HAL
open science

COMM Notation for Specifying Collaborative and MultiModal Interactive Systems

Frédéric Jourde, Yann Laurillau, Laurence Nigay

To cite this version:

Frédéric Jourde, Yann Laurillau, Laurence Nigay. COMM Notation for Specifying Collaborative and MultiModal Interactive Systems. Proceedings of the second ACM SIGCHI Symposium on Engineering Interactive Computing Systems (Berlin, Germany - June 19-23, 2010) (EICS 2010), 2010, Berlin, Germany. pp.125-134, 10.1145/1822018.1822038 . hal-00953315

HAL Id: hal-00953315

https://inria.hal.science/hal-00953315v1

Submitted on 28 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

COMM Notation for Specifying Collaborative and MultiModal Interactive Systems

Frédéric Jourde, Yann Laurillau, Laurence Nigay
University of Grenoble, CNRS, LIG
Domaine Universitaire
38000 Grenoble Cedex 9, France
{frederic.jourde, yann.laurillau, laurence.nigay}@imag.fr

ABSTRACT
Multi-user multimodal interactive systems involve multiple users that can use multiple interaction modalities. Although multi-user multimodal systems are becoming more prevalent (especially multimodal systems involving multitouch surfaces), their design is still ad-hoc without properly keeping track of the design process. Addressing this issue of lack of design tools for multi-user multimodal systems, we present the COMM (Collaborative and MultiModal) notation and its on-line editor for specifying multi-user multimodal interactive systems. Extending the CTT notation, the salient features of the COMM notation include the concepts of interactive role and modal task as well as a refinement of the temporal operators applied to tasks using the Allen relationships. A multimodal military command post for the control of unmanned aerial vehicles (UAV) by two operators is used to illustrate the discussion.

Author Keywords
Groupware, multimodal interaction, specification notation.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported supported cooperative work. H.5.2. User Interfaces: Input devices and strategies. D.2.2. Design Tools and Techniques: User interfaces.

General Terms
Human Factors; Design; Algorithms.

INTRODUCTION
We present the COMM (Collaborative and MultiModal) notation dedicated to the specification of multi-user multimodal interactive systems. An interactive system is multimodal if users can interact with the system using at least two (input or output) interaction modalities either operating in parallel or not. Several multi-user interactive systems (e.g. groupware) are multimodal such as the interactive systems described in [3, 4, 6, 12, 25, 35]. Furthermore, there is a growing interest in the CSCW community for multi-user multimodal interactive systems based on interactive multitouch surfaces [19, 22, 28, 30, 31]. Although multi-user multimodal systems are becoming more prevalent, their design is still ad-hoc without properly keeping track of the design process. The design of such interactive systems gives rise to new challenges due to the multiplicity of users with roles as well as the availability of different modalities and combinations of them. Addressing this issue of a lack of design tools for multi-user multimodal systems, we present the COMM notation and its editor for specifying interactive systems that are both multi-user and multimodal.

The specification step is one of the fundamental steps in any software development process regarding the waterfall [23] or the V lifecycle models [14]. It is essential to have specification tools that help the designer to precisely analyze and describe the user interface and to specify the interaction process [9]. A specification should also help the designer to detect usability problems at an early stage before implementing the interactive systems.

For specifying multi-user multimodal systems, we adopt a task-based notation. As pointed out in [33], task analysis is used in various steps of the development process and we focus here on the specification of a system to be then developed. In a previous study [10], we have reviewed several task-based notations, such as CTT [17], GTA [32, 33], MABTA [13], CUA [21] or UML-G [24] and we have highlighted the lack of support to describe the multimodal aspects of a multi-user interface. Addressing this issue, we present a notation, named COMM, that enables the designer to specify collaborative and multimodal aspects of an interactive system. Rather than creating a brand new notation from scratch, using a completely new graphical language to be learned, the COMM notation reuses and coherently articulates concepts of existing notations, such as CTT [17]. To describe the multimodal aspects of the interaction, we base the notation on the OpenInterface conceptual model [27] for multimodality (definition of a modality [18] and temporal constraints [34]).

We identify two requirements that the notation must satisfy: (1) to describe at a finer grain the group activities in order to clearly identify the cooperative and the

EICS'10 © ACM, 2010. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems.
http://doi.acm.org/10.1145/1822018.1822038

collaborative activities; (2) to explicitly describe the link between the abstract User Interface (UI) specified as group tasks and the concrete UI involving multiple interaction modalities. In order to satisfy the first requirement, we introduce the concept of **interactive role**, in contrast to the concept of role usually used in the CSCW literature, that we will denote in this paper as business role[1] for clarity. To satisfy the second requirement: first, we introduce the concept of **modal task,** which denotes an atomic user action performed on a physical device; second, we refine CTT temporal operators using the temporal relationships between modalities defined in [34] and based on Allen relationships [2].

Figure 1: Two military operators controlling UAVs.

The paper is organized as follows: we first present a state of the art of the existing notations and we focus on their ability to specify concrete user interaction. We then describe the main elements of the COMM notation before presenting the e-COMM editor, an on-line application for specifying interactive systems using the COMM notation. We illustrate our notation by considering the design of a multi-user multimodal military command post. Indeed the COMM notation is currently used in the context of a French Army Research Department project, in partnership with the firms BERTIN, SAGEM, EADS and PY Automation, to design military command posts for the control of two unmanned aerial vehicles (UAV) by two operators, as shown in Figure 1.

RELATED WORK

"*Task analysis is the process of understanding the user's task thoroughly enough to help design a computer system that will effectively support users in doing the task.*" [11]. Beyond analyzing current or envisioning user's tasks, task-based notations are also commonly used to design and specify the user interface (UI) [33]. Finally some task-based specifications are also used as inputs for UI generation. In our study, we focus on task-based specification of the user interface and we do not cover the UI generation aspects. Therefore we do not consider UML-based notations, such as UML-G [24], which are best suited for software implementation and UI generation or transformation.

Several task-based notations have inspired our work. We reviewed the ones dedicated to the specification of multi-user interactive systems. In the field of group work task analysis, there exist several notations dedicated to the description of group activities such as GTA [32], MAD [26], MABTA [13] or CTT [17]. Some of these notations, mostly CTT and CTT-based notations such as CIAN [15], TaskMODL [29] or UserView [20], are used to design and specify the user interface. We analyse these notations in light of our two requirements exposed in the introduction section: capacity to support the specification of (1) cooperative and collaborative activities (2) multimodal interaction.

First the notation should support the specification of both kinds of group work task: collaborative and cooperative. Dillenbourg et al. [5] define the two types of tasks as follows:

> *"Collaboration" is distinguished from "cooperation" in that cooperative work "... is accomplished by the division of labor among participants, as an activity where each person is responsible for a portion of the problem solving...", whereas collaboration involves the "... mutual engagement of participants in a coordinated effort to solve the problem together."*

For multi-user multimodal interaction, this requirement is important because cooperative versus collaborative tasks lead to design of different concrete multimodal UIs. Indeed, cooperative tasks imply a predefined task allocation to business roles (division of labor). At the concrete UI level, it implies that, for each business role, the designer assigns a predefined set of modalities and therefore decides at design time which interaction modalities should be used. In contrast, for collaborative activities, task allocation is performed at runtime. Consequently, when the designer specifies the concrete UI, s/he cannot assign modalities to each business role. Thus, GTA, MAD, MABTA, CTT support only cooperative tasks. Furthermore, considering CTT, a group work task is described with a cooperative task tree and a set of individual task trees for each involved business role. Only CIAN and CUA explicitly identify the two types of tasks, cooperative and collaborative, in addition to the existing types of tasks: user, system, interactive. Nevertheless CIAN and CUA focus on abstract interaction only.

The second requirement is to support the description of concrete multimodal interaction. Existing notations mainly focus on abstract UI: As stated in [16], the notations in particular any CTT-based notations such as CIAN [15], enable the specification of abstract UI and may serve as input for designing a concrete/final UI. Nevertheless we note that GTA is based on (N)UAN [8] which is well suited

[1] Business role: Representing role in the organization

to specify concrete interaction in a textual/tabular form while MABTA makes references to a graphical sketch of the UI.

A notation should support a seamless transition from the specification of the abstract UI to the specification of the concrete multimodal UI and vice-versa. Moreover multimodal interaction requires the specification of temporal constraints between modalities [34]. Temporal operators such as the ones in CTT-based notations (e.g., sequence, parallelism, interleaving) cannot express such qualitative temporal constraints at a fine grain. For example, we consider a multimodal interaction for navigating in a map; a user may say ”here”, using voice recognition, while pointing to the map. The temporal constraints can be defined by the designer in several ways: For instance (1) the user must first speak and, when finished, point to the map; or (2) the user must start to point to the map and, after say “here”, and then finally release her/his finger. For the first example, modalities are used in a sequential way. For the second example, based on Allen relationships [2] there is a temporal coincidence between modalities as defined in [34]: Indeed the speech modality is used in the context of the direct-manipulation modality. This is a particular case of parallelism, not covered by CTT operators.

Motivated by the fact that existing notations only partly satisfy the two above requirements for specifying multi-user multimodal interaction, we define the COMM (Collaborative and MultiModal) notation, an extension of the CTT notation. We chose to reuse CTT because it is a popular and widely used notation for UI specification. Therefore usual concepts, such as business role, object, agent, task or event, are included in the COMM notation. In the following sections, we first describe the key features of the COMM notation before presenting the e-COMM editor that supports the COMM notation.

COMM NOTATION

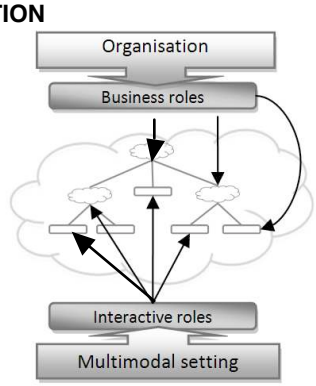

Figure 2: Business Role / Interactive Role.

The COMM notation extends CTT by introducing two concepts, namely **interactive role** and **modal task,** and by considering the five Allen relationships as temporal operators applied to concrete tasks.

- An **interactive role** differs from the usual concept of (business) role in the CSCW literature. As schematized in Figure 2, an interactive role identifies users that interact with the system using modalities, while a business role denotes a role in an organisation such as a technician, an operator, etc.
- A **modal task** is an additional type of task corresponding to the concrete interaction level: A modal task denotes an atomic user action performed on a physical device.
- Allen relationships are used to refine existing temporal operators applied to tasks in order to specify multimodal concrete interaction. Since it is a refinement of CTT temporal operators for abstract interaction, COMM supports a seamless transition from abstract UI to concrete multimodal UI and vice-versa.

We detail these features of the COMM notation by considering an existing multi-user multimodal interactive system presented in [30] as an illustrative example: co-located players are involved in a team play by interacting with a multitouch table and by issuing voice commands. The game is to play Warcraft: two players cooperate to control troop units in order to win the game.

Cooperation vs collaboration

Let us first consider the following cooperative multimodal task implemented in the Warcraft multi-user system [30]: moving a group of troop units. This is a cooperative task since two business roles are clearly identified: the first business role, named *sergeant*, has the responsibility of grouping troop units while the second business role, named *chief*, has to move the group to the newly specified location. As shown in Figure 3, this cooperative task is represented with CTT by two individual interactive task trees, one per business role, and one cooperative task tree. Using COMM, as shown in Figure 4(a), the tasks are represented using a unique task tree: a label decorates the root node to indicate which business roles are involved (i.e., Chief, Sergeant). Similarly, the interactive task is decorated with labels that indicate which business role (i.e., Chief or Sergeant) must perform the concrete task.

Let us now envision the following collaborative task, based on an existing task available in Warcraft, and inspired from [30]: create a unit such as a knight. For performing this task, the first user selects the mode [barrack] (i.e., a building dedicated to the creation of units) and then selects the kind of units to be created; the second user has to confirm the creation of the unit. Such a task cannot be described with CTT since it only supports the description of cooperative tasks. Indeed specifying the constraint that at least two users involved in this collaborative task must interact with different modalities will be complex using existing notations. This will lead to a non-compact and therefore not easily readable specification.

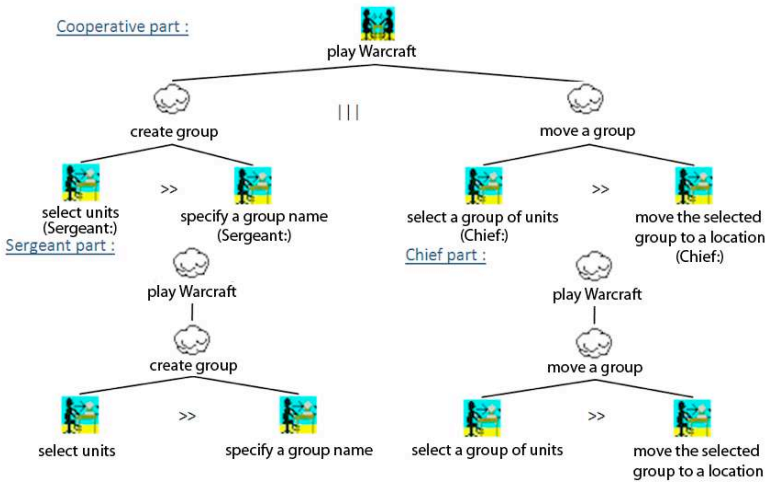

Figure 3: CTT cooperative task.

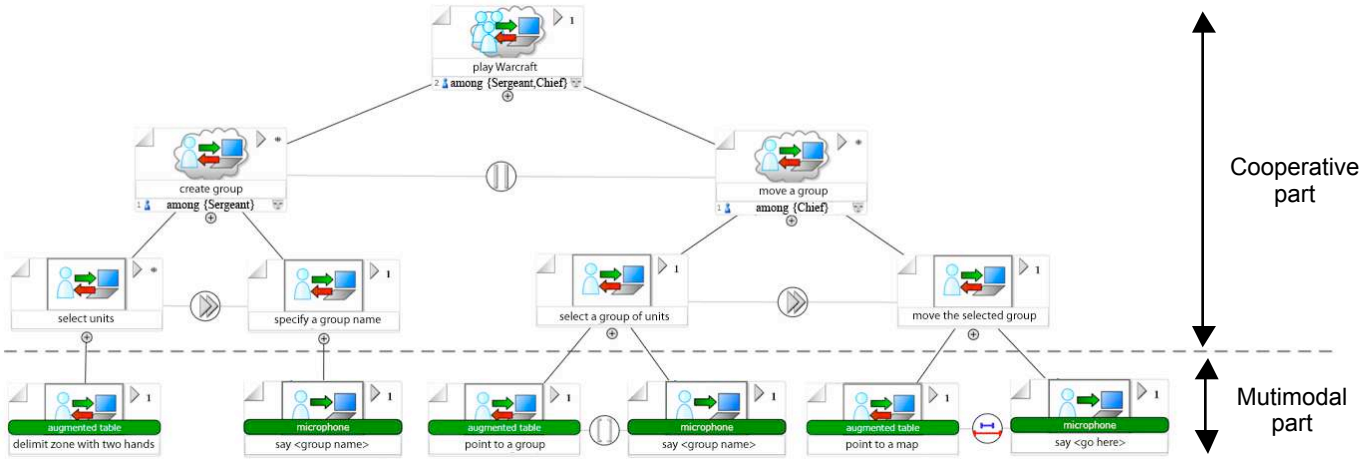

Figure 4 : (a) COMM cooperative task.

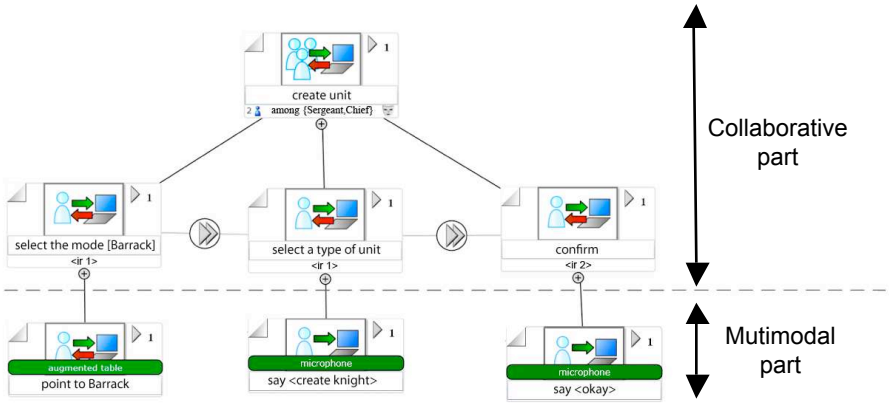

Figure 4: (b) COMM collaborative task.

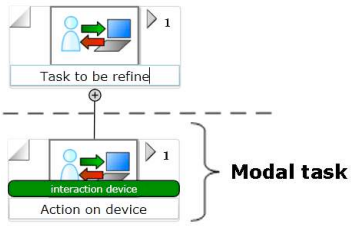

Figure 5: Modal task.

Using COMM, as shown in Figure 4(b), we use the concept of **interactive role** (i.e., <ir1> and <ir2> in Figure 4(b)). An interactive role is associated with each elementary task that composes the unit creation task. We associate the role <ir1> to the two first subtasks and <ir2> to the subtask of confirming the unit creation. This implies that the same role should perform the two first subtasks (specify the unit creation) while the third subtask (confirm the unit creation) should be performed by a different role. Furthermore this specifies that both interactive roles are instantiated at runtime and should be performed by any role (sergeant or chief). To fully describe the collaborative concrete task, the subtasks labelled by interactive roles must then be described in terms of interaction modalities: in the example of Figure 4(b), the two modalities - direct manipulation on the multitouch table and speech commands - for <ir1> and speech commands for <ir2>. For clarity, due to a limited space, we have only represented input modalities.

Interactive roles allow us to specify how tasks are dynamically allocated to business roles and how such tasks can be performed using multiple modalities (i.e., direct manipulation and speech). An interactive role is an ephemeral role played by an actor, dynamically assigned to a business role until the task is completed. As shown in Figure 2, while a business role rules the division of labor, which corresponds to a top-down approach (abstract to concrete), an interactive role is a means to specify how group activities are influenced by user interaction constraints, which corresponds to a bottom-up approach (concrete to abstract). In other words, business and interactive roles make explicit the tension between group work activity requirements and concrete interaction requirements.

Specifying concrete multimodal interaction

In order to fully specify the concrete multimodal interaction, elementary tasks (leaves of the task tree) must be refined in order to describe which modalities are used and how the concrete interaction is performed. To do so, we decompose an elementary task as a composition of **modal tasks** (or monomodal tasks). Based on the definition of an interaction modality as the coupling of a physical device with an interaction language [18], an atomic task is defined as an association of an interactive physical device with an atomic user action that is covered by the interaction language.

In the COMM notation, as shown in Figure 5, a modal task is symbolized by two overlapping rounded rectangles (white and green), each containing a textual label. The first label describes the interaction device and the second one describes the user action (interaction language) performed on the device. For example, the elementary concrete subtask *select the mode [barrack]* of Figure 4(b) is made of the modal task that corresponds to the modality, put her/his finger on the touch table. This elementary concrete subtask is defined by the modal task (<Augmented table, point to barrack>). Another choice of modalities and in particular the choice of devices (for instance a mouse) could lead one to define different modal tasks for each of the subtasks.

A task performed using multiple modalities is described as a combination of modal tasks. This is the case of the task *create unit* in Figure 4(b). When multiple modalities are used at the same time, the temporal constraints between the modalities must be specified. This temporal constraint drives the fusion mechanism in order to detect when fusion must be performed. We consider the task of moving troop units of Figure 4(a). While one user (*sergeant*) can group units by sliding her/his hands on the touch table (left/right), another user (*chief*) can move another group in parallel. To do so the user (*chief*) will select an existing group of units and will then point to the new location on the table while saying <go here>. In the latter case, the temporal constraint that links these two concrete subtasks is specified by a *Coincidence* operator [34] inherited from the Allen relationships [2]: Indeed the speech modality is used in the context of the direct-manipulation modality. Such an operator [34] is used to specify a temporal constraint between modal tasks at a fine grain of description, while we use the parallel operator of CTT to specify temporal relationships between non-modal tasks as in Figure 4(b). We therefore keep CTT operators (i.e., parallelism and sequence) to describe the relationships between tasks at a higher level of abstraction.

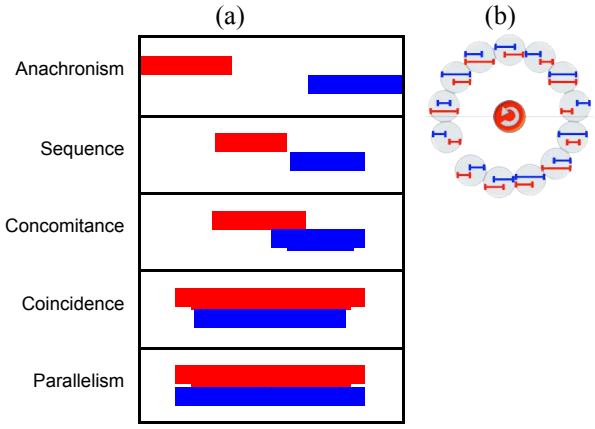

Figure 6: (a) Temporal combinations between modalities. (b) Palette of temporal combinations in the e-COMM editor.

To specify the temporal combinations of modal tasks, we use the five temporal operators [34] described in Figure 6(a). Figure 6(b) shows how the temporal relationships can be specified between two modal tasks using the e-COMM editor. In Figure 6(a), each line corresponds to the time interval in which a modality is used. Sequential and parallel combinations are used in CTT. We add here three new temporal combinations described in [34]. Two modalities are combined anachronously if there is a temporal gap between their usage. Two modalities are concomitant when one modality replaces another one with a time interval during which the two modalities coexist. The coincidence of two modalities is when one modality is only used in the context of another one. This is the case of the example of Figure 4(a), when speech modality is used in the context of

the direct-manipulation modality. The two CTT operators (i.e., parallelism and sequence) are then refined by one of the five temporal relationships defined in [34]. This enables us to ensure continuity between the description of the abstract UI, independent of the modalities and the description of the concrete multimodal UI.

Other elements of the COMM notation

In comparison with CTT, additional features are included in the COMM notation. In particular we consider new types of tasks:

- *User interactive* task (Figure 7-5): A *User interactive* task implies that interaction occurs through a combination of input and output modalities that are described at the level of the **modal tasks** (Figure 7-6). Each user interactive task can be decorated with a role (business role or interactive role). For example in Figure 7-4, the user interactive task is decorated by an interactive role <ir 1> which means that this task can be exclusively performed by the *Pilot* or the *Payload operator* (see our example of UAV command post).
  We also consider the *Action*, *Presentation* and *System* types of tasks. The first one is based on input modalities only without feedback, the second one is based on output modalities only. The last one denotes an internal computation without feedback.
- *Group interactive* task (Figure 7-1): This type of task includes the *Cooperation* task of CTT. But, as explained above, a group task may be cooperative or collaborative. This type of task implies multiple *User interactive* tasks assigned to different roles. A *Group interactive* task is decorated (Figure 7-2) with the list of business roles that could perform the task and with the minimal number of users required to perform the task; this number may be greater than the number of roles: for example, a task may be performed by two pilots and one payload operator (see our example of UAV command post).
- *Group task*: This type of task extends the *User* task of CTT. A *User* task denotes internal and cognitive action by one user in CTT. We generalize it to *Group* task that denotes a decision/discussion by the group without interacting with the system.

Finally, and in addition to the refinement of the CTT temporal operators described previously, we also extend the iteration operator of CTT. As shown in Figure 7-3, the iteration operation, represented by a *, is further detailed in order to specify how the iteration occurs. The iteration operation is preceded by a || or a |> symbol which means that the iteration can be performed in parallel or sequentially.

e-COMM: COMM EDITOR

As highlighted by van der Veer et al [32] and by Molina et al. [16], a notation is fully usable if it exists as a tool for the designer that supports the notation. For instance, we may say that CTTE (CTT editor) has contributed to the success of CTT. Towards this goal, we have developed the e-COMM application that supports the COMM notation. This application is implemented as an **on-line editor** (iihm.imag.fr/demo/ecomm) developed in C# and based on the Microsoft SilverLight technology. The source code will be publicly available and released under the GPL Licence. The e-COMM graphical user interface (Figure 7) emphasizes direct manipulation of the COMM concepts in order to avoid forms, menus or toolbars as much as possible.

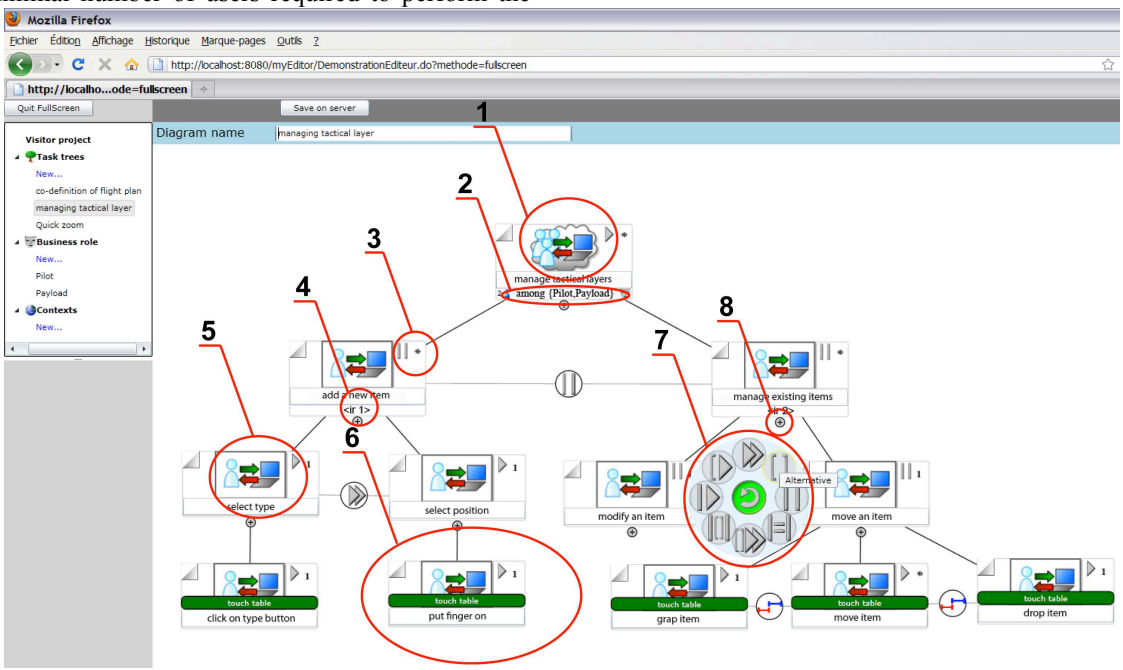

Figure 7: Screenshot of the e-COMM editor.

Indeed, we think that the plethora of filling form popup windows makes a tool less usable and does not encourage focus on the specification under development. Using e-COMM, when a new task tree is created, there is only one node, the root node. Clicking on the (+) symbol (Figure 7-8) leads to the creation of a new subtask. By default, the sequential operator is selected to link the newly created subtasks. The operator can be modified by clicking on the operator symbol and selecting another one: to do when clicking on the operator, the operator wheel (Figure 7-7) appears and shows the available operators. The same interaction technique is used to change the attributes of a task: name, iteration operator and type. Multiple task types are available such as the ones available in CTT and the multiuser task types. Furthermore in order to efficiently navigate in a large task tree the zoom (in/out) action is controlled by the mouse wheel. It is then easy and smooth to focus on a particular part of the tree or to have a global view of the tree. The resulting COMM task tree specification can be saved in different formats including XML, pdf and image formats in order to be easily included in any specification document. In the following section, the trees shown in Figure 8 and 9 have been defined using the e-COMM editor.

CONCRETE EXAMPLE: UAV COMMAND POST

In the context of a French Army Research Department project, in partnership with the firms BERTIN, SAGEM, EADS and PY Automation, the COMM notation is used to specify military command posts for the control of two unmanned aerial vehicles (UAV) by two operators (Figure 1).

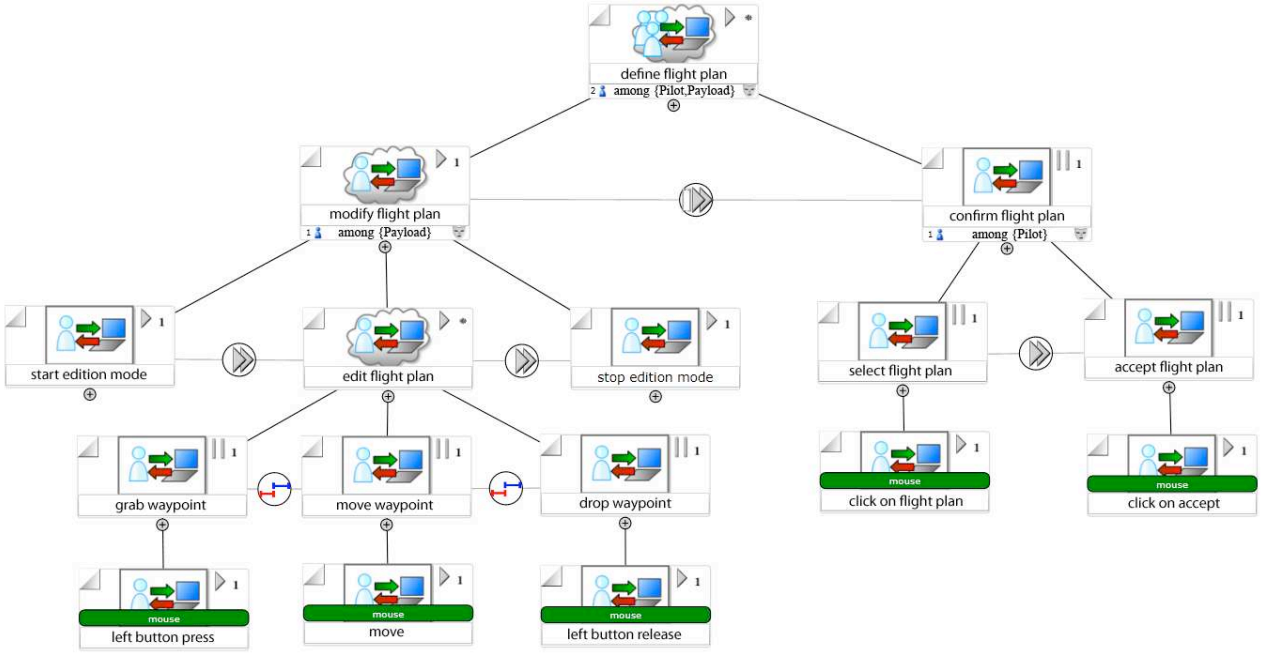

Figure 8: Cooperative task for the modification of a flight plan.

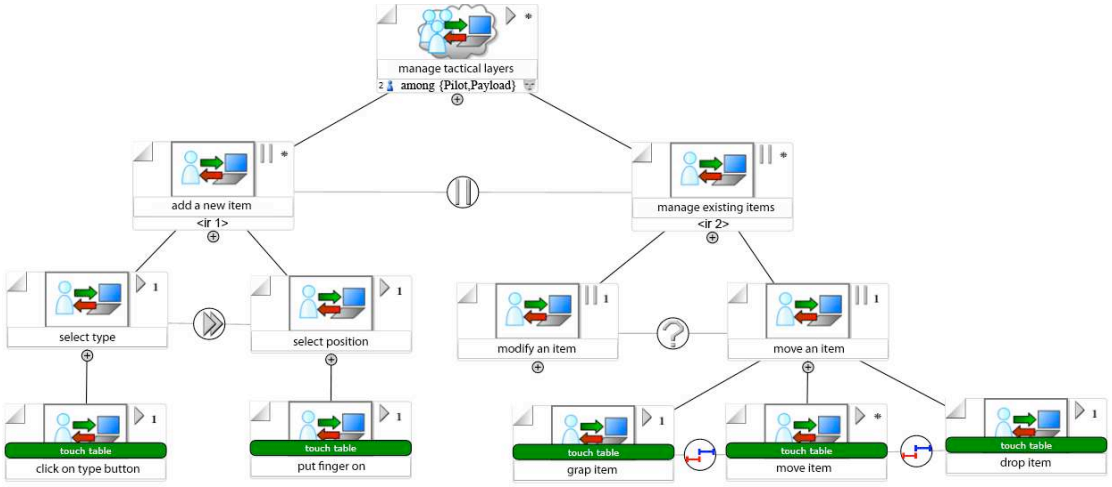

Figure 9: Collaborative task for managing tactical layers.

In this project we adopted an iterative design of the multi-user multimodal interaction to envision the future command posts, and we illustrate the COMM notation by some specifications done for the first iteration using the e-COMM editor. Two operators are involved in the control of UAVs. The first operator has to control the UAV (navigation, guidance, speed, etc). The second one controls embedded devices (payload) such as an infrared camera or a 360º degree camera. In the following, we denote the first operator by the Pilot role and the second one by the Payload operator role. We present three of the specified tasks along with the designed multimodal interaction: (1) to modify a flight plan dynamically and cooperatively; (2) to collaboratively manage tactical layers; (3) to control a multimodal zoom.

Cooperative modification of a flight plan

We consider the first interactive task. During the flight, if required, the flight plan may be modified. Indeed, the Payload operator may observe an event, such as enemy troop movements along the trajectory, detected by the embedded devices, which implies redefining of the flight plan for safety reasons. A flight plan is defined as a set of waypoints linked with straight lines. Then, the Payload operator interactively defines a new trajectory by selecting and moving existing waypoints on a map. When finished, the modified flight plan is submitted to the Pilot. He/she may accept or not the new flight plan. If the new plan is not accepted, the current flight plan is still active and the new plan is cancelled. If the new plan is accepted, the effective UAV trajectory is modified in order to follow the new flight plan. As shown in Figure 8, we describe this interaction by a cooperative task. Indeed, there is a clear separation of work between the two roles. The cooperative task is composed of two individual sub tasks. The first one is performed by the Payload operator: the main interaction is the composition of three modal tasks (selecting a waypoint, moving the waypoint and dropping the waypoint on a map) that must be executed sequentially. The modal tasks are performed using a mouse. When the operator decides to terminate the task, it triggers the individual task *confirm flight plan* with an object as a parameter, an instance of the flight plan class. Then, the Pilot must first select the new flight plan and validate or not the modification.

Collaborative management of tactical layers

For the second interactive task, we consider a set of stacked layers that contains geographical or military information represented on a map such as tanks, buildings or landmarks. For simplicity, we consider that there is only one layer that contains items representing military vehicles or buildings. The content of the layer can be modified by both roles (Payload operator and Pilot). The information contained in these layers are used by the system to compute new trajectories as a function of the environmental context such as enemy vehicle movements. As shown in Figure 9, the two main interactive subtasks are: *add a new item* and *manage an existing item*. Both subtasks may be executed simultaneously by both operators (the task is decorated with the label *among {Pilot, Payload}*) but the execution is exclusive. Indeed, the first subtask must be performed only by the first interactive role `<ir1>` while the second one must be performed by the interactive role `<ir2>`. It means that if the Pilot decides to perform the task *add a new item*, the interactive role `<ir1>` is instantiated with the Pilot role. It also means that if the Payload operator role decides to perform the task *Manage tactical layers*, only the task *manage existing items* is available until the Pilot has terminated her/his task. The corresponding modal tasks are based on finger-based selection of buttons and items displayed on a touch table that offers user identification (e.g., DiamondTouch). In this design, the two users can work in parallel but at a given time, they cannot perform the same task. One user can create a new item by (1) selecting the type of item in a palette and then (2) selecting a location on the map, while the second user is moving an existing item. But the two users cannot work in parallel on the same task: for instance one user cannot select the item to be created in the palette while the other one is selecting the location of the new item on the map. Such a design solution is concisely and precisely specified using COMM by using interactive roles.

Multimodal zoom

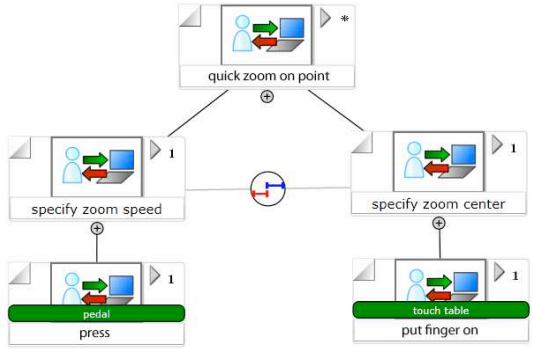

Figure 10: Concrete multimodal zoom task.

The last example focuses on the task of zooming in/out in the map described in the previous section. As shown in Figure 10, the task is specified as a combination of two elementary tasks that involve different modal tasks: one modality is based on a pedal which must be pressed to control the zoom speed, while the other one is based on the touch table to specify the zoom center. The modality based on the pedal is active only if the zoom center is specified, in other words only if the user is touching the touch table using her/his finger. Such a temporal constraint is expressed by a Coincidence [34] (C label) operator. Since no interactive role is specified, the two modalities can be used in parallel by the two operators, one using the pedal while the other one is specified with her/his finger the zoom center.

CONCLUSION AND FUTURE WORK

We have presented the (Collaborative and MultiModal) COMM notation, an extension of CTT for specifying multi-user multimodal interaction. The salient features of the notation are: (1) the concepts of **interactive role** and (2) **modal task**, (3) a refinement of the temporal operators applied to tasks using the Allen relationships. CTT temporal operators are used for abstract tasks and are translated in terms of the five relationships between modalities identified in [34] and based on Allen relationships, when describing temporal constraints between modality dependent tasks, i.e, modal tasks. This introduces a smooth continuity between the specification of the abstract UI and that of the concrete multimodal UI.

Furthermore e-COMM is a fully operational on-line editor that is used in a military project involving industrial partners (BERTIN, EADS, SAGEM and PY automation). The workflow specification is also implemented in the e-COMM editor but not described in this paper.

As future work, we plan to further evaluate the COMM notation and its editor.

Empirical validation, i.e. by experience, is the first type of validation we assessed. The model has been used to specify existing multi-user multimodal systems such as the ones described in [1, 30]. Furthermore we are using the notation for specifying new multi-user multimodal systems as part of the on-going military project. The iterative approach adopted in this project provides a means to compare the result of the design step specified using COMM with the implementation done by another partner, and, then, to evaluate the precision and power of expression of the notation. Moreover in the project, the COMM specifications are resulting from a joint effort from different partners, not only the authors of the notation.

Conceptual validation is the second kind of validation we started to assess. To do so we consider the Notational Dimensions of the Cognitive Dimensions of Notations framework [7]. For instance *low viscosity* (resistance to change in the specification) can be studied when a designer changes a modality. In addition to changing the corresponding modal task, s/he only has to adapt the neighboring modal tasks, only if they are parts of a multimodal task (e.g., changing the temporal constraints). Furthermore interactive roles may be impacted by changing a modality. We plan to further evaluate the COMM notation based on the Cognitive Dimensions of Notations analysis framework and in particular the *Low premature commitment* and the *high visibility* dimensions by conducting experiments with 16 master students (Master Course on Groupware) using the e-COMM editor.

ACKNOWLEDGMENTS

The work presented in the article is partly funded by DGA (French Army Research Dept.) under contract PEA FH/PA. Special thanks to G. Serghiou for reviewing the article.

REFERENCES

1. Ajaj, R., Vernier, F., Jacquemin, C., Navigation Modes For Combined Table/Screen 3D Scene Rendering, in Proceedings of ITS'09 (Banf, Canada, November 2009), ACM Press, 141-148
2. Allen, J. Maintaining Knowledge about Temporal Intervals. *Comm. of the ACM*, 1983, 26(11), 832-843, ACM Press.
3. Brown, B., MacColl, I., Chalmers, M., Galani, A., Randell, C., and Steed, A., Lessons from the lighthouse: collaboration in a shared mixed reality system, in *Proceedings of CHI'03* (Ft. Lauderdale FL, April 2003), ACM Press, 577-584.
4. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. QuickSet: multimodal interaction for distributed applications, in *Proceedings of MULTIMEDIA'97* (Seattle WA, November 1997), ACM Press, 31-40.
5. Dillenbourg, P., Baker, M., Blaye, A., and O'Malley, C. The evolution of research on collaborative learning. *Learning in Humans and Machine: Towards an interdisciplinary learning science*, 1996, 189-211, Elsevier.
6. Etter, R., and Röcker, C., A Tangible User Interface for Multi-User Awareness Systems, in *Proceedings of TEI'07* (Baton Rouge LA, February 2007), ACM Press, 11-12.
7. Green, T., Instructions and descriptions: some cognitive aspects of programming and similar activities, in *Proceedings of AVI'00* (Palermo, Italy, 2000), ACM Press, 21-28.
8. Hix, D., and Hartson, H. *Developing User Interfaces: Ensuring Usability Through Product & Process*, 1998, Wiley & Sons.
9. Jacob, R. J. K., A Specification Language for Direct Manipulation User Interface. ACM Transactions on Graphics, 1986, 5(4), ACM Press, 283-317.
10. Jourde, F., Laurillau, Y., Morán, A., and Nigay, L., Towards Specifying Multimodal Collaborative User Interfaces: A Comparison of Collaboration Notations, in *Proceedings of DSV-IS'08* (Kingston, Canana, July 2008), Springer, 281-286.
11. Kieras, D. Task Analysis and the Design of Functionality. *The Computer Science and Engineering Handbook (2nd Ed)*, 1997, 46-1-25, Chapman & Hall.
12. Kraut, R.E., Fussel, S.R., and Siegel, J. Visual Information as a Conversational Resource in Collaborative Physical Tasks. *Human-Computer Interaction Journal*, 2003, 18(1), 13-49, Lawrence Erlbaum.
13. Lim, Y.K., Task models for groupware and multitasking: Multiple aspect based task analysis (MABTA) for user requirements gathering in highly-

contextualized interactive system design, *in Proceedings of TAMODIA'04* (Prague, Czech Republic, 2004), ACM Press, 7-15.

14. McDermid, J., and Ripkin, K. *Life Cycle Support in the ADA environment*, 1984, Cambridge University Press.
15. Molina, A.I., Redondo, M.A., and Ortega, M., A conceptual and methodological framework for modelling interactive groupware applications, in *Proceedings of CRIWG'06* (Medina del Campo, Spain, September 2006), Springer, 413–420.
16. Molina, A.I., Redondo, M.A., and Ortega, M. A Review of Notations for Conceptual Modeling of Groupware Systems. *New Trends on Human–Computer Interaction*, 2009, 75-86, Springer.
17. Mori, G., Paternó, F.,Santoro, C., CTTE : Support for Developing and Analyzing Task Models for Interactive System Design. *TOSE Journal*, 2002, 28(8), 797-813, IEEE Press.
18. Nigay, L., Coutaz, J., A Generic Platform for Addressing the Multimodal Challenge, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 98-105.
19. Pauchet, A., Colfedy, F., Lefebvre, L., Picard, S., Bouguet, A., Perron, L., Guerin, J., Corvaisier, D., and Collobert, M., Mutual Awareness in Collocated and Distant Collaborative Tasks Using Shared Interfaces, in *Proceedings of INTERACT'07* (Rio de Janeiro, Brazil, September 2007), Springer, 59-73.
20. Penichet, V.M.R., Paternò, F., Gallud, J.A., and Lozano, Maria D., Collaborative Social Structures and Task Modelling Integration, in *Proceedings of DSV-IS'06* (Dublin, Ireland, July 2006), ACM Press, 66-80.
21. Pinelle, D., Gutwin, C., and Greenberg, S. Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. Transactions on Computer-Human Interaction (TOCHI), 2003, 10(4), 281-311, ACM Press.
22. Piper, A., O'Brien, E., Morris, M.R., and Winograd, T., SIDES: a Cooperative Tabletop Computer Game for Social Skills Development, in *Proceedings of CSCW'2006* (Baff, Alberta, Canada, November 2006), ACM Press, 1-10.
23. Royce, W., Managing the development of large software systems, in Proceedings of WESCON'70, IEEE Press, 1-9.
24. Rubart, J., and Dawabi, P. Shared Data modelling with UML-G. *International Journal of Computer Applications in Technology*, 2004, 19(3-4), Inderscience, 231-243.
25. Sallnäs, E., Rassmus-Gröhn, K., and Sjöström, C., Supporting Presence in Collaborative Environments by Haptic Force Feedback. *ACM TOCHI*, 2001, 7(4), 461-476, ACM Press.
26. Scapin, D. L., and Pierret-Golbreich, C. Towards a method for task description. *Work with Display Unit*, 1990, 371-380, Elsevier.
27. Serrano, M., Juras, D., Nigay, L. A Three-dimensional Characterization Space of Software Components for Rapidly Developing Multimodal Interfaces, in *Proceedings of ICMI'08* (Chania Crete, Greece, October 2008), ACM Press, 149-156.
28. Tang, A., Tory, M., Po, B., Neumann, P., and Carpendale, S, Collaborative Coupling over TableTop Displays, in *Proceedings of ACM CHI'06* (Montréal, Québec, Canada, April 2006), ACM Press, 1181-1190.
29. Treatteberg, H. Model-based User Interface Design. *PhD Thesis*, 2002, Norwegian University of Science and Technology.
30. Tse, E., Shen, C., Greenberg, S., and Forlines, C., Enabling Interaction with Single User Applications through Speech and Gestures on a Multi-User Tabletop, in *Proceedings of AVI'06* (Venezia, Italy, May 2006), ACM Press, 336-343.
31. Tse, E., Greenberg, S., Shen, C., Forlines, C., and Kodama, R., Exploring True Multi-User Multimodal Interaction over a Digital Table, in *Proceedings of DIS'08* (Cape Town, South Africa, February 2008), ACM Press, 109-118.
32. Van der Veer, G., Welie, M., Task Based Groupware Design: Putting Theory into Practice, in *Proceedings of DIS'00* (New York NY, August 2000), ACM Press, 326-337.
33. Van der Veer, G., Welie, M, Chisalita, C., Introduction to Groupware Task Analysis, in *Proceedings of TAMODIA'02* (Bucarest, Romania, July 2002), ACM SIGCHI, 32-39.
34. Vernier, F., Nigay, L., A Framework for the Combination and Characterization of Output Modalities, in *Proceedings of DSV-IS'00* (Limerick, Ireland, June 2000), Springer, 32-48.
35. Voida, A., and Mynatt, E.D., Challenges in the Analysis of Multimodal Messaging, in *Proceedings of CSCW'2006* (Baff, Alberta, Canada, November 2006), ACM Press, 427-430.