent, that is to say, it allows users to work on any domain that fulfils certain characteristics. This groupware system supports domain specification in documents or structures that are external to the system, so that the characteristics of the domains are extracted from them. In order to achieve this, SPACE-DESIGN is based on three levels of representation with different degrees of abstraction for the specification of its main components. A number of approaches have followed a similar multi-level approach, such as the one of Dourish [Dourish 98] or the one proposed by the OMG (Object Management Group) in the MOF (Meta Object Facility)[1]. These levels are represented in XML-based languages with the purpose of providing a computational model. They are as follows:

- Meta-meta-model: This is a global model that defines how all models are characterized. It basically represents the syntax that describes how the domain elements and their relationships can be defined at the smallest abstraction levels. In the meta-meta-model, the elements that will make up each application domain are defined. The possible elements are:
    - Entities: They are the main concepts of the domain. Once they have been specified, they can be connected with other entities. In addition, an entity can be made up of other entities as well.
    - Relationships: They represent conceptual connections that allow communication among entities. Each relationship defines which entity types can communicate and the number of objects that can participate in the relationship.
    - Attributes: They are descriptors that characterize the state of the entities or relationships that have been specified.
- Meta-model: This is a specification of an application domain that follows the syntactic constraints of the meta-meta-model. The use of these meta-models allows users to represent, in an intuitive way, the application domain. Starting from a separation of the elements that form the domain, i.e., entities and relationships between them, a visual representation of the entire domain can be produced. Figure 1 shows a graphical representation of the elements that make up the domain meta-model of the use case diagrams domain. This graphical representation is a hierarchical structure that classifies the entities and relationships that make up the domain of use case diagrams.
- Model: A model consists of specific entities and relationships, and is built as an answer to the requirements expressed in a specific modeling task. It is built according to the meta-model specification. In the example of the use case diagrams, a model is a specific diagram with specific actors, use cases and relationships.

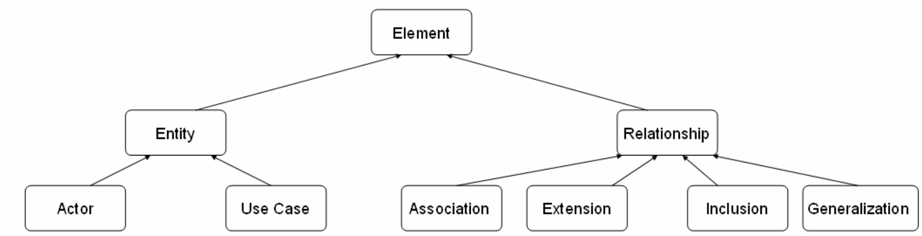

Figure 1: Meta-model of the use case diagrams domain.

[1] www.omg.org/mof/

To have representations of the domain that are external to the system that supports them creates new possibilities. Two examples of tools that use ontologies or meta-modeling aspects to model domains are AToM3 [Lara 02], which is a tool for multi-paradigm modeling that uses graphs for meta-modeling and model-transforming, and WETAS [Mitrovic 04], an intelligent tutoring system that analyzes the correction of the solutions that are built for the problems in a domain which is specified through an ontology. Here, we apply these techniques to the groupware case.

In this line, SPACE-DESIGN has the capacity to process the definition of these meta-models in order to give support to the users' creation of models that fulfil the requirements of the tasks outlined. In order to achieve this, a set of three XML specification files (the domain file, the task file and the communication file) will be processed. In this way, the system will re-configure itself to the specific application domain. Figure 2 shows the XML specification of the use case diagrams domain. This specification follows the constraints that the meta-meta-model defines and is formed according to an XML-Schema document that requires consideration of aspects such as the separation of the semantic part from the graphical representation. This XML specification defines the characteristics of the entities and relationships expressed in the meta-model (see Figure 1). For each element, the XML specification defines its name and its graphical representation. In the case of the relationships, the XML specification also defines the type of entities that can be linked.

```
- <operators>
  - <operator id="actor" area="Use Cases" type="actor" icon="http://space.inf-cr.uclm.es:8080/autoolx/imag/actor.jpg"
      toolbaricon="http://space.inf-cr.uclm.es:8080/autoolx/imag/actor.jpg" link_points="1,2,3,4,5,6,7,8,9,10,11,12">
    - <properties>
        <property name="label" type="String" />
      </properties>
    </operator>
  - <operator id="use case" area="Use Cases" type="use_case" icon="ellipse" toolbaricon="http://space.inf-
      cr.uclm.es:8080/autoolx/imag/circ.jpg" link_points="1,2,3,4,5,6,7,8,9,10,11,12">
    - <properties>
        <property name="label" type="String" />
      </properties>
    </operator>
  </operators>
- <relationships>
    <relationship id="use" operator1="actor" operator2="use_case" link="use_line" directed="yes" />
    <relationship id="include" operator1="use_case" operator2="use_case" link="include" directed="no" />
    <relationship id="extend" operator1="use_case" operator2="use_case" link="extend" directed="no" />
    <relationship id="use_case_gen" operator1="use_case" operator2="use_case" link="generalization" directed="no" />
    <relationship id="actor_gen" operator1="actor" operator2="actor" link="generalization" directed="no" />
  </relationships>
</domain>
```

Figure 2: Specification of the use case diagrams domain using an XML-based language.

In order to verify the validity of the use of meta-models to define application domains that are external to the CSCW system, we have developed several meta-models in different fields. For example, in the field of Software Engineering, a number of domains are intended to support the modeling of software processes (e.g., SPEM[2]) [Duque 06] as well as the building of diagrams used in some stages of the software lifecycle (e.g., use case diagrams [Gallardo 07b], state machine diagrams,

[2] http://www.omg.org/technology/documents/formal/spem.htm

component diagrams, data flow diagrams, task trees, etc.). Moreover, other significant domains have been modeled with our meta-modeling approach, such as digital circuits design, concept maps design, house automation design, etc. In addition, in order to evaluate the versatility of this approach to define application domains, other authors have used meta-models to define collaborative board games for mobile devices [Bravo 05] and to specify domains in collaborative learning environments [Bravo 04b].

This approach for domain specification has evident advantages, such as the availability of re-usable tools, in the form of software components that are configured for or adapted to a specific domain, which supposes a saving of costs and of work for developers. Another advantage is the possibility of interoperability with other tools if standardized forms for the representation of models are used. In addition, it is necessary to emphasize that adopting domain representations that are external to the systems will make the specification of syntactic constraints easier. For example, in the design of use case diagrams, users are prevented from executing incorrect operations such as connecting *Actor* objects by means of *Association* relationships (see Figure 2).

3 Task specification

Once the application domain has been defined by instantiating the corresponding meta-model, it is essential to make a specification of the task to be carried out with the groupware system. In SPACE-DESIGN this specification is made using another XML-based language that allows us to have a computational model that describes the goals that should be fulfilled with the development of the task. This specification represents these goals by means of a set of constraints and requirements [Bravo 04]. Constraints are limitations that must be verified during the task, e.g., to limit the time available for working in the workspace. On the other hand, requirements are formalized statements that specify some characteristics that must be satisfied by the final solution, e.g., an entity with certain attributes must be included in the final design. Constraints and requirements are specified formally, making reference to the elements that make up the domain and that are already specified in the corresponding XML document.

Figure 3 shows an excerpt of the specification of the task of designing a use case diagram including some requirements and constraints that have been defined for this specific case. This specification is read from an external file:

```xml
- <task>
    <description url="http://space.inf-cr.uclm.es:8080/tasks/machine.txt" />
  - <goals>
    - <constrains>
      - <model>
          <entity type="actor" maxallowed="2" />
          <entity type="association" available="no" />
        </model>
      - <others>
          <time unit="minutes" max="90" />
        </others>
      </constrains>
    - <requeriments>
        <entity type="actor" name="user" />
        ...
      </requeriments>
    </goals>
    ...
</task>
```

Figure 3: Excerpt of a task specification in SPACE-DESIGN.

SPACE-DESIGN shows the formulation of the task to carry out in natural language (Figure 4). This is automatically generated from the task specification. The description is extracted from the path indicated in the *description* label (see Figure 3). The constraints are expressed in natural language starting from their formal specification. Requirements are not represented since they are used for later checks on the built solutions. The example in Figure 4 corresponds to the specification in Figure 3.

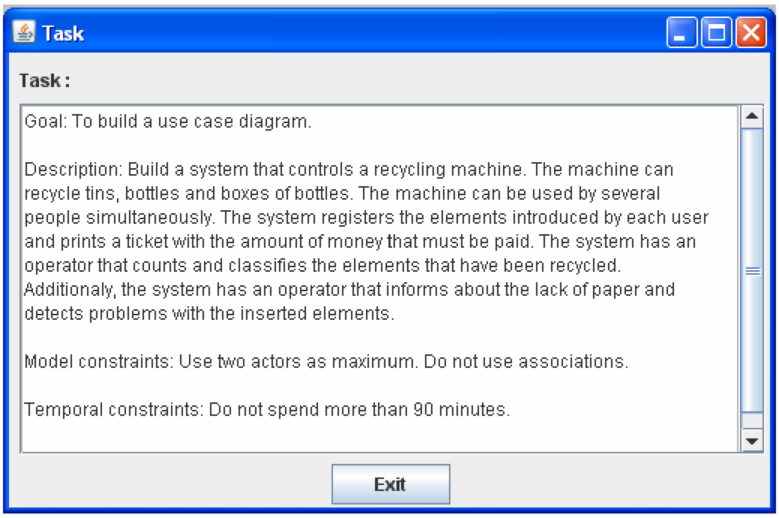

Figure 4: SPACE-DESIGN user interface for the representation in natural language of the task to be carried out.

4 Structured communication

A basic requirement to an effective collaboration between the members of a working group is to support argumentative discussion [Redondo, 04] during design creation. In this case, we propose to incorporate support for argumentative discussion in a distributed synchronous collaborative situation. Thus, each user can present his/her proposals, disagreements, questions or arguments in relation to the work under development [Isaacs 96]. On the one hand, the argumentative discussion is a process that favours collaboration when the users seek to achieve a common goal [Lund 96]. On the other hand, an argumentative discussion allows the individualized analysis of the communicative contributions of each user.

SPACE-DESIGN makes argumentative discussion possible by means of the so-called structured chat [Gallardo 07a], which is a chat with sentence openers. In this sense, the system provides support for the particular necessities of each task or domain at communication level by specifying those sentences, opinions or commentaries that are often used in that specific setting by means of an XML-based language [Bravo 04a]. This technique allows the reuse of previous conversational structures. The structured chat tool extracts the sentences than can be used from the XML document (see Figure 5) and automatically generates the user interface.


```
<?xml version="1.0" encoding="UTF-8"?>
<communication>
  <message id="m1" requiresText="true">
    <text>I think that... </text>
  </message>
  <message id="m2" requiresText="true">
    <text>Why...?</text>
  </message>
  <message id="m3" requiresText="true"> [2 lines]
  <message id="m4" requiresText="true"> [2 lines]
  <message id="m5" repliesTo="m1"> [2 lines]
  <message id="m6" repliesTo="m1">
    <text>I don't think so</text>
  </message>
  <message id="m7" repliesTo="m2"> [2 lines]
  <message id="m8" repliesTo="m2" requiresText="true">
    <text>Because...</text>
  </message>
</communication>
```


Figure 5: Excerpt of XML specification that defines a set of conversational acts.

This user interface contains buttons with labels to express the conversational acts. Thus, when the user clicks on the corresponding button, the text of the label (e.g., “I think that…”, “There’s a mistake in…”) is introduced in the chat text box, so that the user only has to complete the sentence. According to the specification for structured communication, in the conversation there are some initial messages and other messages that reply to others (see Figure 5). There are also messages that may have more than one reply, and even communication cycles are possible. For instance, if a user states an interrogative sentence (“Why…?”), the receivers of the question will have the buttons that reply to the question (“Because…”) or those to initiate other independent sub-conversations activated in the chat. Besides a structured communication, this tool can be configured to permit a free use of the chat, that is to say, each user could communicate without using predefined sentences simply by writing a complete sentence.

Other elements on which to base the arguments of the discussion are the entities and relationships of the model being designed. Thus, if one of those entities or relationships is being discussed, the users can include a reference to that element in the conversation. In order to achieve this, the structured chat contains a dropdown list (see Figure 6) that allows the selection of any domain object in the model.

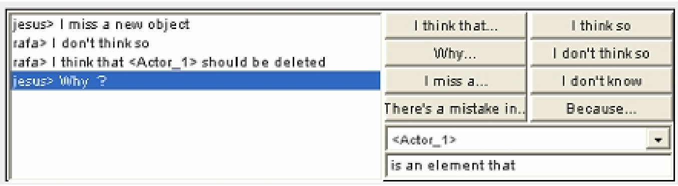

Figure 6: Argumentative discussion about an element of the built model.

5 Coordination mechanisms

In order to develop a groupware system to support the necessities of a working group, it is essential to consider how collaboration takes places and how to harmonize each personal effort. However, it is not possible to provide collaboration rules that can be applied in any collaborative setting, since these rules can change according to the roles of the implied users, the application domain and the collaborative habits acquired in previous experiences. Thus, SPACE-DESIGN adopts a flexible approach, making it possible to apply three regulation models for the working turn (or floor control):

- Simultaneous work. Group members can work in the shared workspace concurrently. The access and concurrent work is not restricted at all, since the established rules do not give preference to any user for the use of the shared workspace. Obviously, this approach can generate a number of conflicts. In addition, blocking is used to avoid inconsistent states of the artefact produced.
- Turn assigned by agreement. The access to the shared workspace is made by agreement. Any user who wishes to make a modification in the shared workspace must send a proposal to the rest of the users in the session, asking for permission to access to the common space. Once the proposal has been formulated, the rest of the users have to go through a voting process in which they express their agreement or disagreement with the proposal.
- Turn assigned by request order. In this case, the permission to work is assigned in the same order in which the requests take place. When a user makes a request, he/she will have to wait until the next turn is available (and all the partners who had formulated the same request before him/her have

finalized their work). This protocol follows a FIFO (First Input First Output) policy.

In SPACE-DESIGN, the model to regulate the use of the shared workspace is established in the XML document that specifies the task to develop. The mechanism based on an assignment by agreement of the working turn is shown in Figure 7. It shows how several users make their requests as well as the answers given by the rest of the working group members. Each participant knows each partner's proposals and answers since these are public. However, depending on the configuration of the collaborative setting, it would be possible not to show the identities of the participants, so that problems arising from lack of privacy would be avoided.

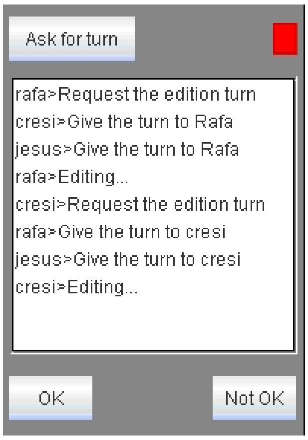

Figure 7: Coordination panel.

6 Collaboration in SPACE-DESIGN

In SPACE-DESIGN, the services to support synchronous distributed collaboration are based on the ISSC (Infrastructure of Synchronization for Collaborative Systems) infrastructure [Bravo, 04c]. ISSC has been implemented by making use of JSDT[3] (Java Shared Data Toolkit). ISSC offers a higher level of abstraction and makes the creation of collaborative systems easier, providing them with a suite of collaborative tools to be incorporated in the shared workspaces as well as management facilities such as an schedule of working sessions, a tool for defining working groups, etc. The following tools have been integrated in SPACE-DESIGN (Figure 8) with the aim of supporting collaborative work:

- Structured chat: This supports argumentative discussion as described in Section 4. It processes the XML document that specifies the communication aspects in order to automatically generate the user interface with the specified sentences and sentence openers.

[3] java.sun.com/products/java-media/jsdt/

- Session panel: This shows the photos and names of the participants in the session; the name is drawn in the same colour of the user's tele-pointer. In this way, a basic awareness [Carroll, 03] mechanism is implemented, so that any user has knowledge of the other participants and of their work.
- Electronic whiteboard: This is the shared whiteboard where the different models are built in a collaborative way when carrying out the proposed tasks.
- Entity toolbar: The user can insert in the whiteboard any of the domain objects. To do this, she/he must use the entity toolbar that is automatically generated by SPACE-DESIGN according to the elements specified in the XML document that specifies the application domain.
- Relationship toolbar: The relationships among the entities are created using this toolbar, which is also automatically generated in the same way as the entity toolbar.
- Options toolbar: This toolbar includes useful functionalities, such as viewing the proposed task, modifying the current configuration (beeps, colours, etc.) or consulting working sessions.
- Drawing toolbar: SPACE-DESIGN integrates tools to support the drawing of figures in several colours, the introduction of text in the whiteboard and the selection and deletion of elements from the model.
- List of interactions: This tool shows a brief description of the actions that each user has carried out in the shared workspace. Thus, the knowledge of the work carried out by the other participants is favoured. Also, the study of how the different models were carried out is facilitated; this constitutes a mechanism that helps people outside the working group to understand the elaboration of the designs.
- Coordination panel: This allows the assignation of the working turn depending on the specified collaboration rules, as described in Section 5.

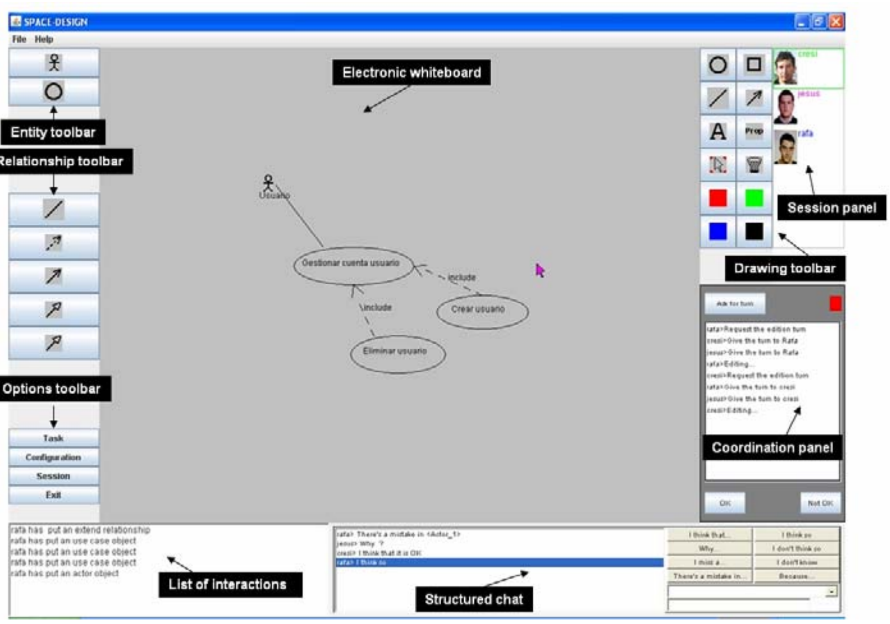

Figure 8: The SPACE-DESIGN modeling workspace for the use case diagrams domain.

7 Evaluation

In order to obtain a validation of the SPACE-DESIGN approach, two kinds of experimental activities were carried out. In the first one, a modeling of different application domains was made and then the system was put in action with each of these domains. Later on, an expert in each application domain was asked for an evaluation of the SPACE-DESIGN functions and of the representations of both task and application domain. In the second kind of evaluation activities, the collaborative solving of a task using SPACE-DESIGN was proposed to two users. After solving the task, they were requested to answer a questionnaire in which they evaluated the functionality of collaborative modeling offered by SPACE-DESIGN. The results of these experiments are discussed below.

7.1 Experts' opinion

This first validation activity was approached using a population made up of seven teachers of the University of Castilla - La Mancha. Each teacher was an expert in a different application domain. The seven application domains in which these teachers are experts are: Bayesian networks, Multi View Process Modeling Language (MVP-L) [Rombach 91], UML Package diagrams, Digital circuits, UML Use case diagrams, Conceptual maps and UML Data flow diagrams. Each one of these application domains was specified according to the XML-based language used by SPACE-

DESIGN so that the system supported the design of models in each of them. We presented the experts how to externally specify the application domain and the task to be solved. Putting the system in action, and using this documentation, the experts evaluated different aspects of the system. The average assessments of these aspects are shown in Figure 9. Every aspect was evaluated using a number ranging from 1 (very bad) to 5 (very good). Figure 9 contains all aspects that were evaluated and the average value, in all cases an average value equal to or greater than 3 was obtained.

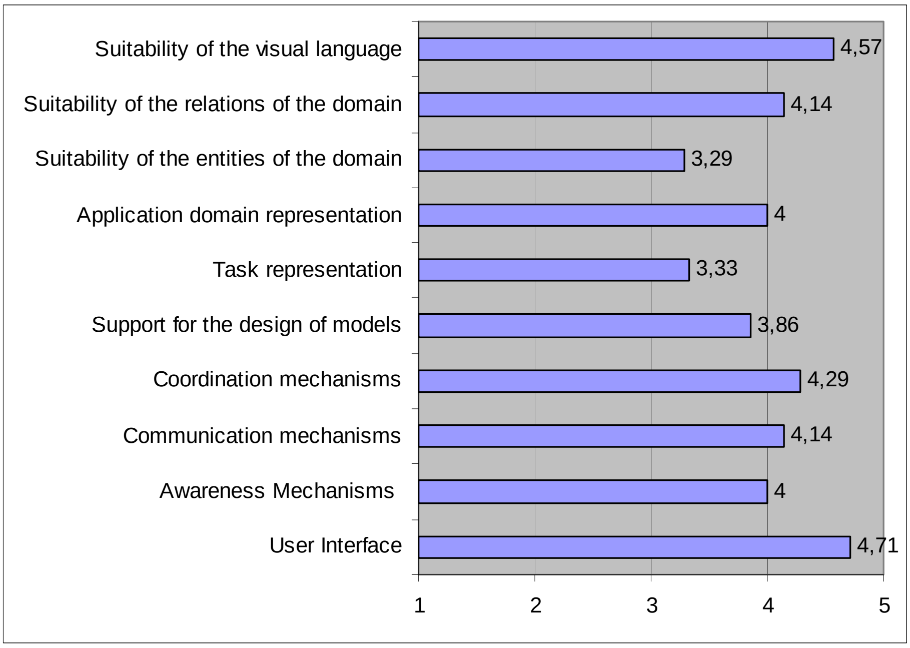

Figure 9: Quantitative evaluation by the experts.

After the first part of the questionnaire where a quantitative evaluation of the system was completed (see Figure 9), a comparative evaluation of SPACE-DESIGN with other systems already used by the experts as well as suggestions for system improvements were requested.

In relation to the comparative evaluation, the experts were asked if they had used any other system for the collaborative design of models in the application domain where they worked on. The answer was that nobody had used such a system. Moreover, the experts were asked if they had used any modeling software tool in which the application domain was defined externally to the tool. Much of the respondents had never used this kind of tools.

Finally, the experts suggested a number of improvements in the system from which some future lines of work arise. The main suggestions are as follows:

- The chat could incorporate a facility that supports audio communication.
- Integrating a video-conferencing tool within the system.
- Introducing more textual information about the graphical models designed.

- Representing changes in the models graphically.
- Incorporating contextual information to objects as the user who created the object or the class to which it belongs.
- Creating a user with the moderator role in the group work.
- Using the XMI[4] standard to represent the application domain and models designed, thus improving interoperability with other tools.
- Validating the system with a set of new application domains such as all UML diagrams, modeling of business processes with BPMN[5], state machines, etc.

7.2 The system in action

The task proposed to the users participating in the evaluation was to solve a problem by designing a model belonging to the application domain of UML Use case diagrams. Once the users had completed the design of the model, they made a qualitative evaluation of the functionalities of SPACE-DESIGN. Thus, they made a number of suggestions recommending some improvements in the system as follows:

- Improving the mechanisms that allow selecting an element of the model that is being designed.
- Including group scroll bars on the interactive blackboard.
- Improving awareness mechanisms so that the actions of the partners could be perceived faster and better.
- Introducing tip text on the buttons.

8 Conclusions

In the work reported here, we have built a groupware modeling system, SPACE-DESIGN, which uses XML-based languages to reconfigure itself and adapt to new application domains, new tasks, and new collaboration and communication rules. This approach, which takes advantage of model-driven development and meta-modeling techniques, makes SPACE-DESIGN more versatile than other current proposals which only support one application domain or a single kind of communication.

The development of domain-independent groupware systems and the possibility of instantiating them by means of the specifications of particular domains supposes a considerable saving of work for developers and authors of collaborative systems. Somehow, our different experiences in designing groupware systems approaching areas (domains) such as Software Engineering [Duque, 06], collaborative board games [Bravo, 05] and collaborative design systems (e.g., digital circuits, home automation, etc.) [Bravo, 04a] contribute validating this work.

However, there is a gap in tools supporting collaborative domain-independent modeling. SPACE-DESIGN fills this gap with the use of meta-models and XML-based languages that specify externally to SPACE-DESIGN the most important characteristics of this groupware modeling system. The SPACE-DESIGN approach was validated in the experimental activities developed. In these activities the SPACE-

[4] http://www.omg.org/technology/documents/formal/xmi.htm
[5] http://www.bpmn.org/

DESIGN functionalities were valued positively by experts and users in most cases. However, it must be kept in mind that several aspects (e.g., the manipulation of the relations of the application domain or the awareness mechanisms) should be improved according to the users' opinion. At present, we are focusing in improving the system as suggested.

In parallel to this work, a complete framework [Gallardo 07b] to support the development life-cycle of this type of synchronous collaborative modeling systems is being developed. This work must extend the lines approached in this article by building a methodological framework, a conceptual framework made up by some ontologies [Duque 07b] and a technological framework that all together will support the development of that kind of systems.

In addition, on the basis of the models described in this article, a methodological process will be developed to automate the analysis of the users' interaction and collaboration [Duque 07a], analyzing both the actions and the artefact carried out by the users. More specifically, the result of this analysis process will be a set of variables that will evaluate and characterize properties of both the work carried out and the model (solution) built.

Acknowledgements

This research has been partially supported by the *Junta de Comunidades de Castilla-La Mancha* (Spain), *Consejería de Educación y Ciencia*, in the PCI08-0069-7887 and PAC07-0020-5702 projects, and by the *Ministerio de Educación y Ciencia* (Spain) in the TIN2005-08945-C06-04 project.

References

[Avouris 04] Avouris, N., Margaritis, M., Komis, V.: Modelling interaction during small-groups synchronous problem-solving activities: The Synergo approach. In Proceedings of the 2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction, Maceio, Brasil.

[Bravo 04a] Bravo, C., Gallardo, J., García-Minguillan, B., Redondo, M.A.: Using specifications to build domain-independent collaborative design environments, In: Y. Luo (ed.): Cooperative Design, Visualization and Engineering. Lecture Notes in Computer Science, Vol. 3190, Springer-Verlag, 2004, 104-114

[Bravo 04b] Bravo, C., Redondo, M.A., Gallardo, J., Ortega, M., Bravo, J.: Organización de Actividades y Especificación de Dominios en un Entorno de Aprendizaje Colaborativo del Diseño Revista Iberoamericana de Inteligencia Artificial, 8 (24), 2004, 97-107

[Bravo 04c] Bravo, C., Redondo, M.A., Mendes, A.J., Ortega, M.: Group Learning of Programming by means of Real-Time Distributed Collaboration Techniques, In: J. Lorés, R. Navarro (eds.): Invited and reviewed papers from Interacción'2004, Springer-Verlag, 2004

[Bravo 05] Bravo, C., García, J., Gallardo J.: Juegos de Tablero Colaborativos para PDA: Especificación, Generación Automática y Experiencias. Actas del VI Congreso de Interacción Persona-Ordenador, 2005, 119-126

[Carroll 03] Carroll, J. M., Neale, D.C.,Isenhour, P.L., Rosson, M.B. McCrickard, D.S.: Notification and awareness: synchronizing task-oriented collaborative activity, Int. J. Hum.-Comput. Stud. 58, 5, 2003, 605-632

[Dourish 98] Dourish, P.: Using meta-level techniques in a flexible toolkit for CSCW applications, ACM Trans. Comput.-Hum. Interact. 5, 2, 1998, 109-155

[Duque 06] Duque, R., Bravo, C.: The usefulness of CSCW systems in Process-sensitive Software Engineering Environments. In: Y. Luo, (ed.): Cooperative Design, Visualization and Engineering. Lecture Notes in Computer Science. Springer-Verlag, 2006, 156-163

[Duque 07a] Duque, R., Bravo, C.: A method to classify the collaboration in CSCL systems In: Beliczynski, B. (ed.): The eight International Conference on Adaptive and Natural Computing Algorithms (ICANNGA'2007). Lecture Notes in Computer Science 4431. Springer-Verlag, 2007, 649-656

[Duque 07b] Duque, R., Bravo, C., Ortega, M.: Towards an ontology to conceptualize solution analysis tasks in CSCL environments. In: Priss, U., Polovina, S. Hill, R. (Eds.). The 15th Intenational Conference on Conceptual Structure (ICCS 2007). Lecture Notes in Artificial Intelligence 4604. Springer-Verlag, 2007, 509-512.

[Gallardo 07a] Gallardo, J., Bravo, C.: Coordination and Communication Protocols for Synchronous Groupware: A Formal Approach. In: D. Cunliffe (ed.): Proceedings of the Second IASTED Conference on Human-Computer Interaction. ACTA Press, 2007, 55-61

[Gallardo 07b] Gallardo, J., Bravo, C., Redondo, M.A.: An ontological approach for developing domain-independent groupware. In Proceedings of the 5th IEEE International Workshop on Distributed and Mobile Collaboration (DMC 2007), Paris, France. (Accepted, in press)

[Garrido 07] Garrido, J.L., Noguera, M., González, M., Hurtado, M.V., Rodríguez, M.L.: Definition and use of Computation Independent Models in an MDA-based groupware development process, Science of Computer Programming, 66, 2007, 25–43

[Grudin 93] Grudin, J.: Groupware and Cooperative Work: Problems and Prospects. In R.M. Baecker (ed.): Readings in Groupware and Computer Supported Cooperative Work. Morgan Kaufman Publishers, 1993, 97-105

[Isaacs, 96] Isaacs, W.N.: The process and potential of dialogue in social change, Educational Technology, 2030, 1996

[Karsai 03] Karsai, G., Agarwal, A., Ledeczi, A.: A metamodel-driven mda process and its tools. In: J. Bezivin, M. Gogolla, (Eds), Proceedings of the 2nd UML Workshop in Software Model Engineering, San Francisco, USA, 2003

[Lara 02] Lara, J., Vangheluwe, H.: AToM3: A Tool for Multi-formalism and Meta-modelling, In Proceedings of the 5th international Conference on Fundamental Approaches to Software Engineering, R. Kutsche and H. Weber (Eds), Lecture Notes in Computer Science, vol. 2306. Springer-Verlag, London, 2002, 174-188

[Lund 96] Lund, K., Baker, M.J., Baron, M.: Modelling Dialogue and Beliefs as a Basis for Generating Guidance in a CSCL Environment, Proceedings of the International Conference on Intelligent Tutoring Systems (eds. C. Frasson, G. Gauthier, A. Lesgold), 1996, 206-214

[Mitrovic 04] Mitrovic, A., Suraweera, P., Martin, B.: The role of domain ontology in knowledge acquisition for ITSs, In Proceedings of the Seventh International Conference on Intelligent Tutoring Systems, ITS2004, Maceio, 2004, 207-216

[Poltrock 98] Poltrock, S., Grudin, J.: CSCW, groupware and workflow: experiences, state of art, and future trends, In CHI 98 Conference Summary on Human Factors in Computing Systems CHI '98, ACM Press, New York, NY, 1998, 119-120

[Redondo 04] Redondo, M.A.: Bravo, C., Ortega, M.: Contextualized Argumentative Discussion for Design Learning in Group, Reviewed Selected Papers from the 5th Conference of the Spanish Association for HCI (AIPO), Interación 2004, Springer-Verlag, 2004

[Rombach 91] Rombach, H. D.: MVP-L: A Language for Process Modeling In-The-Large, Technical Report, University of Maryland at College Park, 1991

[van Joolingen 05] van Joolingen, W.R., de Jong, T., Lazonder, A.W., Savelsbergh, E.R., Manlove, S.: Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. Computers in Human Behavior, 21, 2005. pp. 671-688.

[Wang 00] Wang, W., Haake, J., Rubart, J., Tietze, D.: Hypermedia-based support for cooperative learning of process knowledge. Journal of Network and Computer Applications, 23, Academic Press. 2000.

[Wichmann 06] Wichmann, A., Kuhn, M., Hoppe, U.: Communication through the artefact by means of synchronous co-construction, In Proceedings of the 7th international Conference on Learning Sciences, Bloomington, Indiana, June 27 - July 01, 2006, 825-831