# Collaborative Task Modelling on the Web

Marco Manca, Fabio Paternò, Carmen Santoro

# Collaborative Task Modelling on the Web

Marco Manca, Fabio Paternò, Carmen Santoro

CNR-ISTI, HIIS Laboratory
Via Moruzzi 1, 56124 Pisa, Italy
{marco.manca, fabio.paterno, carmen.santoro}@isti.cnr.it

**Abstract.** Task modelling is a widely recognized activity when designing interactive applications. In this perspective, it is the meeting point between various stakeholders. However, most of the automatic environments that currently allow task modelling only support single users, thus limiting the possible interactions and discussions amongst them. In this paper we present Collaborative CTT, a new Web-based multi-user tool for specifying task models. The tool allows several users, who may even be physically separated, to work on the same model at the same or different time. Among its features, the tool includes mechanisms specific for this type of HCI modelling in order to support coordination, communication and mutual awareness among participants. We discuss the aspects we have addressed in designing the task modelling tool, its main collaborative features, and also report on user feedback gathered through formative tests.

**Keywords:** Task models; Collaborative Modelling; Responsive Web

## 1 INTRODUCTION

Task modelling is a useful method for various activities in the user interface design and development process. It helps to better understand the application domain, record the results of interdisciplinary discussions, support user interface design, usability evaluation, and documentation. The nature of task modelling as a multi-disciplinary process is widely accepted: in order to properly carry out task modelling it is important to involve various experts, stakeholders, designers and users. In addition, with the increasing need for collaboration among stakeholders, and also the need to reduce costs, which often forces teams to collaborate also from different locations, it is becoming increasingly important to create a shared understanding and joint representations of the interactive systems being designed. As a consequence, enabling interactive collaborative modelling in this area can prove to be valuable as it could make collaborations more effective and productive. For instance, in some situations it might be interesting for UI designers to carry out the modelling work together with users at the same time but from different locations, or it would be interesting for the members of the same design team to be able to carry out the modelling activity in a collabora-

tive manner. Unfortunately, most of the tools that support task modelling only allow for single users, and they do not enable various users to share the task models and collaboratively edit them. Thus, we judged it interesting to investigate the opportunities offered by a multi-user, collaborative, Web-based task modelling tool.

In particular, the main goals of this work are to design a tool able to:

- Support a shared view of task models across various devices and associated users;
- Support concurrent editing of task models by multiple users;
- Provide mechanisms to synchronize editing of some parts of the task models;
- Provide mechanisms to create mutual awareness of the concurrent activities in the modelling process.

To these ends, we have opted for a responsive Web-based implementation because this would facilitate its adoption given its wide interoperability across various devices, and we have adopted the ConcurTaskTrees (CTT) notation [19] since it is widely known in the task modelling community.

In the paper after reviewing the state of the art, we introduce some scenarios that have motivated our work; next we discuss how we have designed the features that support the collaboration in editing the task model, describe how it has been implemented, and report on two user tests. Lastly, we draw some conclusions and provide indications about future work.

## 2 STATE OF THE ART

Collaborative modelling has received some attention in some domains. For example, Collaborative Protégé and WebProtégé [12] are extensions of the existing Protégé (http://protege. stanford.edu), also enabling collaborative ontology editing. Support in this direction is also provided by some commercial tools. For example, VPository (from Visual Paradigm, http://www.vpository.com/) offers a central repository for storing user's software design projects with version control capabilities and, in this regard, it is far from providing the support offered by a truly collaborative environment. In [18] a review of tools that support collaborative processes for creation of various forms of structured knowledge was presented. However, we note that little attention has been paid to collaborative modelling support in HCI so far.

Before analysing this aspect more closely, it can be useful to highlight the difference between tools for collaborative task modelling, and tools supporting task modelling of collaborative applications. The first case is the one which we address in this paper, i.e. we analyse tools enabling users to jointly create a model in remote or co-located places (e.g. collocated groups of designers in a room, team members in distant places), in a synchronous or asynchronous manner. In the second case, the focus is on tools allowing designers to model systems where multiple users act in a collaborative manner. So, in the first case the multi-user dimension concerns the users of the modelling tool, in the second case it regards the users of the application to design.

On the one hand, the latter case (tools supporting task modelling of collaborative applications) has been the subject of several contributions. Indeed, proposed notations for modelling multi-user applications include: the COMM (COlaborative and Multi-Modal) notation and its on-line editor for specifying multi-user multimodal interactive systems [13]; CTML [26, 3], a task-based specification framework for collaborative environments, consisting of a language and a tool for editing/animating CTML models; CUA (Collaboration Usability Analysis) [22], a modelling technique allowing designers to model the main features of a group work situation that will affect groupware usability. Other proposals along the same lines have been put forward by Penichet et al. [21], van der Veer et al. [25], Guerrero-Garcia et al., [8], Giraldo et al. [9], Molina et al. [15, 16]. On the other hand, little has been proposed for collaborative task modelling tools. Even recent proposals [1] have addressed the issue of adapting to various device types but not collaborative aspects. Some basic support for collaboration during task modelling has been provided in tools such as HAMSTERS [14] and CTTE [17], which enable users to re-use fragments of task models (even created by other users) within their own task model specifications. However, our goal is to provide a truly collaborative tool in which users actually share the same model, which they can collaboratively modify even at the same time.

A literature review of approaches in the area of collaborative modelling, although not specifically focused on task modelling is in [23], other proposals still in the same area are [10, 24]. Some degree of collaborative support is provided by FlexiLab [11], a UI multi-model editor for HCI implemented as a Web-based application. It mainly supports the possibility of interactively sending fragments of the model from one user to another. This allows several designers to work on separate fragments of the same model, which is especially useful when dealing with large models. While FlexiLab provides some level of collaboration and communication support (for example, it supports sharing models by sending model fragments from one device to another), our proposal addresses the concurrent editing of the same model by multiple users with multiple devices, which implies additional features also for mutual awareness and coordination (e.g. sharing focus, locking a task for editing).

In this area, one relevant experience to mention is SPACE-DESIGN [2, 5], which is a synchronous, generic (i.e. domain independent) collaborative modelling tool, which is extensible and also reconfigurable for a specific domain. SPACE-DESIGN has been used for task modelling through the CTT notation [4]: starting with a CTT specification, the tool adapts its UI to provide some collaborative support for modelling with this notation by including widgets for awareness, communication and coordination. Such paper also reports on a user test that indicates that a generic collaborative modelling tool has advantages in comparison to the use of a single-user tool (such as CTTE) combined with a shared window system such as NetMeeting, especially in regards to the awareness mechanisms offered. The test indicated that when using SPACE-DESIGN, fewer situations of conflict occurred with respect to the alternative setting (CTTE+NetMeeting). While we agree on the fact that the collaborative modelling approach is a more suitable solution, we note that SPACE-DESIGN was able to support only a limited number of basic modelling functionalities (e.g. create, read, update and delete models).

Another relevant experience has been Quill [6, 7], a Web-based development environment that aimed to enable various stakeholders of a Web application to collaboratively adopt a model-based UI design. Quill attempted to support several users to concurrently participate in a common work project in a distributed fashion with live updates. In Quill each user has a specific role (junior or senior), which provides access to specific features of the application. Quill also has a revision control mechanism by which the changes suggested by juniors are passed for review to the senior who has the responsibility for committing them. Similarly to Quill, the results of Collaborative CTT can be included in a comprehensive MBUI-development framework (such as MARIAE [20]). However, in our case the mechanism for deciding the changes is more flexible since rights can be assigned dynamically and not statically to the users. Overall, we can conclude that our proposal addresses an area still underexplored concerning the possibility of supporting collaboration in task modelling.

## 3 TARGET SCENARIOS

The design of the tool has been driven by some scenarios that we briefly describe in this section and are based on our experience in teaching task modelling and in research projects in which task modelling has been used.

*Educational use in the classroom*. In our experience teaching how to create and modify a task model may not be trivial. Once the conceptual aspects have been introduced and studied there is a strong need to do some concrete exercise to better understand how to apply in practise the concepts. An effective exercise is to develop the task model together (teacher and students) in a laboratory in which all students have their own computer (it can be a PC or a tablet or a smartphone). The teacher can start the modelling to show how to approach the associated issues and the students can see the results directly in their devices, while at the same time they can browse the model in order to analyse its features without immediately make any change. At some point the teacher may want to highlight some parts of the model and so impose his/her view on all the devices, centred on the selected part. Once students start to be familiar with the modelling activity it can be useful to gradually allow them to directly carry it out. A good exercise is to make some extension or some modification to the model in the class exercise given that creating a task model from scratch may still be premature. Thus, the teacher may want to ask specific student(s) to detail how a high level task should be carried out or perform some modification on a part of the model developed. This implies the need to assign the possibility of editing the shared model to a specific student and make it possible that her modifications are updated in the teacher and other students views.

*A workgroup aiming at designing an application.* An example of sector in which CTT has been often applied is the air traffic control domain[1] in which the design decisions need to be carefully analysed in order to prevent human errors that can even threaten human life. In designing such applications it is important to involve all the

[1] https://www.eurocontrol.int/ehp/?q=node/1617

relevant stakeholders, such as the air traffic controllers, the application developers, the experts in the relevant regulations. When such groups are in the same room (same place/same time) their discussion could be more effective with a collaborative modelling tool. Thus, they could start the discussion with the task model developed by one of them, and the others can point issues associated with some design decisions using a shared focus, and more clearly indicate alternative ways to accomplish some tasks by directly editing the shared model.

*Distributed synchronous workgroup (different places/same time).* For various reasons the meetings amongst the various stakeholders in the same room are not always possible. Thus, it is useful to support the possibility of collaboratively editing the task model remotely. In this case there is a need for additional tools that support the communication among the participants (e.g. a chat), to have a precise indications of what editing has been made by each involved participant (e.g. a shared log) and to have dynamic feedback on what the other users are doing.

*Distributed asynchronous workgroup (different place/different time).* In some cases is not even possible to arrange a remote meeting because of work constraints. Thus, it is still useful to have the possibility of sharing a model, which can be eased by some cloud support, and facilitate its collaborative editing. In this way when one participant accesses the shared model s/he can work on the modifications carried out by the others. There can be some different opinions regarding how to design some parts, thus it is still important on the one hand to give all participants the possibility of proposing and discussing their solutions, and on the other hand to make a final decision, which can be taken by the moderator or through a vote.

# 4 THE DESIGN OF THE COLLABORATIVE FEATURES

In this section we describe how we addressed the requirements raised by the target scenarios in the proposed tool.

## 4.1 Roles and access rights in handling task models

In terms of roles we have adopted a solution in which there is a 'Moderator' who is the user who starts the collaborative session and invites the other participants ('Collaborative Users'). The latter can have various ways to participate, which are defined by their assigned access rights.

Such rights/authorizations are related to the ability to modify the model, invite further participants, and/or the possibility to assign the shared focus to other members.

Thus, all users can read the models (e.g. visualise, navigate, etc.), but only those who have received the corresponding rights can modify them (a locking mechanism is provided in order to avoid that the models get inconsistent states due to e.g. simultaneous changes).

The environment supports dynamic groups, thus users can be added or leave the collaborative session at any time.

### 4.2 Enhancing mutual awareness among users

Users can independently browse the task model. When they set the focus on one task then their personal view is adapted in such a way to centre the entire model around that task. Figure 1 shows an example in which two users have different focuses on the same model at the same time, and thus receive different views of it. One user has the focus on the *EnableAccess* task while the other on *WithDrawCash*. The red circles in Figure 1 indicate the number of task that are not visualized because they are out of the screen area.

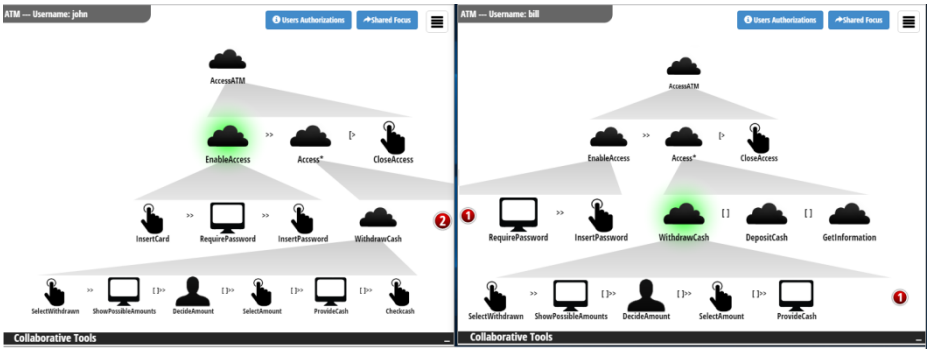


**Figure 1**: Example of multiple views of the same task model

Figure 1 refers to the initial version of the application. In the second version, we added some cues **for enabling users be aware of which task the other users are currently focusing on**. The various local focus of the other users are represented through small circles shown near the correspondent tasks, to highlight the part of the model that is being considered at that time by the other users. The circles have different colors (each colour is associated to a different user) and contain the initial letters of the name of the corresponding user (see Figure 2).

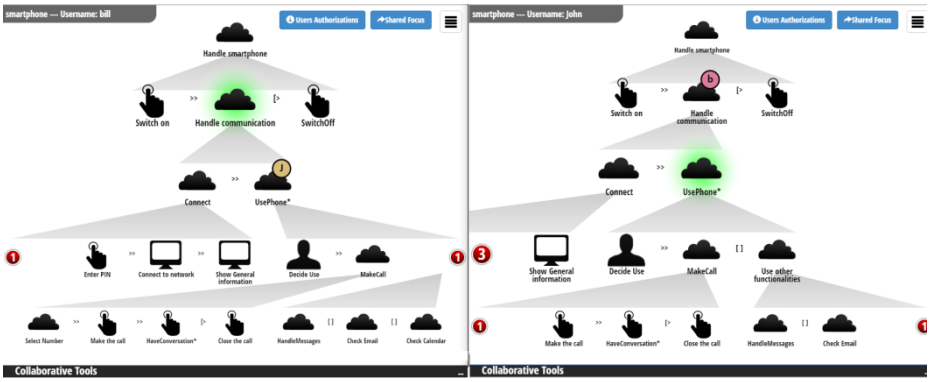


**Figure 2**: Task models annotated with representations of the local focus of the other users

In addition, each user has an **event logger** panel (see Figure 3 bottom-left part), listing in temporal order all the actions carried out by each user while collaboratively working on a task model. For each action it indicates a timestamp, the user who carried it out, the type of action (e.g. lock, unlock, edit, set shared focus, update temporal operator), and the task(s) involved. The logger considers actions carried out even at different times by different groups of users on the model. It is worth noting that, in the second version of the tool, we decided not visualising anymore the lock/unlock events generated when an authorised user performs a model modification, in order to avoid too long lists of events. Indeed, the lock/unlock events are very frequent and in any case a graphical feedback is provided to users in the main area of the application to highlight the occurrence of such events.

### 4.3 Coordination between users collaboratively handling a task model

One further issue addressed in this work has been how to design flexible collaborative editing while supporting an efficient and coordinated way to work. This has been addressed by considering the typical hierarchical structure of task models. In order to support flexibility, users are for example allowed to change at the same time parts of the tree-like task model structure that are independent each other, which means that there is no intersection between the subtrees currently modified by the users. Indeed, modifying a task can involve different types of possible changes, e.g. the user can delete the task and some/all of its children, change name, type, category, associated platforms, specify whether the task is iterative or optional, modify its description, associated objects and pre/post-conditions, if any. In order to coordinate the editing work, when a user is modifying a task, that task and its subtasks (in practise the subtree having as its root the task currently edited) are 'locked' to avoid concurrent changes on that part of the task model by other users at the same time. When this **locking mechanism** occurs, all the other users participating in the collaborative session receive a notification of the lock and the locked task will be highlighted in red in their view of the task model so that they are aware that it will not be possible to edit it (see Figure 3) anymore until the other user unlocks it. When a user locks a task, a timeout is set so that if the lock is not released within the defined time interval then the system performs an automatic procedure to release all the locked tasks.

In particular, Figure 3 shows the user interface of Collaborative CTT (initial version) while modelling an ATM (Automated Teller Machine) system. As you can see, the logger panel, the voting system and the chat are located in the bottom part of the user interface and can be hidden when more screen area is necessary to edit the model. There are two users: John (on the left side), and Bill (on the right side). On the left side John selected the Access task and started editing it: in this way he locked the selected task and all its subtasks. This event is communicated to the server-side part of the environment, which updates the model and sends this information to all clients involved in the collaborative session. Each connected client receives the lock infor-

mation and automatically updates the interface by adding a red background to the subtree locked (see Bill's view in Figure 3). When the tasks are locked other users cannot edit them.

When the editing is finished the locked tasks are released and all users are notified of the unlock operation. Moreover, users will also be notified of the changes in the task attributes in a temporary banner shown in the top area and in the event console log, if a new task has been added (the new task is highlighted with a blue background in the other users' view) or if a task (and its subtasks) is deleted. The user who plays the role of moderator also has the possibility to reject modifications carried out by other users when s/he deems them inappropriate.


**Figure 3.** Example of task modification in Collaborative CTT.

The rationale behind how we have designed the locking was rooted in the hierarchical nature of the task model specification. Since a high-level task is described hierarchically in terms of its sub-tasks, by locking the entire sub-tree we aim to prevent two users editing parts that are semantically tightly connected simultaneously.

In the initial version of the application the locking mechanism was activated as soon as the user opened the panel for editing the task (see an example in Figure 3, left side). However, sometimes users just open this panel to see additional information about the task and not necessarily to edit it, so these cases do not really require the use of a locking mechanism. Thus, in the second version we decided postponing the time when the locking mechanism is actually activated: it is carried out only when the user actually selects the operation to do (e.g. add/edit/delete task).

By considering in particular the target scenario of educational use in the classroom, and our experiences in the discussions carried out during task modelling activities, we noticed that often there is a need for sharing the same focus on the task model. During

a collaborative session, it can happen that different users select different tasks and have different model layouts in front of them at the same time, depending on the task currently having the focus. This supports a flexible way to work but at some point there could be the need to discuss some specific parts of the model, and thus it is important that all users have the same model layout in front of them with the part under discussion shown in the central part of the working area. In Collaborative CTT this is achieved through the possibility to **set a shared focus**. This operation allows all users participating in the collaborative session to coordinate their focus on a particular task (only if they have the corresponding authorization). The selected task will be placed in the centre of their working areas, it will be highlighted in green and the icons associated to this task and its immediate siblings will be enlarged, while the presentation of the neighbouring tasks gradually decrease in size when progressively moving further away from the task that currently has the focus.

### 4.4 Supporting communication between users

In the collaborative application we provided support for communication between the participating users by means of a **chat**, which is especially useful when the involved users are not in the same place. Figure 3 shows the chat (in the first version of the tool). In the second version we provided users with the possibility to interactively select tasks within the chat. By means of typing the *[task]* keyword within the chat, the tool shows a list of the tasks included in the currently task model, from where they can interactively select the task to refer within a conversation. In addition, if the user types some letters of the task name the list of the task names is automatically filtered accordingly. After the message is sent by a user, it is added to the chat area of all users. If a chat message contains a task identifier this is shown as a link, which can be selected in order to place the corresponding task in the centre of the working area.

Within the application we also provided users with a **voting system**, which can be useful to make a decision when there are different views on how to address a specific aspect of a task model. It allows a user to propose a topic for the vote, which is shown to all users who can express their agreement or not, and finally shows the result. If the topic concerns a task, selecting the task name in the topic description makes it possible to centre the model in the personal view around such task. It is worth noting that in the evaluations reported in this article the voting system was not used because just two users were considered for each test session.

### 4.5 Cloud Support for sharing task models

Users who collaboratively edit a task model may be located in various places and using different devices, thus we decided making the models shared in a collaborative session available in the cloud. In particular, all the users have a private repository and also have access to a shared repository in which the models collaboratively shared by users are saved. It is worth noting that, since the task models created through Collaborative CTT and those created through Desktop CTTE share the same XSD schema

describing the underlying language, users can indifferently use Collaborative CTT and desktop CTTE for accessing the models contained in their spaces.

### 4.6 Implementation

From the implementation point of view Collaborative CTT has been obtained by applying the Model View Controller (MVC) pattern. The model is the task model description stored in the server-side. Each user request that implies some modification in the task model is sent to the controller (server-side), which manipulates the model and sends back the response to all involved clients that update their view accordingly. All the communication involving the collaborative functions exploits Web socket mechanisms that allow pushing information from server to clients avoiding polling. When a user accesses Collaborative CTT, a Web socket connection to the server is opened and the client subscribes to receiving updates about collaborative functions (such as shared focus, current users focus, temporal operator update, add/edit/delete a task, chat messages, propose or vote a topic) specifying the corresponding callback function that will handle the received information.

## 5 USER FEEDBACK

Two formative tests were conducted to collect user feedback on the usability and usefulness of the features provided in the tool, and receive suggestions for improvements. In both cases the tests were carried out in pairs.

In the initial test the two users were in the same room, while in the second the two users were in different rooms. Thus, the first evaluation addressed the same time/same place setting, and represents the first (but also the second) scenario, whereas the second evaluation covered the same time/different place setting (distributed synchronous workgroup scenario*)*. For the initial assessment we deemed it more useful to deal with users in the same room to better control the experiment and more easily monitor the users' behaviour.

The second test was carried out with a version of the tool which had undergone some small refinements as a result of the first formative test in order to improve its usability. In particular, in the second version we reduced the time when a task is actually locked in order to increase the possibility for users to work in parallel, we refined the chat (which was not used much during the first exercise), and we improved the mutual awareness between users by also showing where the local focus of each user was positioned. The purpose was not to provide a formal comparison between the two tests because various conditions changed, but to obtain progressive feedback that has been useful to orient the evolution of the tool.

### 5.1 Participants and Tasks

**Initial test**

Fourteen people (2 females) aged 25 to 47 (M =32.2, SD =6) participated. All had good experience/familiarity with CTT. They were selected by using the professional network of authors, choosing people having familiarity with CTT notation and potentially interested in the tool. In the end, a pool including experts in HCI (e.g. academic researchers) and Computer Science students (with familiarity with CTT) participated in the evaluation exercise. For the test, users were asked to edit a previously created task model, which describes an ATM system in its "current" design. By using the tool, they had to edit this task model in a collaborative manner so that the new model would describe a possible, envisaged, "new" ATM system. The description of the features that the new system should support (and which they had to include in the model), were provided to them through four tasks to carry out.

In particular, users were required to include the specification of the following tasks in the model: i) add the possibility to access the ATM system using additional modalities apart from the current one (which is typically done through inserting a card and then typing in the code), namely: using either fingerprints, a smartphone or a smartwatch; ii) once a user has logged in to the system, the presentation should adapt by means of e.g. enlarging the fonts, improving the contrast, removing elements in the UI; iii) once the user has selected withdrawal, the system should calculate the amount that the user typically gets and then suggests it to the user, who can accept it or not; iv) the possibility to visualise the current user balance and see the transactions made during a certain interval of time (the user would have to choose a timeframe from: today/1 week/15 days/1 month). After jointly accomplishing such tasks, users had to independently fill in an online questionnaire.

**Second test**

We were not able to involve four of the 14 people who participated to the first user test, so in the end only 10 users (2 females), aged 25 to 47 participated in the second test (M =33.4, SD =6.2), all having quite a good experience/familiarity with CTT.

For the test we asked the users to edit a task model containing a *partial specification* of the features typically supported by a smartphone (e.g. enter a pin to access, make a call, handle messages). In the test, users were requested to edit the task model so that it will also include additional possibilities according to the following tasks. *Task1*: refine a task named "*Show General Information*" by further showing the time, the battery level and the network connectivity level. *Task2*: edit the "*HandleMessages*" task by modelling the tasks supporting users while they *create a message to send to a contact*. In our case, only two types of messages were considered: SMS and Whatsapp messages. Users had to model the fact that, in both cases the user can use text to create the content of the message. However, in the case of SMS messages, the user can also send, attached to the textual message, *memos*, *contacts*, *calendar events*, and *notes*. In the case of Whatsapp messages, the user can send additional types of files: images, videos, and audio files (in addition to memos, contacts, calendar events, and notes). *Task3:* add the possibility that a telephone call can occur *any time* during

the use of the phone and then interrupt any task the user was currently doing with the smartphone. At the end of the telephone call, the user should be able to continue the interaction suspended previously.

As in the previous test, after jointly accomplishing such tasks, users had to independently fill in an online questionnaire.

### 5.2 Procedure and Design

Before the tests, the users were provided with instructions about how to access the tool, a general textual introduction, and a video showing its main features. In both tests users performed the test in pairs. For the first test they were in the same room, each using a PC, and they were placed in such a way that they could easily talk to each other, but could not see the screen of the other participant. They were allowed to talk and chat freely during the test. For the second test, the two users were in different rooms, still using the same equipment as in the first test (PCs). In both cases, two researchers observed the interactions occurring during the experiment.

One of the users initially acted as the moderator, inviting the other user to join the collaborative session: in this condition the two users completed the first two tasks, and then they swapped their roles. This was done in order to have both users act in both roles and test the corresponding functionalities.

After the test, the users filled in a questionnaire, which included first a demographic section (about e.g. education, experience/familiarity with task modelling), and then a section with questions specifically related to the tool.

### 5.3 Results

In the questionnaire, a 5-point scale was used to provide ratings on the tool features: [1 to 5; 1 = not usable at all/not useful at all, 2 = not very usable/not very useful, 3 = neutral, 4 = usable/useful, 5 = very usable/very useful]. We report the median and Interquartile Range (IQR) values.

**Setting shared focus.** *First test.* Usability [Median=4; IQR=5-3.25=1.75] Usefulness [Median=4.5; IQR=5-4=1]

Many users found this mechanism useful (one user even suggested extending it to temporal operators) for better turning/pointing the team's attention toward a specific task-related issue/discussion, and especially useful to quickly focus on a task when dealing with large model specifications. However, from the usability point of view, one user found the provided mechanism difficult to understand since it requires two actions (clicking on the task and then select the button for setting a shared focus). Another user suggested making the visualisation of the shared focus different from the user's own focus (although each user has only one focus at any given time), to better distinguish them. There was only one user who explicitly criticised having his

current focus changed by others: instead, he would have preferred to see where the other users currently had the focus and then decide to change his own focus accordingly. The second version of the tool addressed this issue to some extent by providing the possibility to show also the local focus of the other users.

*Second test.* Usability [Median=4; IQR=4.75-4=0.75] Usefulness [Median=4.5; IQR=5-4 =1]

A user said "*sometimes I forgot that the other user had set the shared focus, thus I made modifications to a wrong subtree*." Another user said that he would have liked to use the mouse right click to access the button to activate the shared focus instead of using the menu in the top-right part of the application. Regarding the usefulness of this functionality, one user suggested further testing this functionality with more than two users. Another user had qualms about the fact that when using this functionality the overall interaction would slow down a bit.

**User authorisations.** *First test.* Usability [Median=4; IQR= 5-4=1 ] Usefulness [Median=4.5; IQR=5-4 =1 ]

Two users would have preferred a different, more compact layout for their settings (e.g. one row per user, using checkboxes or toggle switches).

*Second test.* Usability [Median=4; IQR=4-4=0] Usefulness [Median=4.5; IQR=5-4 =1]

Nothing was particularly noted apart the fact that, in line with what had already been highlighted in the previous test, a user suggested having a more compact layout for visualising users' access rights (he suggested using accordion menus).

**Mutual awareness mechanisms.** *First test.* Usability [Median=4; IQR= 5-4=1 ] Usefulness [Median=4; IQR=5-4 =1]

Users were asked whether it was easy for them to be aware of other people participating in the same session and their current activities (e.g. understand when another user joins a collaborative session, or be informed of the actions that other users are doing/have done on the shared model). Overall, users expressed high appreciation of the usefulness of the support provided by the tool allowing them to be aware of other users' activities. Nonetheless, three users recommended some further improvements to its usability, with different suggestions: one proposal was to associate a colour to each participant to more easily identify users in the same session (and also the user who currently acts as the moderator), and/or to identify the current users by changing different portions of the task model; another user suggested using a short sound to signal when a new user joins a session; another user suggested using a small square around the graphical task representation and then identify the users who are currently focusing on that task by displaying their names (or initials) beside the square. Some of these aspects were addressed in the second version.

*Second test.* Usability [Median=4.5; IQR=5-4=1] Usefulness [Median=5; IQR=5-5 =0]

In the second test one user expressed concerns over the possibility that using the users' initials could cause conflicts, and so suggested using icons rather than initials. Another user raised the issue that it is difficult to know who the users currently connected in the session at any given time are. Another user said: "*As 'Owner' of the task, I received an overwhelming amount of notifications of task modifications, which interrupted my work several times. I suggest collecting all the notifications into a side box, in order to not block the owner's work.*"

**Chat.** *First test.* Not evaluated in the first test because users were in the same room.

*Second test.* Usability [Median=4.5; IQR=5-4=1] Usefulness [Median=5; IQR=5-5 =0] One user raised an issue connected with the fact that it was difficult for the moderator to discuss a modification to the model suggested by another user before accepting/rejecting it. In addition, the same user said *"When the chat window is minimized, every time I receive a new message/ information about a new event, I must maximize it in order to read the message/event notification. I suggest that you write (the first part of) the event notification / text message in the window header. In this way, while the chat window is minimized, I can get an idea of the event notification / message"* Other two users also highlighted the importance of better drawing the user's attention to the most recent message (e.g. by blinking for a few seconds). A user suggested having the possibility to have a voice chat for more easily communicating with the other users.

**Visualisation of logged events.** *First test.* [Median=4; IQR=5-4 =1 ] Usefulness [Median=4; IQR=4-4 =0 ]. This feature received quite mixed comments. On the one hand, one user found it very useful and reported looking more often at the area dedicated to event logging than the one showing the model. Nevertheless, the user suggested better structuring the visualization of the logs, by indicating, for example, first the type of event and the author, in order to speed up the extraction of relevant information. On the other hand, a pair of users said that they did not look much at this panel, while one highlighted the usefulness of this feature especially for remote users. Indeed, users often talked to each other, not only to identify a shared strategy for editing the task model, but also to request confirmation of actions made through the tool (instead of just checking the event log). Another user suggested rendering just the editing events in the panel (e.g. not providing information on the locking events), since they are the really meaningful ones from the user's perspective. Another user suggested adding the possibility to go through past events and even 're-play' them.

*Second test.* Usability [Median=4; IQR=4-3 =1 ] Usefulness [Median=4; IQR=4.75-3 =1.75]

Two users suggested hiding it by default and having the possibility to show it on request. Another user said that he noticed some changes sometimes but then he pre-

ferred looking at the model to understand what happened. Another user suggested classifying the events, by distinguishing between events occurring on the task model and other types of events (e.g. chat modifications, notifications about user joining the session, etc.)

**Coordination (lock mechanism).** *First test.* Usability [Median=4.5; IQR=5-3=2] Usefulness [Median=5; IQR=5-5=0]. Users really appreciated the availability of the locking mechanism to avoid including inconsistencies in the model due to concurrent and uncontrolled modifications. However, some users highlighted that the lock mechanism can slow down the collaborative process excessively, suggesting keeping it only for the time that it is strictly necessary (e.g. when the user actually starts modifying some property of the model, and releasing it just afterwards).

*Second test.* Usability [Median=4; IQR=5-4=1] Usefulness [Median=5; IQR=5-4 =1] A user complained that, as the moderator of the session, he received many notifications about task changes, which made it difficult to work on the model properly: "*As owner of the task, I received many notifications of task modifications, which interrupted my work several times. I suggest collecting all the notifications into a side box, in order to not block the owner's work*." ." Another user said that the locking mechanism could be difficult to handle, he suggested better using the chat for coordinating the work.

**Rejection/Acceptance.** *First test.* Usefulness [Median=4; IQR=4.75-4 =0.75 ]

On the one hand, users acknowledged the need and the importance of providing the moderator with the possibility to act as "super-user" to decide on the modifications to actually apply to the model (among the ones proposed by other users), and then maintaining the control of it. Nonetheless, two participants suggested providing the moderator with some means for justifying rejection of a proposal made by another user (e.g. by means of adding a text field where the moderator can explain the reasons for rejecting a change), so that all members can develop and keep a shared mutual knowledge/view of the correctness of the specification (documented in the model) and its rationale and evolution. On the other hand, confirming every step done by the other partners was judged a bit tiring from the moderator's point of view (a user admitted sometimes having lost his own focus to check requests of change from the other user).

*Second test.* Usefulness [Median=4; IQR=4.75-4=0.75]

Two users acknowledged its usefulness but at the same time they highlighted that the moderator frequently had to interrupt his work to deal with accept/reject requests. Another user pointed out the fact that when a request arrives, the user cannot discuss it with the partner but just accept/reject it.

**Most usable functionality and Least usable functionality.** *First test.* The functionalities that were most appreciated from a usability point of view were the shared focus (seen as a way to have a better "organised" collaborative session), and the possibility

to concurrently modify a model. Among the least usable functionalities, users reported the locking mechanism ( which could slow down the collaborative editing), and the event log list (not particularly structured and currently including events not very meaningful from the user's perspective).

*Second test.* Four users particularly appreciated the chat (which was improved), one user most liked the fact that the task model portions been edited by other users are highlighted graphically. Regarding the least useful functionalities, one user mentioned the logger, and two mentioned the locking mechanism.

**Most useful scenario(s) of use.** *First test.* For assessing this aspect we envisioned four basic scenarios of use (corresponding to those introduced in Section 3) and we asked users to select the scenario(s) (one or more than one) they found most suitable for exploiting the features of Collaborative CTT. The usage scenarios which received the highest approval were: distributed workgroup (selected 10 times) and workgroups aiming at designing an application (10 times as well). The educational scenario was also rated highly (9 times). The scenario that was judged the least useful was the different places/different times scenario (2 times). In any case, the tool was judged by users as highly flexible in supporting rather different scenarios. From users' comments it seems to offer the best opportunities when synchronous (same time) scenarios are to be supported. An advantage highlighted by users is the fact that, by using the tool, users do not need to exchange task model specifications. The educational setting was also judged appropriate for using the tool because in such settings the tool is able to support a good interaction between the teacher and the students while facilitating the work of both. In other words, Collaborative CTT facilitates teachers explaining task models (by using e.g. shared focus functionality and being a Web-based tool) and at the same time it makes possible an active and collaborative participation of students in building task models, giving them the opportunity to put in practice and apply the theoretical knowledge gained in concrete examples.

*Second test.* One user said that the tool can be fruitfully used in all the four mentioned settings. However, for a future version of the tool he suggested improving that the mechanism used by the moderator to accept/reject the suggested modifications because it is time consuming (and thus he has less time available for working on the model). One user declared that the application should fit all the target scenarios, especially the "same time/different places" one. Another user declared that the Educational use fits particularly well. However, also other settings are suitable, but in these cases there should not be anyone needed to confirm/reject the changes of other members.

**Further suggestions.** *First test.* One user suggested adding a non-transparent background when the circular menu for task editing appears, in order to avoid visualisation problems between the circular menu and the task model visualised underneath. Additional suggestions included adding a voice chat in the system and using sounds for notifying important events.

*Second test.* A user suggested removing the locking mechanism and increasing mutual awareness through user icons; another user suggested adding the possibility to edit tasks with drag-and-drop; another user suggested enabling right-click when possible.

### 5.4 Discussion

*First test.* Overall, the results of the test show that Collaborative CTT was appreciated although some aspects (e.g. the lock mechanism and the limited level of mutual awareness) should be subjected to further refinement. Participants especially liked the flexibility provided by the tool in supporting different types of scenarios of use, the most promising ones being when users exploit the tool in a synchronous manner. Another aspect that users particularly liked was the possibility to work (through a Web-based tool) on the same shared model in a concurrent yet organised/controlled manner. In this way the possibility of reworking and duplication as well as the need of exchanging models between members should be reduced, thereby leading to faster and more productive task modelling sessions. As evaluators, we noticed low parallelism between users (i.e. when one user started to edit one task the other rarely started editing another task). However, this can be explained by the users' low familiarity with the tool, and the fact that they tended to follow the sequence of test tasks strictly. Participants verbally discussed the strategy to follow to build the task model for satisfying the test requirements and, being in the same room, they did not use the chat much. Nonetheless, they fruitfully used other tool features (e.g. shared focus) to coordinate their activities during actual editing.

*Second test.* The researchers noticed increased parallelism among participants: in all the tests users started to work on different tasks from the beginning and then they used the tool features to coordinate/verify their work for finally satisfying the test requirements. This enabled us to test the appropriateness of the tool in situations where users actually edit different parts of the model in parallel. This improved parallelism was probably due to two factors. The first one could be greater users' familiarity with the tool: users felt more confident controlling the tool and its features, and therefore exploited it in a more flexible manner. Another possible explanation could be that the remote chat-based communication used in the second experiment was slower than the direct communication used in the first test, therefore users were further stimulated to more efficiently and concurrently edit the model to save time. Nonetheless, the comments received suggest further refining the mechanism supporting the modifications made to the shared task models, which currently may overload the work of the moderator, especially when many requests for modifications have to be analysed in a short time.

## 6 CONCLUSIONS

Currently, most of the automatic environments enabling task modelling only support single users, thus limiting the possible interactions and discussions amongst them. In this paper we present a new Web-based multi-user tool for specifying task models. Among its features, the tool includes relevant mechanisms supporting coordination, communication and mutual awareness between participants. In the paper we discuss the aspects we have addressed in designing the collaborative features in a task modelling tool, what type of mechanisms have been developed for their support, and also report on two formative user tests which provided promising feedback, also identifying aspects that could be subject to further refinement. A video showing the tool is available at https://www.youtube.com/watch?v=AapwdNIz5V8&feature=em-share_video_user.

Future work will be dedicated to further empirical testing in both educational and industrial projects.

## 7 REFERENCES


1. Anzalone, D., Manca, M., Paternò, F., Santoro, C., Responsive Task Modelling, Proceedings EICS 2015, ACM Press, pp. 126-131
2. Duque, R., Gallardo, J., Bravo, C., Mendes, A.J.: Defining Tasks, Domains and Conversational Acts in CSCW Systems: the SPACE-DESIGN Case Study. J. UCS 14(9): 1463-1479 (2008)
3. Forbrig, P., Dittmar, A., Brüning, J., and Wurdel, M., Making Task Modeling Suitable for Stakeholder-Driven Workflow Specifications. In C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2011, LNCS 6765, pp. 51–60, 2011. © Springer-Verlag Berlin Heidelberg 2011
4. Gallardo, J., Molina, A.I., Bravo, C., Redondo, M.A.: Collaborative Modelling of Tasks with CTT: Tools and a Study. CADUI 2008: 245-250
5. Gallardo J., Molina A., Bravo C. and Gallego F.. A System for Collaborative Building of Use Case Models: Communication Analysis and Experiences - Experiences of Use and Lessons Learned from the Use of the SPACE-DESIGN Tool in the Domain of Use Case Diagrams. In Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE-2014), pages 59-68.
6. Genaro Motti, V., Raggett, D., Van Cauwelaert, S., Vanderdonckt, J.,: Simplifying the development of cross-platform web user interfaces by collaborative model-based design. SIGDOC 2013: 55-64
7. Genaro Motti, V., Raggett, D., Quill: a collaborative design assistant for cross platform web application user interfaces. WWW (Companion Volume) 2013: 3-6
8. Guerrero-Garcia, J., Gonzalez-Calleros, J., Vanderdonckt, J., Comparative analysis of task model notations Vol. 22 (NE-1), ENC March 2012, pag. 90-97
9. Giraldo, W.J., Molina, A., Ortega, M., and Collazos, C.A., Integrating Groupware Notations with UML. P. Forbrig and F. Paternò (Eds.): HCSE/TAMODIA 2008, LNCS 5247, pp. 142–149, IFIP, 2008.

10. Gutwin, C., Penner, R., Schneider, K.A.: Group awareness in distributed software development. CSCW 2004: 72-81
11. Hili, N., Laurillau, Y., Dupuy-Chessa, S., Calvary, G.,: Innovative key features for mastering model complexity: flexilab, a multimodel editor illustrated on task modeling. EICS 2015: 234-237
12. Horridge, M., Tudorache, T., Nyulas, C., Vendetti, J., Fridman Noy, N., Musen, M.A.,: WebProtégé: a collaborative Web-based platform for editing biomedical ontologies. Bioinformatics 30(16): 2384-2385 (2014).
13. Jourde, F., Laurillau, Y., and Nigay, L., COMM Notation for Specifying Collaborative and MultiModal Interactive Systems. EICS'10 © ACM, 2010.
14. Martinie, C., Barboni, E., Navarre, D., Palanque, P.A., Fahssi, R., Poupart, E., Cubero-Castan, E.,: Multi-models-based engineering of collaborative systems: application to collision avoidance operations for spacecraft. EICS 2014: 85-94
15. Molina, A.J., Redondo, M.A., Ortega, M.,: A methodological approach for user interface development of collaborative applications: A case study. Sci. Comput. Program. 74(9): 754-776 (2009)
16. Molina, A.I., Redondo, M.A., Ortega, M.: A conceptual and methodological framework for modeling interactive groupware applications. In: Dimitriadis, Y.A., Zigurs, I., GómezSánchez, E. (eds.) CRIWG 2006. LNCS, vol. 4154. Springer, Heidelberg (2006)
17. Mori, G., Paternò, F., Santoro, C., Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions, IEEE Transactions on Software Engineering, August 2004, Vol.30, N.8, pp.507-520, IEEE Press.
18. Noy, N.F., Chugh, A., Alani, H.,: The CKC Challenge: Exploring Tools for Collaborative Knowledge Construction. IEEE Intelligent Systems 23(1): 64-68 (2008)
19. Paternò, F., Model-based Design and Evaluation of Interactive Applications, Springer Verlag, ISBN 1-85233-155-0, 1999.
20. Paternò F., Santoro C., Spano L.D., Engineering the authoring of usable service front ends. The Journal of Systems and Software. Elsevier, Volume 84, Issue 10, October 2011, pp. 1806-1822
21. Penichet, V.M.R., Lozano, M., Gallud, J.A., and Tesoriero, R., Task Modelling for Collaborative Systems. M. Winckler, H. Johnson, and P. Palanque (Eds.): TAMODIA 2007, LNCS 4849, pp. 287 – 292, 2007. Springer-Verlag Berlin Heidelberg 2007
22. Pinelle, D., Gutwin, C., Greenberg, S., 2003, Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. ACM Transactions on Computer-Human Interaction, Vol. 10, No. 4, December 2003, pp. 281–311.
23. Renger, M., Kolfschoten, G.L., de Vreede, G.-J., 2008: Challenges in Collaborative Modeling: A Literature Review. Chapter in Book Advances in Enterprise Engineering I, Vol. 10, Lecture Notes in Business Information Processing, pp. 61-77.
24. Rittgen, P., Group Consensus in Business Process Modeling: A Measure and Its Application. IJeC 9(4): 17-31 (2013)
25. Van der Veer, G., Kulyk, O., Vyas, D., Kubbe, O., Ebert, A., Task modeling for collaborative authoring. ECCE 2011: 171-178.
26. Wurdel, M., Sinnig, D., and Forbrig, P., Toward a Formal Task-Based Specification Framework for Collaborative Environments, V. López-Jaquero et al. (eds.), Computer-Aided Design of User Interfaces VI, 221. Springer-Verlag London Limited 2009, Chapter 20, pp. 221-232.