



HAL
open science

Contextual Modeling of Group Activity

Patrick Brézillon

► **To cite this version:**

Patrick Brézillon. Contextual Modeling of Group Activity. CONTEXT 2017, P. Brézillon; R. Turner; C. Penco, Jun 2017, Paris, France. pp.113-126, 10.1007/978-3-319-57837-8_9. hal-03971083

HAL Id: hal-03971083

<https://hal.sorbonne-universite.fr/hal-03971083>

Submitted on 3 Feb 2023

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.

Contextual Modeling of Group Activity

Patrick Brézillon

University Pierre and Marie Curie (UPMC), France

Patrick.Brezillon@lip6.fr

Abstract. We extend the Contextual-Graphs formalism for modeling interaction in a group activity. Group activity is modeled as the cyclic traversing of a contextual meta-graph representing the group activity in terms of member activities, and of the shared context as a hub for interaction management. We propose a model of group-member interaction at the level of the transitions between turns. The model relies on the representation of members' activities in terms of independent tasks that are structured by simulation parameters in the shared context.

Keywords: Group activity, Interaction model, Task realization, Turn, Contextual graphs, Contextual element, Context-based simulation

1 Introduction

The term “group” is used when the completion of some works requires crossing the organizations frontiers, enabling the creation of groups involving actors from different organizations (i.e. different hierarchies). The notion of activity takes into account contextual aspects of the task realization in addition to its theoretical definition: the activity is the (physical and mental) behavior that the actor exhibits for realizing the task [9]. Complementary to the engineering approach, the cognitive approach provides a computational support for effectively representing, analyzing and simulating activities in real-world applications. However, this supposes the integration of the contextual dimension in the representation of the activity.

Brézillon and Pomerol [5] propose the following operational definition of context: "Context is what constrains the focus without intervening in it explicitly." It results a conceptual framework that supposes a uniform representation of elements of knowledge, reasoning and context that are needed in a task realization [2].

Initially, a contextual graph (CxG) represented the model of a task realization, each path is a practice developed by an actor in a particular context for realizing the task. Formally, contextual graphs are acyclic and series-parallel due to the time-directed representation that ensures algorithm termination. A contextual graph contains four items: actions, contextual elements, activities and Executive Structures of Independent Activities (ESIA). An action is the building block of the model at the given representation granularity. A contextual element is a pair of a contextual node and a recombination node. The former has one input and N outputs (branches) corresponding to N known values of the contextual element leading to N different methods for real-

izing a subtask. The latter is a $[N, 1]$ relationship that represents the moment at which the instantiation of the contextual element does not matter anymore. An activity—a complex action described as a contextual graph by itself—is identified by actors in different task realizations as a work unit. Finally, an ESIA expresses the execution of different independent (sub-) activities in a parallel or sequential way, regardless of their order. In some sense, an ESIA is a kind of complex contextual element.

This paper is organized as follows. Section 2 presents what must be taken into account for extending the Contextual-Graphs formalism from a user's task realization to a group activity. Section 3 introduces the elements for extending the formalism for modeling group activity. Section 4 then discusses the modeling of interaction between an operator and a simulator in an application. Finally we conclude with the possibilities offered by this modeling of group activity.

2 From user task realization to group activity

2.1 Modeling of task realization

Actors use knowledge and experience (their mental representation) to interpret and to realize their tasks [3]. An actor selects a particular method to realize the task according to different contextual elements and their instantiations depending on actor's preferences, the task, the situation at hand and the available resources in the local environment.

An actor develops a practice jointly with the building of a proceduralized context, i.e. a context-specific model represented by an ordered series of instantiated contextual elements [2]. Because the structure of a contextual graph is structured by contextual elements, practices can be organized by the contextual elements in a tree representation [6].

The use of the Contextual-Graphs formalism for modeling a group activity requires some extensions for managing the way in which the different members participate in the group activity (e.g. ordered or concurrent subtasks, negotiation process, etc.). Generally, the viewpoint of the activity processing is chosen for having a homogeneous presentation of the process. The choice of the member's viewpoint on a group activity implies that the contribution of each actor in the group activity is clearly identified as a specific activity, and, second, the modeling of a group activity as a contextual graph must make these members' activities explicit. As a consequence, the context in which the group activity is performed must be enriched with specific contextual elements for modeling group-member interaction during the practice development that is generally followed by simulation.

2.2 Modeling group activity

Benitez-Guerrero et al. [1] present effective activity as an instance of an activity model that describes the family of actors that can participate in the activity, how the activity can be carried out, the family of objects that can be manipulated or produced, and which roles actors and objects will play in the activity. The notion of effective

activity is similar to our notion of practice and leads to define the realization of a group activity as the development of a group practice.

A group practice implies the building of a shared base of contextual elements. Brézillon et al. [4] present the results of an experiment for modeling verbal exchanges between two participants for building collaboratively an answer to a question. The two participants begin the development of a shared context by gathering a maximum of contextual elements. Each actor finds a way to integrate the contextual elements provided by the other participant in his own mental representation of the question. When participants think that their shared context is sufficient for building the answer, they enter the second phase of the process by organizing, assembling and structuring these shared contextual elements in a proceduralized context that will lead then to build the answer.

The shared context contains in addition specific contextual elements for the management of the group activity (turn, acceptance, etc.). Thus, the shared context is a crucial place for managing member interactions during the group-practice development as well as the management of the tasks in each member's activity.

3 Extension of the CxG formalism for group-activity modeling

Based on what happens in enterprises, we use the notions of roles and tasks for representing a group activity in terms of members' activities. Interaction between group members corresponds to a movement of the group leadership from one member to another one, and, concretely, from one subtask inside a member activity to another subtask in another member activity. The natural extension of the CxG formalism for group activity then is to represent the group activity as a contextual meta-graph where member activities are represented on the exclusive branches of a contextual element for deciding who is the manager for the turn. This implies that:

- There is a special contextual element that controls which group member is the manager of the current turn;
- The activity of each group member must be modeled as a set of independent tasks structured by another special contextual element corresponding to the management of the independent tasks in the manager activity;
- The traversing of the contextual meta-graph corresponds to the realization of a specific independent task in the manager's activity;
- The development of the group practice then corresponds to a sequence of turns taken in different members' activities, a turn corresponding to the realization of an independent task and the management of special contextual elements controlling, on the one hand, transitions between turns and, on the other hand, the task status of the group practice development;
- The shared context is composed of contextual elements specific to (i) each member activity, (ii) information that is transferred between member activities, and (iii) interaction management between members during the group-activity development.

The special contextual elements will be called hereafter "simulation parameters" (SPs). Let us first introduce a running example.

3.1 A context-based model of group activity

Garcia and Brézillon [6] use the example of a paper submission to a journal for illustrating their purposes on their CxG-based simulator. The paper-submission processing (i.e. the group activity) requires interaction among actors, which have with different roles: author, editor, reviewers and publisher. Fig. 1 shows the contextual meta-graph for managing all members' activities at the same level, and thus facilitating their coordination based on the role of manager of the group activity holds by a group member during the traversing of the contextual meta-graph.

Each branch corresponds to the activity linked to a role (i.e. the tasks associated with the role). The active role during a traversing of the contextual meta-graph is selected by the instantiation of the contextual element "Manager" to one of the values "author", "editor", "reviewer" or "publisher".

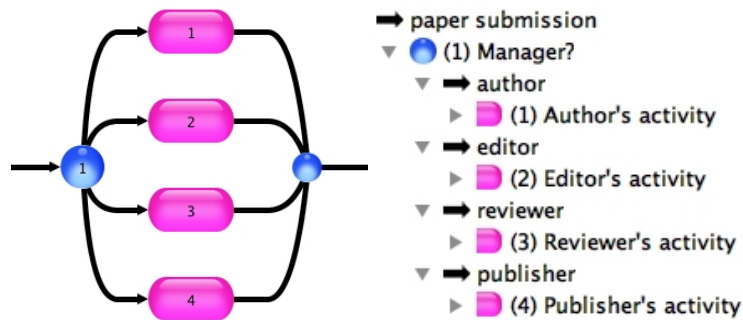


Fig. 1. The contextual meta-graph of the submission example [6]

The cyclic traversing of the contextual meta-graph represents the group-practice development across member's activities through the role of manager successively played by group members. This allows a given member to play different roles, such as the editor being a third reviewer. By entering in a member activity, then an independent task is selected and realized in that member activity. Thus, the member is the temporary manager (a simulation parameter) of the group-practice development, and the specific independent task to realize is managed by another simulation parameter for controlling the status of the group-practice development.

The challenge here is to monitor the group-practice development when different members held successively the role of manager during the cyclic traversing of the contextual meta-graph. This supposes the organization of members' activities in terms of specific subtasks, each subtask corresponding to the traversing of the contextual meta-graph. Thus, there are two simulation parameters that allow, on the one hand, the management of role—the "manager"—and, on the other hand, of the group activity—the "task status", the latter simulation parameter orients the group-practice development towards the specific task inside the manager's activity.

3.2 Simulation parameters

The simulation parameters discussed in previous section have a general interest in collaborative work. In the following we call them “MANAGER” and “TASK_STATUS”. Such simulation parameters have a different nature of those of the contextual elements intervening in the task realization. They impose a hierarchical organization on each member activity in terms of independent task. This organization is dynamical because “MANAGER” and “TASK_STATUS” for the next turn are instantiated just before exiting the independent task at hand. For example, MANAGER may be “editor” or “reviewer” in the submission problem, and “Head of the Department” and “Secretary” in another problem. This shows that simulation parameters and subtasks are intertwined.

Indeed, four simulation parameters manage a group-practice development, namely the MANAGER, the RECIPIENT, the SENDER, and TASK_STATUS. RECIPIENT is the member that will become manager at the next turn. MANAGER is the group member that is concerned by the traversing of the contextual meta-graph (this is implemented by an action “MANAGER = RECIPIENT” at the beginning of the contextual meta-graph, which represents the group activity. SENDER is the previous manager who just solicits the MANAGER for executing the turn at hand (this is implemented by an action “SENDER = MANAGER” at the end of the contextual meta-graph for the next turn). TASK_STATUS specifies the independent task in the next manager’s activity to consider. At the end of the independent-task realization, the instantiations of RECIPIENT and TASK_STATUS are modified according to the result of the independent task realized. Thus the interaction pattern is managed at the end of turns through the simulation parameter RECIPIENT. This shows that interaction sequence is related to the notions of roles and tasks in a strategy of “moment-to-moment interactions” [10], which is a simple mechanism to model group-activity realization but its simplicity provides, thanks to the Contextual-Graphs formalism, insights that inform on more complex interactions. .

For example, the editor (the RECIPIENT that becomes MANAGER) receives a submission (TASK_STATUS = “Submitted”) from an author (SENDER). At the end of the independent task, the editor’s decision is made (say, accept the submission for the journal) and the submission will be sent to a reviewer (RECIPIENT) with TASK_STATUS modified for “to be reviewed” and SENDER instantiated to “editor” for the next turn.

The notions of simulation parameter, independent task and turn are domain-independent but their values have a semantic depending on the domain and thus are domain-dependent.

3.3 The model of a turn

Now it is possible to give a definition of a turn. A turn corresponds to the traversing of the contextual meta-graph, entering the manager’s activity, realizing an independent task, managing key contextual elements for the next turn, and exiting the contextual meta-graph for going back in the shared context.

Fig. 2 represents the model of a turn where the independent task is represented by a proceduralized context and a sequence of actions, and the shared context is composed (among others) of the simulation parameters `RECIPIENT`, `SENDER` and `TASK_STATUS` and contextual elements transferring information between two successive turns. The simulation parameters play the role of activation conditions and post-conditions of the turn. The turn model is the building block of the modeling of a member's activity in terms of independent tasks and, as a consequence, of the group activity. The turn model is domain-independent but the internal description of the independent task at a finer granularity is domain-dependent (like a kind of instantiation of the turn model in a domain).

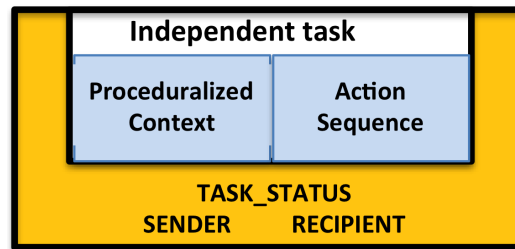


Fig. 2. Model of a turn

The turn processing is defined by: (1) select the member to be manager; (2) select the independent task in the manager's activity to realize; (3) compute the proceduralized context and make the decision (i.e. the sequence of actions to execute); and (4) manage the instantiations of the simulation parameters `RECIPIENT` (i.e. the next `MANAGER`) and `TASK_STATUS` (the next independent task in the `MANAGER` activity). It is the basic principle of the CxG-based simulation that is discussed in another paper [7]. However, for modeling an activity, the turn mechanism requires to:

- Identify turns, i.e. decompose each member's activity in independent tasks that encapsulate the operational knowledge, the reasoning and the actions that the manager has to perform for addressing the sender's request and the transmission of the relevant results to the next group member (i.e. the interaction management).
- Organize the turns during a group-activity development with a mechanism ensuring the flexibility of the interaction management among group members.

Once the representation of member activities in terms of turns made, the next step is to represent the assembling of the turns taken in different member activities in a sequence during the group-practice development in a given context.

3.4 A model of turn mechanism for interaction simulation

The turn mechanism plays a role of synchronizer (coordinator) of the group activity. A group-practice development corresponds to a sequence of turns taken in different roles (we do not discuss here about the possibility of parallel turns such as the review

process that concerns the same role holds by two or more actors in the submission example). The group activity is developed jointly with the incremental building of a turn sequence. Interaction is described as a sequence of transitions from one turn in the activity of one member to a turn in the activity of another member. The turn sequence and the on-going interaction process are progressively builds along the cyclic traversing of the contextual meta-graph in different shared contexts.

At a lower level, this corresponds to the assembling of independent tasks coming from different members' activities. The dynamical aspect of the turn assembling come from the fact that an independent task may lead to different conclusions and, thus, a turn may have different recipients for the next turn according to the specific working context associated with the manager activity. Thus, a group practice is not a linear extrapolation of an actor's practice where practices have a fixed structure given by contextual elements.

The turn mechanism can be implemented by a system of production rules. For example, the following rule will move the focus on the next actor:

```

IF      RECIPIENT is <NIL>,
THEN   stop the simulation
ELSE   MANAGER = RECIPIENT,
       go into MANAGER's activity.

```

The rule corresponds to the first simulation parameter ("Manager?") in Fig. 1.

Fig. 3 illustrates the conceptual representation of the turn management for the paper-submission example. (This part of the work will be developed in the paper on CxG-based simulation [7].)

The implementation of the turn mechanism is a simple extension of the CxG formalism that, indeed, can be applied to a task realized by a unique actor. The explicit consideration of the shared context with the contextual meta-graph opens the door to more options than before like:

- To stop the simulation (with an action "RECIPIENT = <nil>" at the end of a turn),
- To introduce loops for managing a negotiation between two actors, and for realizing several times a sub task,
- The possibility for an actor to change his goal when an unexpected event occurs (e.g. the actor select an object and discovers that finally the object is not adapted to his objective),
- To allow an actor (or a group) to backtrack in his reasoning, and
- To allocate several actors to a same role (e.g. reviewers in the paper submission example).

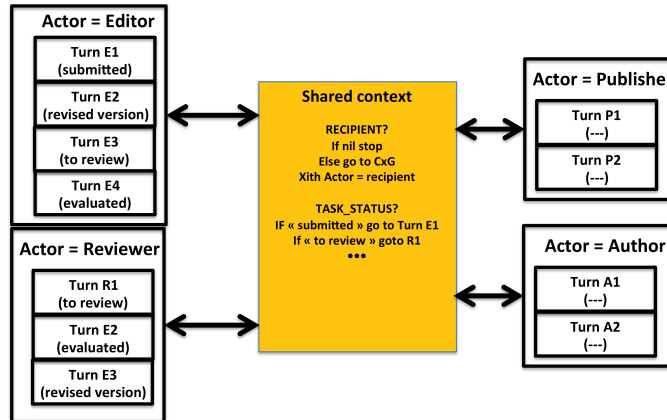


Fig. 3. A conceptual representation in term of turns of the submission example

4 Modeling Interaction

The mechanism for going from one turn to the next one is the inference engine in a CxG-based simulation of the movement of the focus during group-member interaction. The interaction modeling is introduced in the framework of a real-world application.

4.1 Brief presentation of the TACTIC Project

In the TACTIC project [8], the group activity is the monitoring of a (small part of) a battlefield through simulation, and the group leading this activity is composed of two actors, namely the operator and the simulator, which interact through an interface. The model of the operator's activity for the simulation task is a contextual graph (we use here only a limited part of the operator's activity, the task "Give an order of recognition"). The simulator's activity concerns the simulation of event evolution on the battlefield from three complementary sources of information: spatial coordinates of objects (the field map), temporal coordinates (the chronology of events) and socio-technical coordinates (ODB, the order of battle).

The TACTIC project concerned the modeling of operators' activity in the realization of the task "give an order of recognition", that is all the simulation activity was considered from the unique viewpoint of the operator. Fig. 4 represents the modeling of the activity "Manage a unit" as a contextual graph. The activity is rather sequential, beginning by (1) the choice of an area where to select a unit (first contextual element), (2) the choice of a unit (second contextual element), (3) checking if the selected unit may realize the required recognition mission, and (4) positioning the unit in the center of the window of the field map (the different colors of actions in Figures are explained in the next section).

This modeling is made according to the operator's viewpoint. As a consequence, a part of the operator's reasoning (in yellow on Fig. 4) expresses a compilation of ele-

mentary actions realized by the operator and the simulator through the interface. For example, “Show mission orders” implies to (a) select the unit, (b) ask for opening the order window, and (c) show the list of the mission orders. The lessons learned here point out that some actions depend on the simulator (e.g. show mission orders of the unit) but others concern the interaction of the operator and the simulator with the interface (e.g. open the mission-order window).

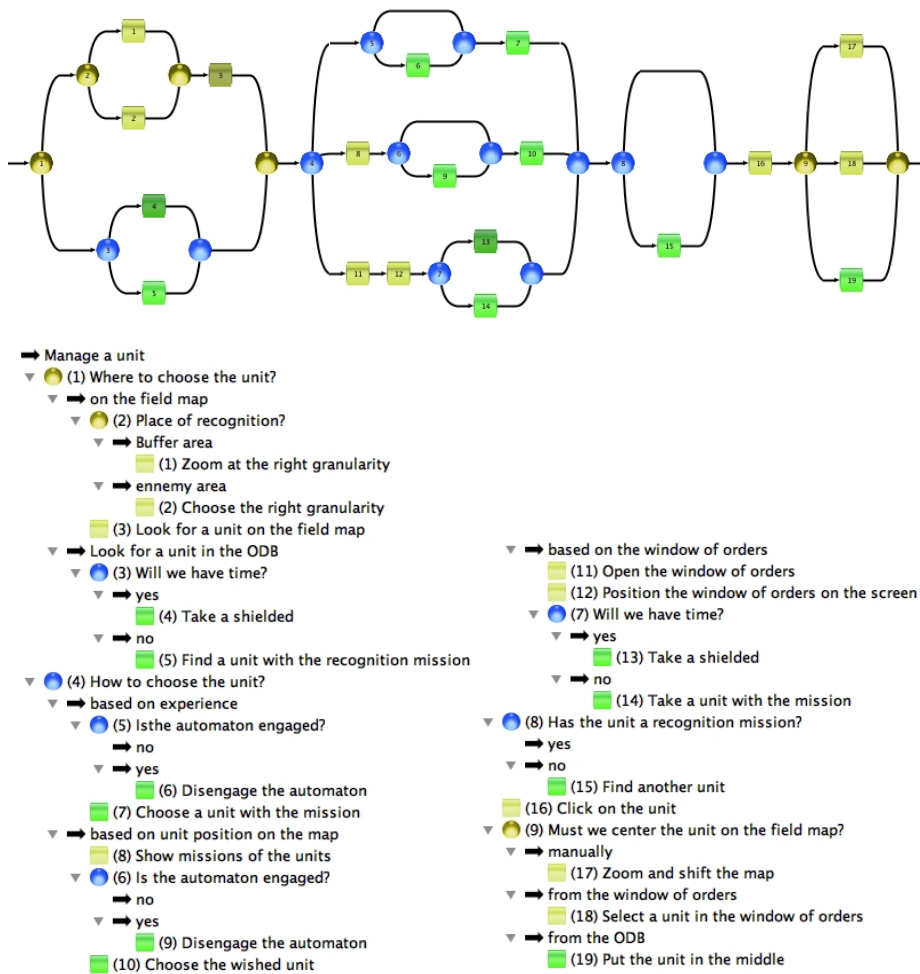


Fig. 4. The contextual graph of the activity « Manage a unit » from the operator’s viewpoint

Another lesson is that the simulator had to be considered as a group member in this activity. Items in yellow on Fig. 4 are places where the role of the simulator has been interpreted according to the operator’s viewpoint, like “Where to choose the unit?” (The first contextual element on Fig. 4) and letting in green actions that are specific of the operator, like “Find a unit with the recognition mission” (lower actions on the left

on Fig. 4). This mixing of roles in a task realization leads to a representation that is difficult to interpret.

4.2 Conceptual modeling of operator and simulator interaction

Operator and Simulator interact through an interface (the place of cognitive interaction with the operator) and the screen (the place of physical interaction with the interface for the visualization of the simulation).

The modeling in the CxG formalism highlights that computer-mediated management of interaction has a dimension that is often ignored, namely the role of the interface in the interactions. The non-explicit consideration of the interface is responsible of unnecessary turns. Thus, the operator has to interpret, on the one hand, the simulation (his task) and, on the other hand, the functioning of the interface for translating domain actions on the simulation into commands to the interface (interface actions). For example, by clicking on the pause button, the operator thinks to stop the simulation, while this action is transferred by the interface to the simulator that suspends the simulation. As a consequence, the operator associates the control of the simulation with actions on the interface that plays the role of the simulator for the operator. Thus, operator-simulator interaction (cognitive interaction) is considered secondary to interaction with the interface (physical interaction). Indeed, operators in our experiment said to “interact with the simulation”, not with the simulator because the simulation was the visible (i.e. compiled) part of the combined work of the interface and the simulator.

When interacting with the simulator, the operator based his choice on contextual information like his preferences, the task, the situation, and the environment. On its side, the simulator monitors the simulation based on three sources of information: the map, the ODB (order of battle) and the chronology. For executing an action like “Select a unit”, the operator must each time look on the interface what is the best information source in the current context because each of them has advantages and disadvantages (browsing a long list of items in the ODB, too many items appearing on the map) that imply additional actions like use a search engine for the ODB or zoom on the map [8]. Thus, the operator “contextualize” each domain action with interface actions in order to select the best action in a given context. Domain actions will be more easily associated with an interface action if the shared context is made explicit, resulting in greater flexibility of the interface, not only with respect to the actor, but also with respect of the task realization.

Fig. 5 shows a conceptual view of the previous discussion in order to simplify operator’s activity during interaction with a simulator. A first observation concerns the need to distinguish the interface interaction and operator-simulator interaction. This supposes the separation of “domain_actions” and “interface_actions” [3]. The second observation concerns the cognitive interaction, i.e. to make compatible the mental map of the operator with the “mental map” of the simulator (i.e. the three sources of information). Clarifying the fact that a simulation results of the interaction of the operator with a simulator gives the interface the role of a flexible communication medium included in the shared context of the simulation through which the operator (with domain_actions in his reasoning) and the simulator (with domain_actions in its model of the battlefield) communicate with a simple translation in interface_actions.

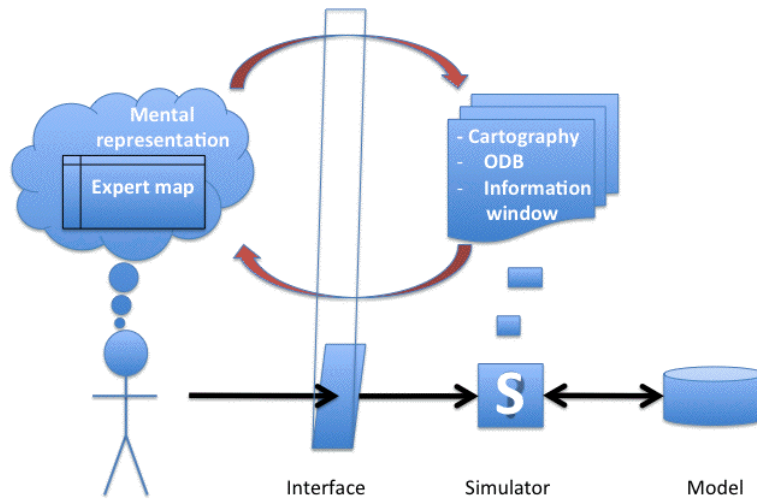


Fig. 5. A model of user-simulator interaction

4.3 Implementation of the interaction model

The contextual meta-graph of this particular group activity contains two branches only, one for the operator and the other for the simulator (but a more precise model would have the interface as a third group member). Fig. 6 shows the result of a partial re-writing of the initial operator's activity described in Fig. 4 by separating the independent tasks specific of the operator from those specific of the simulator (i.e. the two columns), often by "decompiling" operator's actions in Fig. 4. The rectangles with a label O_i or S_j in Fig. 6 represent the independent tasks and key contextual elements management respectively of the operator and of the simulator. Arrows symbolize the transitions between turns. Rectangles represent the part of the turn corresponding to the independent tasks like in Fig. 2 and arrows correspond to the conceptual representation of interaction mechanism on Fig. 3.

Most of the independent tasks in Fig. 6 are not detailed (i.e. the branches of contextual elements are not visible) to facilitate a focus on the study of the turn mechanism symbolized by arrows. Items concerning the domain keep their initial colors (i.e. green square boxes for actions and blue circles for contextual elements). Simulation parameters for the shared-context management appear in the figure as yellow items. Actions in pink (both in operator and simulator activities) are interface actions (e.g. the action 19 "Click on the chosen unit" is an operator's action on the interface).



Fig. 6. Partial model of the operator and simulator interaction

Such an interaction model presents several interesting features. First, the use of the simulation parameters introduces, in a natural way, the notion of loop (e.g. independent task O4), thanks to the association of the contextual graph and the shared context. Secondly, a turn starts in the shared context by checking the instantiation of MANAGER, and then of TASK_STATUS that directs the focus on an independent task in MANAGER's activity. For instance, the first turn in the operator's activity corresponds to Value (TASK_STATUS) = "find_area (0)" that moves the focus on the independent task O1 and yellow actions at the end of the rectangles on Fig. 6 modify the instantiation of the simulation parameters TASK_STATUS and RECIPIENT for the next turn. Thirdly, the turn with the independent task O4 is particular in the sense that there are three possible outputs, two towards another operator's turn (arrows on the left) and one towards the turn S4 in the simulator's activity. The two arrows on the left correspond to a loop in the operator's activity for changing of unit. Fourthly, in addition to the simulation parameters discussed previously for managing the turn mechanism, there are other yellow contextual elements that are also instantiated in one member's activity and use in another one and thus are shared by the group members. For example, the contextual elements "local area" and "automaton" in Fig. 6 are instantiated by the operator and guide the choice of the next independent task in the simulator's activity (contextual elements 2 and 8). However, such contextual elements are domain-dependent and only ensure an information transfer between members.

In addition to what is presented in Fig. 6, the explicit association of the shared context with the contextual meta-graph solves the main weaknesses of the contextual

graph used alone, namely: (1) the turn O4 allows the operator to change his reasoning when encountered an unexpected event (e.g. the unit do not have the required mission) and thus compensates the acyclic nature of contextual graphs; and (2) the simulation parameter “RECIPIENT” = <nil> allows to “stop” the turn mechanism and to exit the group activity (and stops the CxG-based simulation).

The lessons learned from the TACTIC project are threefold. The first lesson concerns, at the physical interaction level, the differentiation of the actions of the operator and those of the simulator for the control of the simulation, and finally the identification of the actions that are related to the interface. The second lesson is to make compatible, at the level of the cognitive interaction, the operator’s mental representation and the three sources of information of the simulator, say the “simulator’s mental representation”. The third lesson is the interest to consider the interface as a real group member in a computer-mediated group activity (but not modeled in this study) with the operator and the simulator.

5 Conclusion

The paper proposes an approach for a contextual modeling of group activity at two levels. At a conceptual level, group activity is modeled as a sequence of interactions between the group members. In our model, a turn is the building block of the group-activity representation, and the shared context is the place where are managed interactions between group members. At the implementation level, a group activity is represented as a contextual meta-graph composed of members’ activities that are built as independent tasks. The contextual meta-graph is structured by simulation parameters that manage member interactions through the turn mechanism in the shared context. Thus, a group activity is simultaneously built and developed by combining task realization and interaction management.

The set of changes in our initial conceptual framework ensures a logical shift of our paradigm of Contextual Graphs from the representation of a task realization by an actor to a group activity. Indeed, a group activity is expressed as the cyclic traversing of the contextual meta-graph in different contexts, while a task realization corresponds to a unique traversing of the contextual graph.

We discussed all these points in reference to different real-world applications, namely, (1) the paper submission modeling, which proposes a modeling of a group activity as a contextual meta-graph; (2) the building of the answer to a question, which points out the need to build a shared context from contextual elements specific to each actor, contextual elements ensuring exchanges between members’ activities during interaction, and contextual elements for managing interaction (simulation parameters) as a prerequisite to a group activity; (3) the battlefield simulation, which introduces in the group-activity modeling the importance to separate cognitive interaction (mental representations and thus context) and physical interaction (at the interface level) in order to model interactions in a group activity. Interaction is modeled in relation to the mental maps of the group members. The interface must be considered as a full member in the group activity.

The notions of turn mechanism, independent task, key contextual elements open the door to a new type of simulation of interaction in a group activity such as the CxG-based simulation that is the topic of a companion paper [7]. A tool for exploiting this new version of the CxG formalism, the CxG Simulator is under study.

6 References

1. Benitez-Guerrero, E., Mezura-Godoy, C., Montané-Jimenez, L.G.: Context-Aware Mobile Collaborative Systems: Conceptual Modeling and Case Study. *Sensors*, 12 (2012) 13491-13507
2. Brézillon, P.: Task-Realization Models in Contextual Graphs. In: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), *Modeling and Using Context (CONTEXT-05)*. Heidelberg: Springer Verlag, LNCS 3554 (2005) 56–68
3. Brézillon, P.: Modeling expert knowledge and reasoning in contextual graphs. In: H. Christiansen, I. Stojanovic, & G. Papadopoulos (Eds.), *Modeling and Using Context (CONTEXT-15)*, Heidelberg: Springer, LNAI 9405 (2015) 18-31
4. Brézillon, P., Drai-Zerbib, V., Baccino, T., Therouanne, T.: Modeling collaborative construction of an answer by contextual graphs. In: B. Bouchon-Meunier, & R.R. Yager (Eds.) *Proceedings of IPMU-2006*, Paris: Editions EDK (2006) 11-26
5. Brézillon, P., Pomerol, J.-Ch.: Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3) (1999) 223-246.
6. Garcia, K., Brézillon, P.: A contextual model of turns for group work. In: H. Christiansen, I. Stojanovic, & G. Papadopoulos (Eds.), *Modeling and Using Context (CONTEXT-15)*, Heidelberg: Springer, LNAI 9405 (2015) 243-256
7. Garcia, K., Brézillon, P.: CxG-based simulation. In: P. Brézillon, R. Turner & C. Penco (Eds.), *Modeling and Using Context (CONTEXT-17)*, Paris: Springer, LNAI (submitted)
8. Kabil, A., Brézillon, P., Kubicki, S.: Contextual interfaces for operator-simulator interaction. In: H. Christiansen, I. Stojanovic, G. Papadopoulos (Eds.), *Modeling and Using Context (CONTEXT-15)*, Heidelberg: Springer, LNAI 9405 (2015) 483-488
9. Sarrazin, P., Biddle, S., Famose, J.-P., Curry, F., Fox, K., Durand, M. (1996). "Goal orientations and conceptions of the nature of sport ability in children: A social cognitive approach", *British Journal of Social Psychology*, 35 (1996) 399-414
10. Hackmans, J.R., Morris, C.G. : Group tasks, group interaction process, and group performance effectiveness : A review and proposed integration. *Advances in experiential Social Psychology*, 8 :45 (1982) 1-55