CxG-based simulation of Group Activity

Simulation basée sur les graphes contextuels d'une activité de groupe

Patrick Brézillon[1]

[1] Sorbonne University, Laboratoire d'Informatique de Paris 6 (LIP6), F-75005 Paris, Patrick.Brezillon@lip6.fr

RÉSUMÉ. Le formalisme des Graphes contextuels permet de représenter la réalisation d'une tâche par un acteur. L'objectif est de fournir aux décideurs un panorama clair des différentes pratiques pour réaliser une tâche, et ainsi pouvoir choisir la meilleure pratique dans le contexte rencontré. Le formalisme des Graphes Contextuels a été utilisé avec succès dans de nombreux domaines tels que la médecine, la biologie, et les transports. Dans ce papier, nous proposons une extension du formalisme pour modéliser des activités collaboratives, et illustrer les extensions apportées dans une application de taille réelle pour l'Armée (projet TACTIC). Les extensions portent sur (1) l'introduction de paramètres de simulation pour manager les interactions entre acteurs, (2) une notion de tour pour modéliser une traversée du méta-graphe contextuel (le modèle de l'activité du groupe), (3) un mécanisme de gestion des tours entre les membres du groupe pour manager l'utilisation cyclique du méta-graphe contextuel, et (4) un simulateur basé sur les graphes contextuels pour étudier les interactions entre les acteurs. Une réelle compréhension d'une activité collaborative ne se limite pas à aider le décideur, mais à fournir aussi des outils pour construire des applications opérationnelles.

ABSTRACT. The Contextual-Graph (CxG) formalism allows the representation of a task realization by an actor. The objective is to provide decision makers with a clear panorama of the different ways to realize a task (i.e. the practices), and thus of the implications of choosing one alternative or another. The Contextual-Graph formalism has been successfully used in many fields, such as medicine, biology, and transportation. In this paper, we extend the formalism to modeling group activity, and illustrate its extensions on a real-world application called the TACTIC project. The extensions concern the introduction of (1) simulation parameters for managing the interactions among the actors, (2) the notion of turn for representing a traversal of the contextual meta-graph (the model of the group activity), (3) a turn mechanism for incorporating the cyclic use of the contextual meta-graph, and (4) a CxG-based simulator for studying interaction among group members. A real understanding of group activity does not just help decision makers, but also provides real requirements for building successful applications.

MOTS-CLÉS. Activité de groupe, activité collaborative, modèle d'interaction entre acteurs, réalisation d'une tâche, contexte, Graphes Contextuels (CxG), simulation basée sur les graphes contextuels, mécanisme de management de tours entre les acteurs.

KEYWORDS. Group activity, collaborative activity, actor-interaction model, task realization, context, contextual graphs (CxG) formalism, CxG-based simulation, turn-management mechanism.

1. Introduction

The notions of team and group often are commonly used interchangeably to refer to a set of people who are concerned with a given mission. However, such notions differ on their characteristics and purposes. Generally, an organization's structure relies on the notion of team. A team is a group of employees in a company making a collective effort in the interest of a common cause [39]. Team members are selected according to their characteristics and skills in order to accomplish the mission. Before team members start working together, the leader learns about the team members, identifies their competences and skills, and is able to design a precise work plan. Once this process is completed, the team leader is responsible for dividing the work among the team members and for controlling the evolution of the teamwork. Teamwork is a relatively well-specified notion.

The term "team" is no longer relevant when the completion of some missions requires crossing the organizations' boundaries, enabling the creation of groups composed of actors from different organizations (i.e. different hierarchies). An actor can be spontaneously added to (or removed from) a group at any moment. Group members act as a whole in order to get something done [8]. Group work results from the distribution of the workload among group members. A group lasts up to the completion of the work. As in a team, there is a unique group leader, but each group member can ask for a support directly to other members. The meaning of group work is no more accurate than for teamwork because the term “work” is at a too high conceptual level and is not related to what is actually done. The use of “task” instead of “work” corrects the problem because there is an emphasis on the way in which the task is done (one speaks of task model, not of work model except some particular aspects like performance).

The engagement of members during group work is often discussed based on three concepts: coordination, cooperation and collaboration. Coordination is the technical organization of the different elements of a collective activity to enable participants to work together effectively according to a plan. The term coordination has a well-accepted definition across disciplines such as cognitive sciences (AI, psychology, etc.) or engineering (CSCW, interface design, etc.) [5, 23, 30]. Collaboration and cooperation are ambiguous terms that have different meanings across domains and sometimes from one author to another even within the same domain. Collaboration is “working together” to reach a negotiated and consensual solution to create something new with a shared vision. Collaboration concerns the realization of the same task by different actors. The unit is not the participant but the group with a global and collective responsibility. There are two realizations in parallel, a joint realization and individual realizations. Each participant contributes to the action of any other participant of the group to increase the performance, and collaborative interaction is permanent. Cooperation means “operate together” with, first, a negotiated division of the task realization among actors (a common goal reached by autonomous actions) and, second, a pooling by assembling subtasks realized in a flat way, each participant having to bear a definite part of the common task realization. Cooperative interaction is limited to the organization, the coordination and the monitoring of the task realization.

The ambiguity between collaboration and cooperation is related to the nature of actor interaction, and more specifically: (a) the goal of interaction, (b) the shared objectives of the actors; (c) the entities involved (from individuals to enterprises); (d) the context that actors need to share; (e) the type of interaction (i.e., collocated vs. remote and synchronous vs. asynchronous); (f) the needed technologies to support the different types of interaction; and (g) the scope of the joint work regarding actors’ individual work. As a consequence, it is difficult to identify clearly what group work is because what is a simple collaboration at one moment may become cooperation at the next moment. Moreover, the mechanism ensuring interaction between members during group work is difficult to model because interaction is dynamical and depends essentially on the type of group work. As a consequence an efficient modeling of group work requires making the context of the group-work development explicit.

The operational definition "Context is what constrains the focus without intervening in it explicitly" [18] points out the relationship between context and the focus of the actors (of the group work) that implies two parts in context, namely, external knowledge and contextual knowledge. External knowledge contains elements that are not important at the moment when we consider the focus. Contextual knowledge constitutes a kind of pool of elements that may be related to the focus. A contextual element, which is an “element of the natural world,” allows the management of alternatives for realizing a part of the task according to its specific instantiation (i.e. the value selected for the focus at hand).

The four main sources of contextual elements are the actor, the task, the situation and the local environment. The focus is associated with a working context that contains a set of contextual elements (i.e. the contextual knowledge), their values corresponding to known alternatives and their instantiations at a given instant. In the situation at hand, the focus allows the actors to select a subset of contextual elements in the contextual knowledge for realizing the task at a given moment. This subset of instantiated contextual elements is called the proceduralized context hereafter. The working context has a dynamic nature because the focus evolves with the progressive realization of the task. As a consequence, the way in which a result is obtained is as important as the result itself and, thus, it is more important to model practices that are effectively developed by actors rather than (official) procedures.

Brézillon and Pomerol [19] defined decision-making as a two-step process of contextualization. The first step concerns the selection of contextual elements for diagnosing the situation and the gathering of their instantiations at the moment of the diagnosis. The role of instantiation is to contextualize the situation in a unique way. The second step corresponds to the actions that have to be performed to realize the task after the proceduralization of the context. Thus, actors develop practices instead of only applying procedures [11]. A practice is developed jointly with the building of a proceduralized context, i.e. the implementation of a context-specific model of the situation [12].

Modeling a group task requires a formalism allowing a uniform representation of elements of knowledge, reasoning and context like the CxG formalism [10]. Initially, a contextual graph represents the model of a task realization, not the task model. Each path in a contextual graph represents a practice that was effectively developed by an actor in a working context leading to specific decision-making. Thus, a contextual graph captures the accumulated experience of one or several actors that realize the same task individually. The contextual graph is a kind of “contextualized procedure” at a tactic level, and a practice is the instantiation of the contextual graph in a specific context at the operational level. Formally, contextual graphs are acyclic and series-parallel due to the time-directed representation that ensures algorithm termination. Series-parallel graphs are used in a wide range of application, the most common being electrical circuits [24] and scheduling problems [25]. In these domains, flows follow all the branches simultaneously while only one path at a time is followed in a contextual graph due to the uniqueness of the instantiations of contextual elements.

A direct use of this formalism for representing a group activity requires some extensions for managing the way in which the members participate in the group activity (e.g. ordered or concurrent subtasks, negotiation process, etc.). This implies we must: (1) express the activities of all actors in a unified view for modeling interaction, and (2) enrich the context of the group-activity development with specific parameters for modeling group-member interaction between members during their activity.

This paper is organized as follows. Section 2 introduces modeling a group activity in an extended version of the Contextual-Graphs (CxG) formalism, including notions of simulation parameters for managing the interactions among actors, of turn for modeling the traversal of the contextual graph, and of the turn mechanism for managing the CxG-simulation. Section 3 presents the dynamic visualization of the development of a group activity by the use of a new type of simulation called CxG-based simulation (the global visualization of a group activity) and a complementary representation of Contextual Graphs called Practice-Tree representation (the local visualization of a group activity). The presentation of this section relies on the TACTIC project, a real-world application for the French army. Section 4 offers a discussion on the new perspectives offered by the extended CxG approach.

2. Modeling of group activity

2.1. *Related works*

Many works have been developed to propose modeling languages for understanding the way(s) that group work is performed. Their objective varies, from understanding the task to the building of a useful and responsive user interface, to model a piece of software for supporting a group task. However, none of these works provides a simple, clean, and comprehensive model that can be used at any level of granularity. The APM modeling language [21] gives a general overview of the modeled process and provides a useful way to determine the input and output of a task, as well as the flow of the process. Nonetheless, the diagram can quickly become unreadable. In their favor, the Contextual Design model [7] and the DUTCH method [44] are based on a series of diagrams, each one covering one aspect of the task modeling. Unlike APM, these approaches are good in providing different granularity levels for understanding the group task. However, as the modeling is divided in several diagrams, it is difficult to have a quick overview and understanding of the task. Each representation is built separately. Moreover, the diagrams that represent the interaction among actors are just a small tweak from either UML Activity diagrams or Workflow diagrams incorporating legends denoting the passing of parameters. Finally, the CTT model [35] and a view of the CIAN notation [33] are based on a diagram with a hierarchical structure that divides a task into smaller pieces of work, until arriving at a single unit performed by an actor. CTT provides a wide range of operators to specify the sequence, input and output, choices, parallelism of actions, etc. However, this view is difficult to follow, as there are many graphic details in a single tree-like structure. Moreover, when modeling a complex task that includes several actors, and actions, the tree-like structure fails to show the sequence of the task performance and the interaction among actors in an easy manner. CIAN tries to overcome this problem by adding sociogram, data, and interaction view. However, it also requires several diagrams to represent a task completion.

Such task modeling often relies on a decontextualization process in order to have a large application of the task model, while actors have to apply the task model in specific contexts that require adaptation of the task model. As a consequence, the actor uses knowledge and experience to interpret and decide how to contextualize the task realization in the specific context [13]. Indeed, the mental representation of the actor is a mixture of interpretation of the domain (operational knowledge) intertwined with an interpretation of the communication means. The expert map—a kind of mental signature of the actor—is a good approximation of his mental representation. An expert map is like an actor's ontology of operational knowledge of the domain. However, different actors have different mental representations of the way to realize the same task [13]).

The notion of "activity" extends the notion of "task". An activity is the concrete implementation of a task that is realized by an actor in a specific situation. The task is what an actor must do, and the activity is the (physical and mental) behavior that the actor exhibits during task realization [41]. Associated with each activity, there are one or more scripts that describe the ways an activity must be executed. A particular context can determine the script to be run for an activity. Thus, activity takes into account the result as well as the process leading to the result, i.e. the task realization from an external viewpoint. A task has a formal definition, but the notion of activity includes practical aspects of the realization of the task in addition to its theoretical definition.

A group activity is related to the way in which the group jointly realizes the task [1]. Benitez-Guerrero et al. [6] present the 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 the notion of practice introduced previously, and leads to define the realization of a group activity as the development of a group practice.

A group activity relies on the activities of the members, but it is more than the simple sum of them because they are interdependent. Each member generally intervenes several times during the development of the group activity. Each intervention of a member in the group-activity development can be represented as an independent task and the interaction between members during the group-activity development is better understood as a cycle of realization of independent tasks taken in the activities of the different members. As a consequence, the group-activity development corresponds to a sequence of turns, each turn being composed of (1) the choice of the temporary manager, (2) the selection of independent task in the manager's activity; (3) the realization of the independent task and (4) the definition of the next manager and its independent task to realize according to the conclusion of the turn at hand. The last point leads to a dynamic organization of the group activity, including loop in a member's activity, stop of the task realization and different possible paths. A turn is a local contribution of an actor to the group activity that corresponds to one's traversal of the contextual graph (a path).

A group activity supposes the sharing of contextual elements. This shared context is composed of the contextual elements of the working context attached to the group activity and the contextual elements that are specific to the group-activity management. The shared context plays an important role in the management of actor interaction during the group-activity development as well as the management of the independent tasks in each actor's activity.

2.2. *Contextual approach*

2.2.1. *Introduction*

A practice is the expression of an actor's activity in a task realization in a specific context, and a contextual graph corresponds to the implementation of all the practices that have been developed by different actors for realizing the same task in different contexts. The development of a practice by a group corresponds to the course of a sequence of turns taken in the activities of different group members. Each traversal of the contextual graph corresponds to a turn. A sequence of turns is built incrementally during the cyclic traversal of the contextual graph and of its working context. The development of a group activity is not a linear combination of development of actors' activities.

The practices are structured in the contextual graph by contextual elements [10]. This structuring is different, on the one hand, from the knowledge bases of expert systems represented in a flat way because context is not represented explicitly, and, on the other hand, from knowledge organization in an ontology where links between concepts depend on the domain (is-a, kind-of, etc.) while elements in our context model concern the operator, the task, the situation and the local environment. Indeed, Clancey [22] proposed that solving a particular problem (e.g. diagnosing a patient) involves creating incrementally situation-specific models. "Situation-specific" refers to a particular case, setting, or scenario. The "situation-specific model" is embedded in the task realization as a static model-based description fixed initially and filled progressively during the problem solving. Conversely, the context-specific model (i.e. the proceduralized context) is built progressively jointly with the practice development, depending on the instantiations of the contextual elements used in the reasoning. The importance of the difference between procedures and practices, which are developed in particular contexts, is acknowledged also in other domains with the distinction between task and activity [32], logic of functioning and logic of use [38], etc.

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 corresponding recombination node. When a contextual node is encountered, an element of the situation must be analyzed. The value of the contextual element, its instantiation, is taken into account as long as the situation is under analysis. The contextual node has one input and N outputs (branches) corresponding to N known values of the contextual element leading to N different methods for realizing 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. The working context contains all the contextual elements that exist in the contextual-graph, their values (on the different branches) for all the known alternatives, and the instantiations, i.e. the value taken by contextual elements on a specific path of the contextual graph that is currently followed.

2.2.2. *Contextual meta-graph*

We distinguish now the contextual meta-graph that is the model the group activity and the different contextual graphs representing the activities of the members. Figure 1 shows the contextual meta-graph for separately representing the activities of the operator and of the simulator in the TACTIC project (see next section) at the same level, and thus facilitating their coordination based on who holds the role of manager during the traversal of the contextual graph. Adding a new actor consists of adding a branch in the contextual meta-graph and the activity of this new actor in the group activity. The shared context is associated with the contextual meta-graph and includes the working context associated with the two contextual graphs and simulation parameters (see next subsection). The contextual graphs representing group-members' activities are represented as independent tasks at a finer granularity for integrating member interaction.

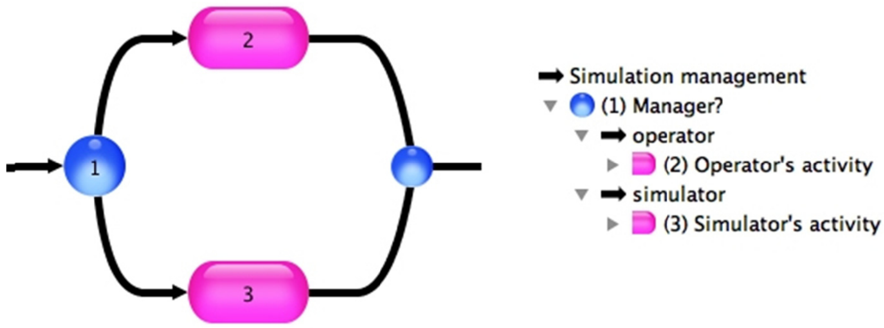

Figure 1. *The contextual meta-graph of the TACTIC project*

Figure 1 shows the group-activity model in which each branch corresponds to a group member considered through its activity (i.e. the realization of the tasks associated with the role). The current active role in the group activity corresponds to the instantiation of the contextual element "Manager" with the value either "operator" or "simulator". At a given moment, an actor is the temporary manager of a specific part of the group activity. This specific part is an independent task in the actor's activity for managing a request of the other actor. Thus, the development of a group practice progresses from one turn taken in an activity to the next turn taken in another activity. This allows the identification of the exact points of interaction among actors inside activities.

The processing of a turn corresponds to (1) the selection of the actor that will be the manager of the turn, (2) the selection of the independent task in the contextual graph of the manager's activity, (3) the computation of the proceduralized context and the realization of the independent task, and (4) the updating of the shared context for the next turn with the definition of the next manager of the group practice and the independent task to realize. At the level of the CxG representation, this means entering the contextual meta-graph for reaching and executing an independent task in the manager's activity, exiting the contextual meta-graph with an updated shared context and re-entering the contextual meta-graph in the new shared context for the next turn. It is the basic principle of the CxG-based simulation that is discussed in a following section.

2.2.3. *Simulation parameters*

The modeling of the cyclic traversal of a contextual graph requires two particular contextual elements, one for identifying the manager (called "MANAGER"), and another one for determining the part of its activity to consider as an independent task to realize during the turn (called "TASK_STATUS"). Being different than the contextual elements of the working context, such contextual elements are called "simulation parameters" because a simulation corresponds to the cyclic traversal of the contextual meta-graph during the group-practice development. "MANAGER" and "TASK_STATUS" do not intervene either in the task realization or its definition (they are domain-independent), but offer the possibility of a hierarchical organization of the independent tasks in a group practice. In that sense, simulation parameters are task-independent, but their values have a semantics depending on the domain and thus are domain-dependent. For example, MANAGER is "operator" or "simulator" in the TACTIC project, and "editor", "author", reviewer" or "publisher" in the paper-submission example discussed in [27]. The hierarchical organization of the independent tasks, which results of these simulation parameters, is managed dynamically because the simulation parameters for the next turn are instantiated just before exiting the turn at hand according to the result of the independent task. Simulation parameters are easily distinguishable from contextual elements related to the task realization because their instantiation are made by particular actions in the contextual graph (e.g. actions like MANAGER = "operator"), while the others are instantiated from the working context or provided by the user during the simulation.

Indeed, there are four simulation parameters for managing a group-practice development in the CxG software, namely MANAGER, RECIPIENT, SENDER, and TASK_STATUS. RECIPIENT is the group member that will become manager at the next turn. MANAGER is the group member that is concerned by the traversal of the contextual meta-graph. It is instantiated by the action "MANAGER = RECIPIENT" that is generally the first item on the contextual meta-graph. SENDER is the manager of the previous turn and was instantiated by the action "SENDER = MANAGER" at the end of the previous turn. TASK_STATUS specifies the independent task in the 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 (with different possible values if this independent task has different outputs). Thus the interaction pattern is managed at the end of turns through the simulation parameters RECIPIENT and TASK_STATUS, the two others, MANAGER and SENDER, being deduced automatically. These actions guide the group-practice development and are a way to represent a mechanism for turn management [13]. This shows that a sequence of interactions is related to the notions of roles and tasks in a strategy of "moment-to-moment interactions". In the CxG formalism, simulation parameters are represented in upper case letters and the actions where they are instantiated are in yellow color in figures.

2.2.4. *Modeling turns*

As said previously, a turn corresponds to one traversal of the contextual meta-graph with: (1) the actor that is the instantiation of MANAGER, (2) the independent task in the manager's activity that is the instantiation of TASK_STATUS; (3) the realization of the independent task and (4) the choice of the next manager and its independent task for the next turn according to the conclusion of the independent task just realized.

Figure 2 represents a model of a turn, based on previous observations. The shared context (the yellow part of Figure 2) is composed of the simulation parameters SENDER, RECIPIENT and TASK_STATUS, and the working context. The independent task, which is realized during the turn, has two parts, the proceduralized context and the action sequence as established in [19]. The simulation parameters are the conditions of activation of the turn and the post-conditions once the turn is completed. The turn model on Figure 2 is the building block of the group-activity representation. This model is domain-independent, because all the domain-dependent parts are at a finer granularity in the internal representation of the independent task, i.e. the proceduralized context and the action sequence in Figure 2).

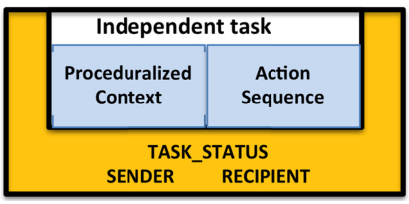

Figure 2. *Model of a turn*

The next step concerns the representation of the movement between turns during the development of the group practice for ensuring the cyclic traversal of the contextual meta-graph. Concretely, this supposes to traverse again the contextual meta-graph for realizing the next turn in the updated shared context that is the place for the turn mechanism. At a conceptual level, the turn management corresponds to a management of the manager role and of the independent task being considered. At the implementation level (i.e. the CxG formalism), a turn corresponds to a complete path from the input to the output of the contextual meta-graph. In some sense, the shared context plays the role of synchronizer of the turn mechanism and a kind of "hub" of the activities of the members because, beyond the simulation parameters, there are contextual elements in the working context that allow information exchange between independent tasks.

The sequence of turns generated during the development of the group practice is a model of group-member interaction. Thus, the introduction of simulation parameters makes it possible to represent group activity as a sequence of interactions between group members that communicate through the shared context. However, this turn mechanism requires two things. First, identify turns, i.e. decompose each actor's activity in independent tasks that (a) are structured by simulation parameters, (b) encapsulate the operational knowledge, the reasoning and the actions that the manager has to perform and (c) transmit the relevant results of the independent task to the next actor. Second, organize the turn sequence during a group-practice development with a mechanism for ensuring the flexibility of the turn-sequence building. Thus, a contextual meta-graph represents the knowledge needed for realizing independent tasks in activities and the way in which this knowledge must be used all along the progress of the group activity.

The turn mechanism can be easily represented by production rules exploiting the shared context as represented in Figure 3. For example, when a turn ends, the system detects what to do next as shown in rule in the shared context.

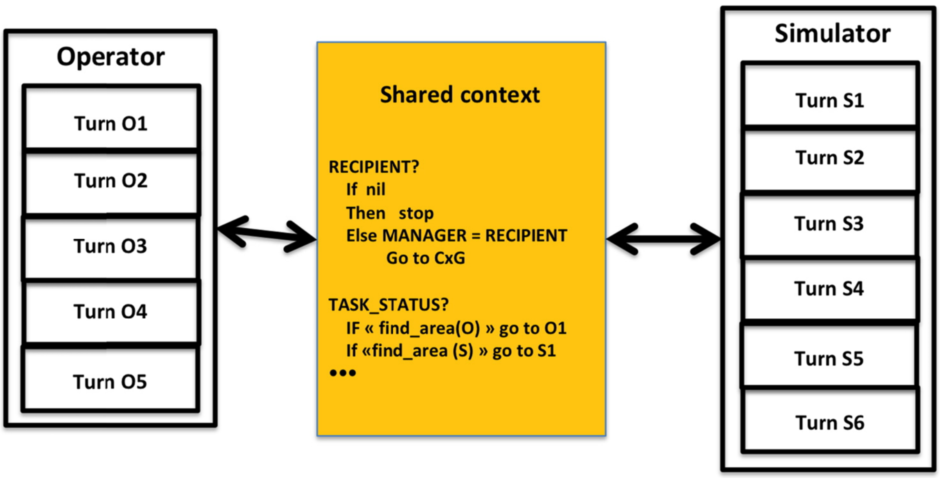

Figure 3. *A conceptual representation in term of turns of the TACTIC example*

Thus the turn mechanism can be managed simply at the shared-context level, keeping intact the initial properties of the CxG formalism (only a management of the simulation parameters was added). However, this explicit consideration of the shared context associated with the contextual meta-graph open the door to more options like to stop the simulation (with an action RECIPIENT = <nil> at the end of a turn), to introduce loops, say, for managing a negotiation between two actors, the possibility for an actor to change its objective if an unexpected event occurs (e.g. the actor selects an object and discovers that the object is not adapted to his objective), or to introduce backtracking in the reasoning of one actor (or at the level of the group). This opens also the door to the same role being allocated to different actors working in parallel and a new insight on the problem of cooperation versus collaboration discussed in the introduction (we will discuss the collaboration continuum in another paper). Note that the user keeps control of the group-activity development by providing instantiations required.

3. Visualization of group activity

3.1. *Global visualization by CxG-based simulation*

3.1.1. *Model-based versus CxG-based simulation*

Simulation is the imitation of the operation of a real-world process or system over time [4]. Simulation is often used with scientific models of natural systems or human systems to gain insight into their functioning, especially when the real system is not accessible, or it may be dangerous or unacceptable to put in danger [42]. Another use of simulation is training [40, 34]. Simulation also is a cognitive activity like reasoning or task realization. Frequently, the simulation is conducted using software tools. For example, computer simulation is used to formally model theories of human cognition and performance, such as in ACT-R [2].

Usually, a simulation describes the evolution of a (formal) model, starting from a set of initial conditions. The model expresses a statement about a real system that is based on formalized concepts and hypothesis about the functioning of the real system. Formal models address the evolution of a real system, and the corresponding trajectory is unique because the model structure is unique (parameter values are constant during the entire simulation). Thus, a model-based simulation relies on {model structure, parameters, initial conditions on variables}, where model structure, parameter values and initial conditions are fixed during all the simulation. Such a model is given by a structure that is specified by parameters that appear in the relationships between variables (a typical formalism of representation is differential equations). A model-based simulation gives a description of the evolution of the variables with respect to an independent variable, generally time, given a set of values for the parameters and a set of initial conditions for the variables. The evolution of some variables is then compared to temporal observations of the real system. In a formal model, time appears through the evolution of the variables from the model structure and relationships between variables (e.g. $y(t)$ in a model expressed in the formalism of differential equations like $dy/dt = -a*y + b$). The working context in a model-based simulation (initial conditions and parameter values) only concerns the initial state of the simulation: The initial conditions $y(0)$ specify the initial state of the model, and the parameter (a and b) values generally are not modified during the simulation. There is no “unpredicted event” during an experiment and “browsing” a model is exploiting its mathematical properties for predicting variables’ evolution (number and stability of steady states, self-oscillations, exponentially decreasing curve, etc.) for different sets of parameter values that verify some constraints, such as the condition to have an unstable steady state.

At this quantitative level, model-based simulation is used to find the best set of parameter values and initial conditions describing a set of real-world observations (generally by optimization methods). Here, the formal model is used for the prediction of any behavior of the real system in other contexts, assimilating this context to constraints and initial conditions. In a model-based simulation, the working context describes the initial state only, while variables evolve during the entire model-based simulation.

At the qualitative level, the conceptual simulation is often used to refer to a type of everyday reasoning strategy commonly called “look ahead” reasoning [36] or “what if ” reasoning [43]. Scientists use conceptual simulation in situations of informational uncertainty to make inferences from their data using a process of alignment by similarity detection. The conceptual simulation obeys the cycle of hypothesis–conceptual simulation–alignment. New representations are generated by reference to a familiar situation and by taking what is known and transforming it to generate a future state of a system [20]. The process occurs at the conceptual level and it involves mentally playing out, or “running,” a model of the visualized situation, so that changes can be inspected. Thus, conceptual simulation is a form of model building, which is likely to occur when no easily accessible, existing source for analogy is available [43].

The CxG-based simulation covers quantitative and qualitative aspects [13]. In Contextual-Graphs formalism, we make a parallel between practices and models, except that the structure of the practice is not known initially but built progressively with its development. A CxG_Simulator develops a practice by instantiating contextual elements that are encountered on the followed path (the equivalent of the initial conditions in a model-based simulation) and thereby choosing the path corresponding to a practice. Thus, a CxG-based simulation is based at the tactical level (or qualitative level) on a contextual graph, which is an experience base containing all the practices developed for realizing a given task modeled, and, at an operational level (or the quantitative level), on a particular practice of the contextual graph in the working context at hand. Time dependency appears because (1) an unpredicted event may modify the instantiation of a contextual element and thus the CxG-based simulation itself, and (2) the execution of an action may impact practice development in different ways independently of what is involved by the execution of the action.

At the quantitative level, a CxG simulator needs to know the instantiations of the contextual elements that will be used on the path of the contextual meta-graph corresponding to the developed practice and the effects of action execution. The execution of an action may modify the instantiation of a contextual element. The change in the working context leads the simulator to consider another practice with different consequences: halting the simulation (e.g. the required resource is no longer available), restarting the simulation in the new working context, repeating a routine action in the practice development, etc. The instantiation of a contextual element not yet reached during the current practice development will impact the simulation only once the simulator arrives at this contextual element.

A model-based simulation is a top-down (deductive) model, while a CxG-based simulation corresponds to a bottom-up (inductive) model. In a model-based simulation, the whole working context is defined at the start of the simulation and stays constant during the simulation, while in a CxG-based simulation, the working context evolves during practice development. A formal model is given initially (its structure is compared to observations), while a practice (the contextualized model of a task realization) is built progressively from the contextual graph and evolves with its working context. In that sense, a CxG-based simulation is a particular type of simulation. The behavior of a CxG simulator is comparable with the usual model-based simulator’s behavior, supposing that (1) contextual elements in the contextual graph can be compared to the parameters in the formal model (a change of parameter values impacts the model behavior as a change of instantiation modifies the practice developed), and (2) variables in a model-based simulation are related to the result of the progressive building of the practice corresponding to the working context.

Table 1 gives a comparison of model-based simulation and CxG-based simulation according to seven characteristics [13].

	Model-based	CxG-based
Goal	Represent a real system	Represent a task realization on the real system
Real system	An internal viewpoint	An external viewpoint
Tactical level	A model structure	A graph of model structures (practices)
Operational level	Simulation from an initial state	Simulation and building of a context-specific model
Working context	Initial values of variables and parameters (constant during the simulation)	Contextual elements and instantiations (may vary during the simulation)
Simulation	Evolution of the variables in the model	Building and use of a model specific of the working context with practice development
Type of support	Prediction, interpretation of deviation (real-system centered)	Task realization on the real system (use-centered)

Table 1. *Comparison of model- and CxG-based simulations*

A CxG-based simulation is based on a group-activity model (e.g. like in Figure 4 in the next subsection) and its shared context for describing the development of a group practice. A CxG-based simulation corresponds to the progressive development of a group practice according to the status of the shared context at each step (i.e. the instantiations of the contextual elements). Unlike formal models used in classical simulation, a context-based model has a static expression as a contextual meta-graph (i.e. the set of all the practices already used for developing the group activity) at a tactical level and a dynamic expression with the building and development of the "best" practice corresponding to the specific context at hand (i.e. the series of turns obtained by the cyclic traversal of the contextual meta-graph) at an operational level.

A practice that is developed from the working context at hand is the "best practice" because the practice is built at the same time it is used, thus taking into account all that occurs during this process. During a CxG-based simulation, the instantiation of contextual elements may be altered by either an external event or an internal event. The external event corresponds to an unpredicted event, i.e. not represented in the contextual graph (e.g. an external resource stops being available). An internal event occurs as the result of an action execution. An action (or an activity) is executed at the operational level (e.g. execution of an external program or a service). The way in which an action is executed matters at the operational level, but consequences impact the development of the practice at the tactical level (e.g. a change of practice).

3.1.2. *The TACTIC project*

The TACTIC project [31] aims at the migration of an existing battlefield simulator's interface from PC ("click-simulation") to tactile devices ("touch-simulation"). Command & Control systems (C2 systems) provide an operator with a view on an operational situation, typically a map with symbols representing military units. The operator interacts with a simulator for realizing different tasks like "Give an order of recognition" that is modeled as a contextual graph in Figure 4. This task has three actions and four activities, including two occurrence of the activity "Manage a unit" (ovals 62 and 67). The system gathers information from various sources and allows operators to interact and give orders directly. C2 systems also are used, or are being adopted, in non-military areas, for example in civil safety or by large private operators.

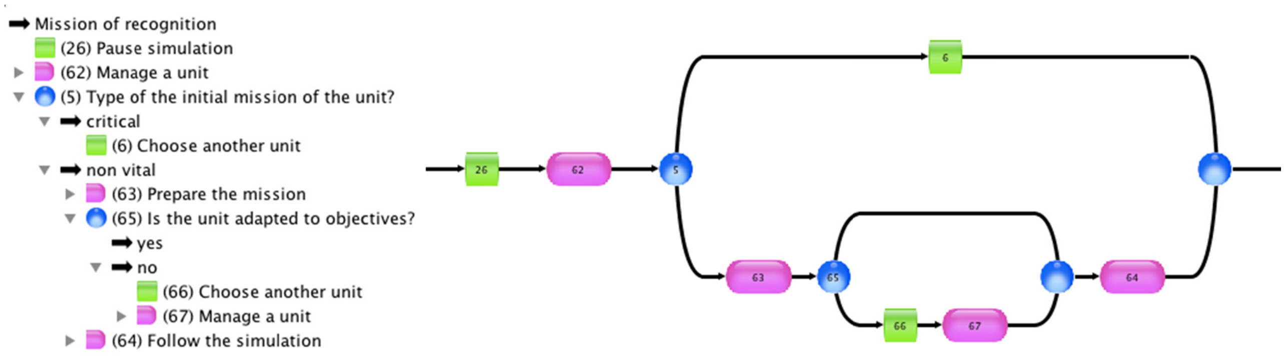

Figure 4. *Model of the task realization "Give an order of recognition" in the CxG formalism*

Such C2 systems have three complementary sources of information commonly used together: spatial coordinates of objects (the field map), temporal coordinates (the chronology) and socio-technical coordinates (ODB) [31]. The three sources of information constitute a cognitive 3-dimensional referential in which events take place in an evolving context. A unit is associated with a knowledge network with close information, such as the life-bar, and other more distant information such as belonging to an automaton. Such a knowledge network should allow different views adapted to the desired level of aggregation of the context.

Figure 5 represents the modeling of the activity "Manage a unit" on Figure 4 as a contextual graph and its legend. 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.

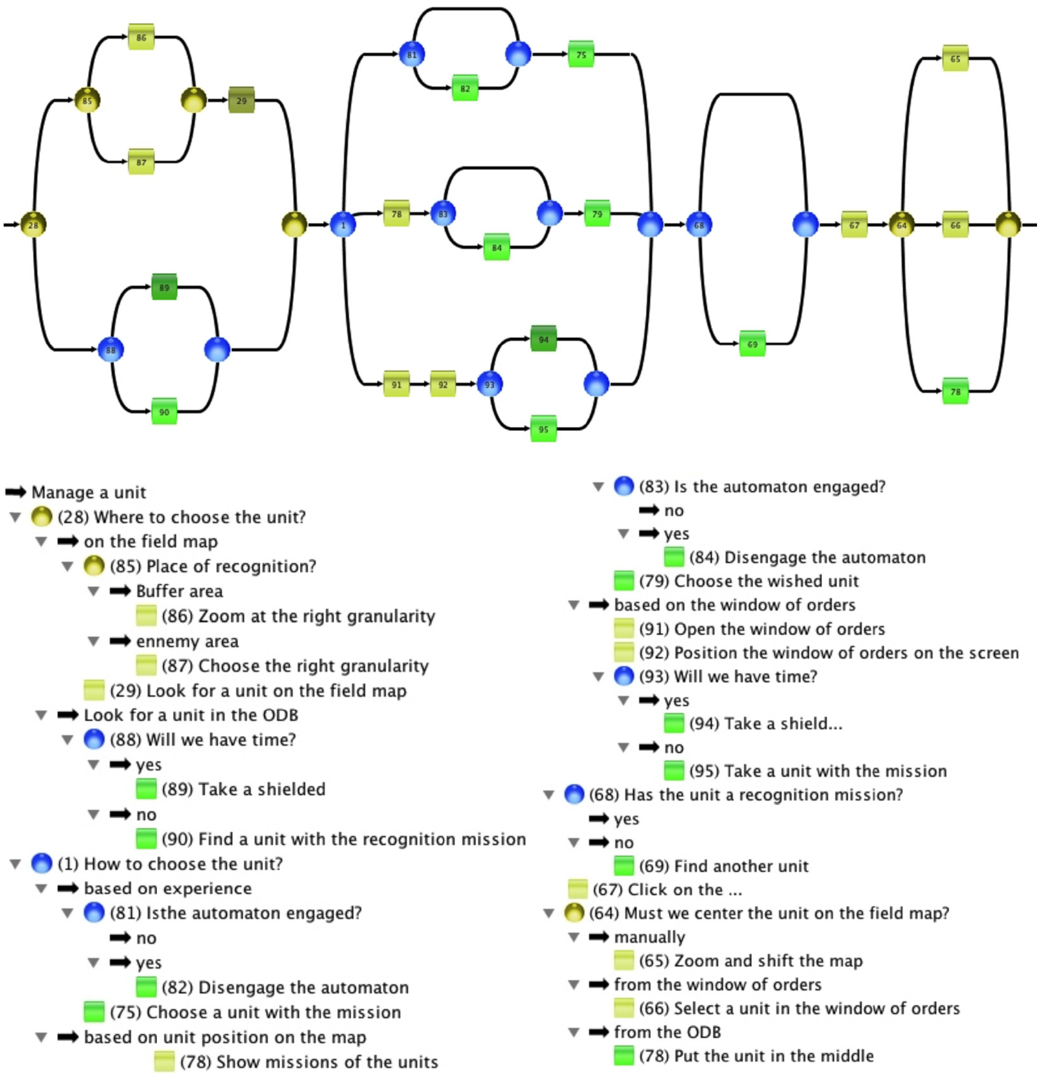

Figure 5. *The contextual graph of the activity « Manage a unit » in Figure 4*

This modeling is made according to the operator’s viewpoint. As a consequence, a part of the operator’s reasoning (in yellow in Figure 5) expresses a compilation of elementary actions realized by the operator and the simulator through the interface. Items relative to tasks keep their initial colors (i.e. green square boxes for the domain actions and blue circles for contextual elements). Items for the shared-context management are in yellow. Actions in pink (both in operator and simulator activities) are interface actions as shown by the action 19 “Click on the chosen unit” in Figure 6. For example, “Show mission orders” implies (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).

Thus the simulator has to be considered as a group member in this activity. Items in yellow in Figure 5 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 Figure 5) and green actions that are specific of the operator, like “Find a unit with the recognition mission” (Action 90 on the left part in Figure 5). This confusion of roles in the modeling of a task realization leads to a representation that is difficult to interpret. Operator-Simulator interaction goes through an interface (the place of cognitive interaction with the operator) and the screen (the place of physical interaction for the visualization of the simulation). Thus, operators must deal with an interpretation of the domain (the simulation resulting of the operator-simulator interaction) intertwined with an interpretation of the interface functioning for translating actions on the simulation into commands to the interface. For example, by clicking on the pause button, the operator thinks to stop the simulation, while this action on a button of the interface sends a command to the simulator that suspends the simulation. Nevertheless, the operator associates the control of the simulation with actions on the interface that plays the role of the simulator, and operator-simulator interaction is considered secondary to interface simulation.

Two main changes simplify the development of a group activity by the operator and the simulator as represented in Figure 6 (see the contextual-graphs representation that is given Figure 7). The first change aims to discriminate the activities of the operator (left part of Figure 6) and the simulator (right part of Figure 6), which are intertwined in Figure 5. The second change consists of rewriting each activity in terms of independent tasks (the rectangles) and of the identification of simulation parameters (the yellow actions). The arrows show all the possible movement from one turn to another as given by the simulation parameters MANAGER and TASK_STATUS. MANAGER appears in the contextual meta-graph, and TASK_STATUS defines a structure of group-member activities in terms of independent tasks.

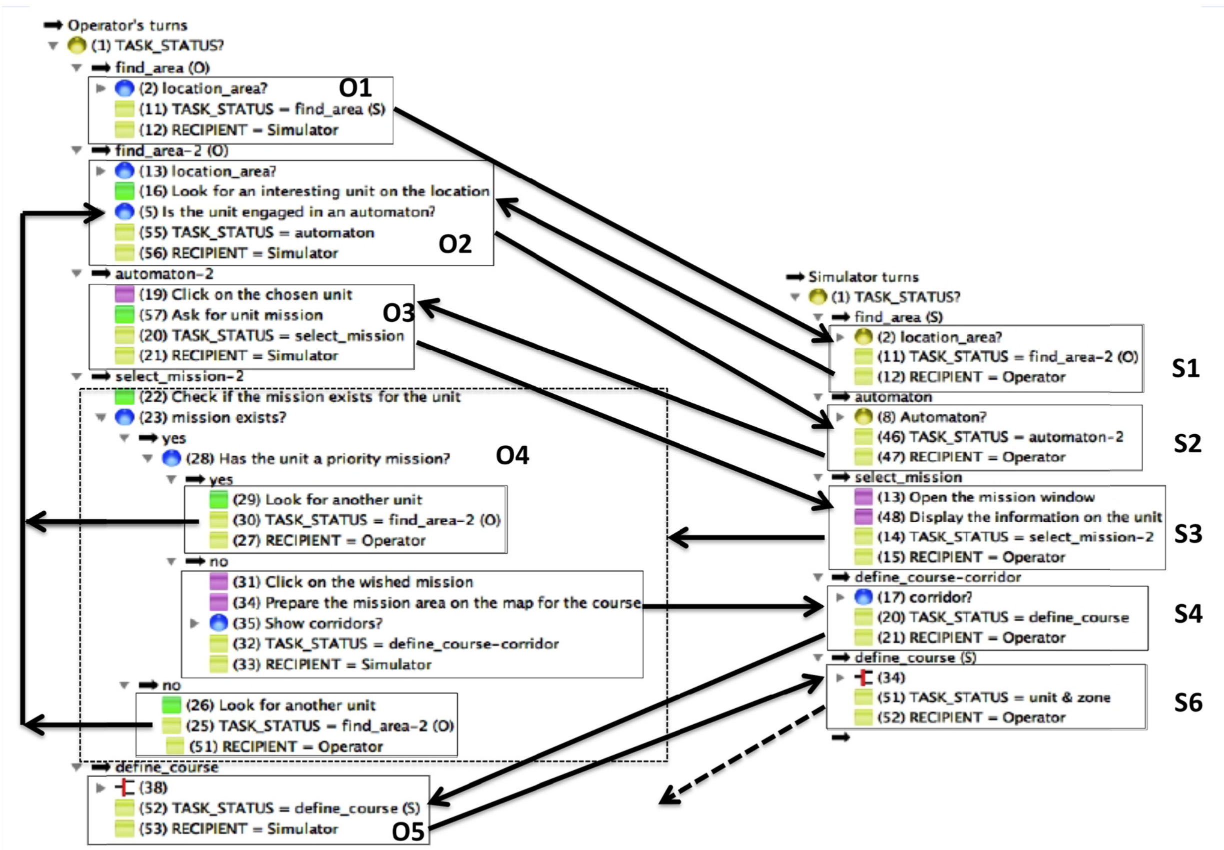

Figure 6. *Movement between turns in a simplified view where contextual elements are "closed"*

The main features of this representation are:

- A group activity is a set of actors' activities. The TACTIC project is interesting in the sense that it shows that the viewpoints of all the actors must be distinguished, even if some "actors" are software pieces. The initial operator's viewpoint (Figure 5) shows that the representation is a mixing of the operator's viewpoint and of the operator's interpretation of the simulator functioning. Figure 6, where the two viewpoints are distinguished, again shows that it would be necessary to define the interface as a third actor. The non-explicit consideration of the interface as an actor is responsible for unnecessary turns.
- Activities are represented as sets of independent tasks. A turn starts by checking the instantiations of the simulation parameters MANAGER and TASK_STATUS. This leads to focus the traversal

of the contextual graph on an independent task (represented by a rectangle on Figure 6). For example, the first turn in the operator's activity corresponds to Value (TASK_STATUS) = "find_area (O)", the rectangle O1 contains the independent task with the closed contextual element "location_area?" and the actions in yellow modifies the instantiation of TASK_STATUS and RECIPIENT for the next turn.

- Modeling nonlinear reasoning. The turn corresponding to the independent task O4 is particular in the sense that there are three possible outputs, two going towards an independent task in the operator's activity (arrows on the left toward the independent task O2 for finding a new unit for the mission) and one towards the independent task S4 in the simulator's activity to dissociate the unit that belongs to an automaton. The two arrows on the left correspond to a revision of the actor's reasoning (change of unit). Moreover, it is possible to stop a simulation by the action "RECIPIENT = "NIL".

In addition to the simulation parameters, there are other contextual elements in yellow that are instantiated in one activity and use in another one and thus are shared by group members. For example, the contextual elements "local area" and "automaton" in Figure 6 are instantiated by the operator and guide the selection of the independent tasks in the simulator's activity (contextual elements 2 and 8).

3.1.4. *Simulation of a group activity*

Managing of the group-practice development by the turn mechanism leads to a new type of simulation, namely the CxG-based simulation, in which turns are the building blocks of the model. The CxG-based simulation opens new perspectives when compared to previous types of simulation because group activity is expressed as a context-based combination of independent tasks. This allows complex developments based on the dynamical combination of a limited number of independent tasks and thus does not require a complex model, because the model of a group activity is built dynamically.

A CxG-based simulation relies, on the one hand, on the model of the group activity (the contextual meta-graph), and, on the other hand, the model of the turn mechanism at the level of the shared context that is associated with the contextual meta-graph. A CxG-based simulation corresponds to a cycle with (1) an initial shared context (i.e. a set of instantiations), (2) a turn-based development, and (3) the triggering of a new iteration of the simulation if a new turn is initialized with a new manager and a new independent task.

The CxG software has been enriched with simulation capabilities to provide the user with a simulation engine to follow the development of a group practice. The simulation engine and the user, first, progressively anticipate the contextual elements in the shared context that need to be instantiated, and, second, the simulation engine visualizes the path to follow in the contextual meta-graph. During the CxG-based simulation, the group-activity development follows a path that relies on the instantiation of the contextual elements of the shared context encountered on that path. The instantiations of the contextual elements on other branches do not matter. The simulation engine only needs the instantiation of contextual elements encountered on the path followed in the contextual meta-graph. The instantiation can be: (1) known prior the practice development, (2) obtained from the user by asking the user during simulation, or (3) obtained by the simulation engine along the followed path (e.g., the change of instantiation of a simulation parameter by an action on the path). As a consequence, a CxG-based simulation in the TACTIC project results essentially of a cooperation between the operator and the simulator, the former providing instantiations corresponding to how the operator wishes to influence the evolution of the battle, and the latter providing the resulting simulation of the group-activity development.

We discuss CxG-based simulation in this section based on its use in the TACTIC project with the example of the activity "Manage a unit" as introduced in Figure 6 in its representation in terms of independent tasks. The contextual meta-graph on Figure 7 is the graphical representation of the legend representation on Figure 6 with the representation by red rectangles of the independent tasks in the operator's activity (O1 to O5) and in the simulator's activity (S1 to S6), green and blue items correspond to the collective tasks, yellow items correspond to the simulation-parameter management, and items in violet represent actions that would be specific of the interface (see next section on this point). The movement between the independent tasks, which is represented by arrows on Figure 6, corresponds to the transition from one turn to the next one, i.e. the move across the shared context to restart a new traversal of the contextual meta-graph in the new context, each traversal of the contextual meta-graph corresponding to a turn. We will illustrate the turn mechanism by considering only the realization of the independent tasks O4 (in the operator's activity) and the independent task S4 (in the simulator's activity), i.e. not towards the independent task O2 in the operator's activity.

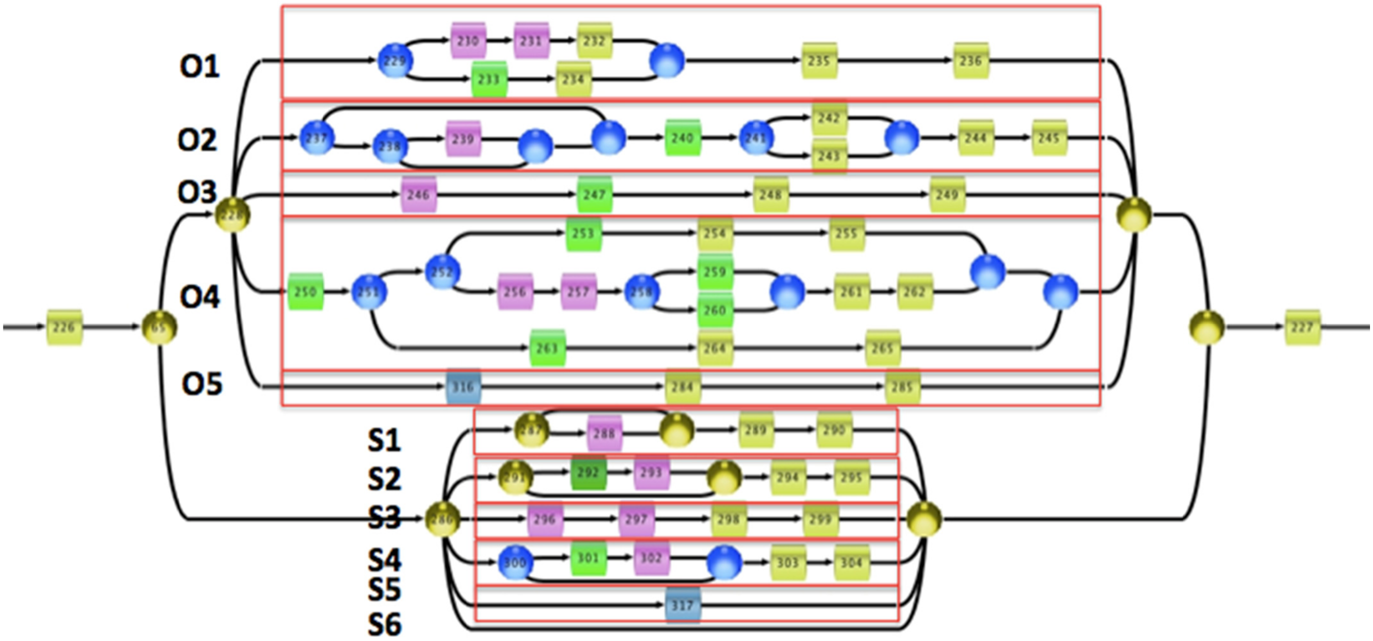

Figure 7. *CxG representation of the model with the independent tasks (Oi and Sj) presented in Figure 6*

Concretely, Figure 8 corresponds to the start of the CxG simulation for the turn O4 (see Figure 6). The shared context (the window with a red frame on Figure 8) initially contains all the contextual elements of the contextual graph because any practice is possible. The contextual elements in the shared-context window are ordered chronologically and the operator selects the instantiation of the first contextual element (MANAGER) from the popup window just after the name of the contextual element where are presented the known values. The instantiation (i.e. the choice of a value in the popup window) of the first contextual element (MANAGER) limits the focus on the branch of the contextual graph corresponding to this instantiation, and to the contextual elements on this part of the graph as shown on Figure 9.

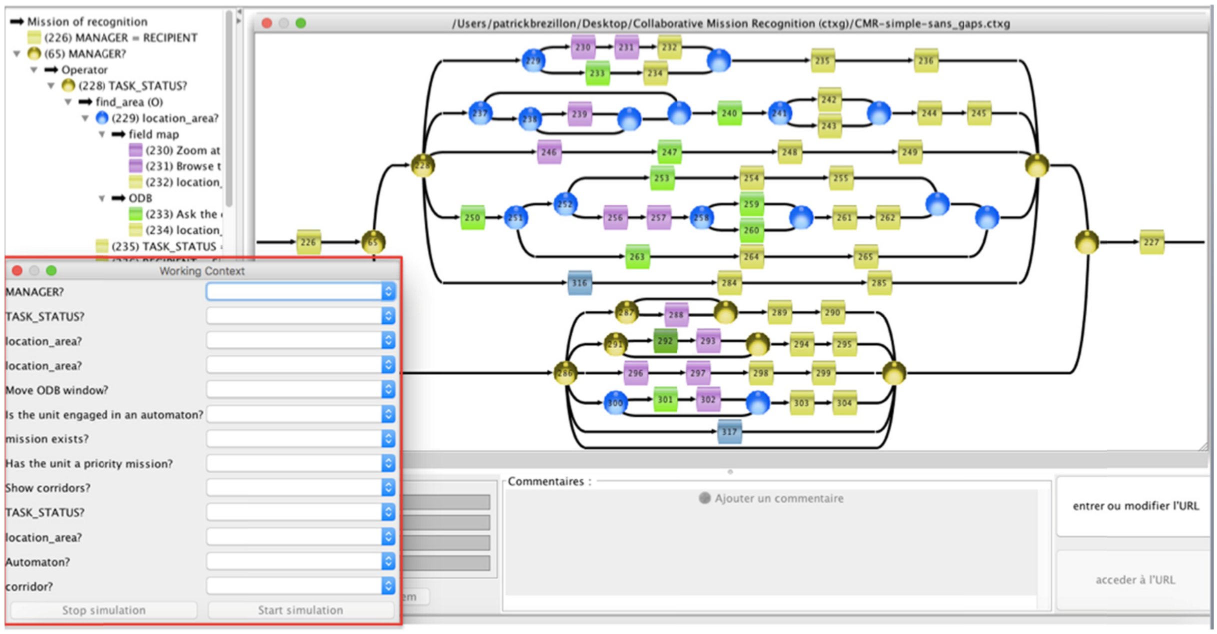

Figure 8. *Start of the CxG-based simulation of "Manage a unit" with the shared context in the red frame*

At each instantiation, the left window of the shared context is updated with the relevant contextual elements and the part of the contextual graph of interest is represented as a grey shade on the graphical window. The next instantiation of TASK_STATUS to "select mission-2" allows to enter the branch where is the independent task O4 (the grey shade also is reduced consequently to this branch). Once all the contextual elements on the path are instantiated, the simulation engine then starts the second phase of effective simulation.

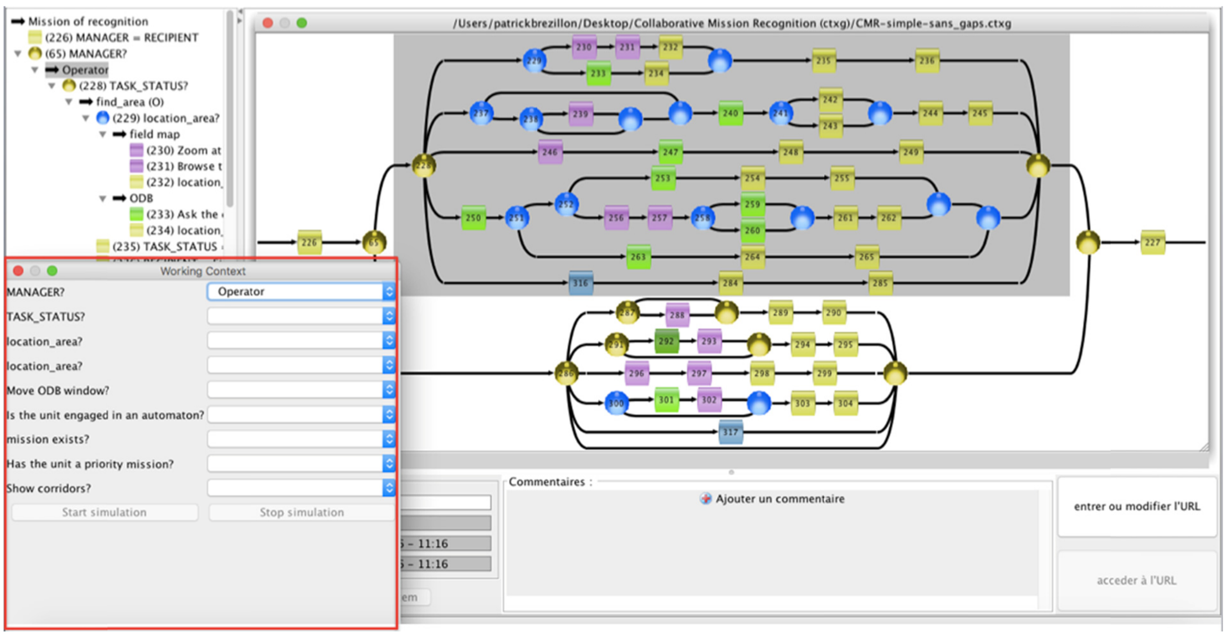

Figure 9. *Instantiations leading to the realization of the independent task O4*

Figure 10 presents the progress of the CxG-based simulation by a progressive transformation of the black color into red color of the path followed by the reasoning held during the realization of the independent task O4. Figure 10 also shows the end of the simulation as expressed by the window "Show next turn" on the right part of Figure 10, with the instantiations made for the simulation parameters with MANAGER and RECIPIENT = "simulator", SENDER = "operator" and TASK_STATUS = "define_course-corridor", which will lead to the independent task S5 in the simulator's activity if the user decides to continue the CxG-based simulation.

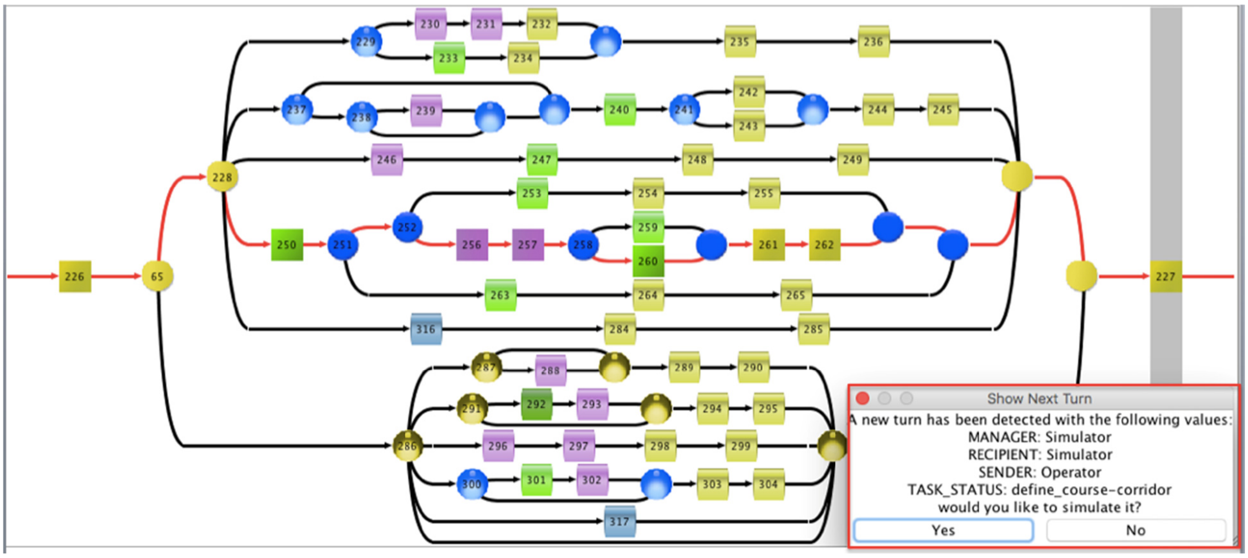

Figure 10. *CxG-based simulation of the turn S5 that follows turn O4*

The user controls the evolution of the CxG-based simulation by providing at the beginning of each turn the instantiations of the contextual elements that would be crossed at the next turn. Thus, the user can decide to replay the turn, check a turn that is different from the normal successor or even stop the CxG-based simulation. Thus, the user may try different alternatives to understand the best practice in a given context (or the impact of the context on the practice selection). That situation occurs in an application for the subway in Paris when an operator arrives in the control room and studies the practices developed by the previous operator during incident solving and compares them to alternatives he would have chosen [37]. The pre-filled information can be changed at any time, providing a flexible experience to the user, who can decide to replay or skip the simulation of a turn or change the actor at any time.

The explicit association of the shared context with the contextual meta-graph solves the main limitation of the contextual graph, namely the acyclic nature of contextual graphs that are considered without the working context. In the TACTIC-project example: (1) the turn O4 allows the operator to change his reasoning when an unexpected event is encountered (e.g. the chosen unit cannot accept the mission); and (2) the simulation parameter “RECIPIENT” = <nil> allows the turn mechanism to stop the simulation at the end of any turn, for example, when an action cannot be executed because a lack of external resources. Moreover, the turn mechanism allows the introduction of loops for managing a negotiation between actors as well as for realizing an independent task several times (e.g. the review process allocated to several reviewers in the paper submission example).

Another effect in more complex models with several actors is to let the user focus on the different types of interaction between actors and decide which interaction to promote in a given context. The strategy that is used for modeling group activity is to “divide and conquer” by decomposing the group task into “elementary” independent tasks that communicate only through the shared context. Thus, each independent task in the group activity corresponds to a step in the reasoning of an actor, and the interest of this modeling process is to focus on the interaction into group activity based on the turn mechanism. The methodology developed for a group activity would apply to an actor’s activity too.

3.2. *Local visualization by Practice-Tree view*

3.2.1. *The approach*

In many situations, people try to diagnose accurately the current state of the world beforehand to undertake an appropriate action. Decision makers reduce the complexity of a decision tree by using as much contextual information as they can and, thus, postpone actions to execute. For example, for solving an incident on a subway line, the operator makes the decision to stop a train, or wait in a station for a while according to contextual information such as the number of travellers, peak hours or not, weather, etc. What is noticeable in this type of reasoning is that the action sequence thus generated is insensitive to events (fully expanded alternative).

Although the CxG formalism has been applied successfully in several fields (e.g. medicine, transportation, business management and military operations), there are some limitations. As an experience base, a contextual graph contains an organization of all the known practices for a task realization. This structuring of practices by contextual elements provides a coherent global picture of the group activity in various contexts, but make it difficult to follow a specific practice and to know the decision that is associated with it. Often, experts need to quickly identify the right practice that matches the specific context at hand (especially in emergency situations), first, to evaluate a situation at hand in order to make immediate decision (e.g. handle an incident in the subway), and second, to infer the type of object of the reasoning corresponding to the content of a given practice. An object of the reasoning can be either a physical object (e.g. a digital image in medicine) or a cognitive object (e.g. a diagnosis).

The choice at hand in many real situations is a choice between scenarios that are sequences of decision maker's actions and changes of the state of nature. Unfortunately the examination of all possible actions and events quickly becomes computationally unmanageable. For many decision makers, the most urgent need is, on one hand, to reduce the number of events to consider and on the other hand, to undertake robust action and, if possible, macro-actions. Postponing actions at the end of a sequence of chance nodes is a way to cope with the multiplicity of events. Theoretically, this does not change the situation, but, practically, it does so because, in this way, decision makers capture as much information as they can to transform chance nodes into certainty by determining which events occurred and thus which macro-action is relevant.

The Practice-Tree view is another formalism for representing differently the one input – one output structure of contextual graphs [26]. It provides a simple way to communicate what to do in a specific context: each leaf corresponds to one practice given as the roadmap for making a decision and realizing an action. It is a reorganization of all the items of a contextual graph based on the property of series-parallel structure of the CxG formalism, thanks to the representation of contextual elements as tuples: [contextual node (the moment at which the question is asked) and a recombination node (the moment at which the subtask realization is made)]. With the Practice-Tree view, the decision maker thus can envision the sequence of actions on each leaf in term of "fully expanded alternative" [17].

Practically, the transformation of the graph into a tree relies on the following observation: an item just after a recombination node is equivalent to have this item on all the branches merging at the recombination node. This rule applies recursively as long as it is possible. Figure 11-A presents a simple contextual graph with two contextual elements (items 1 and 7) and an action 5 between them. The sequence of action 5 and contextual element 7 can be copied on the three branches of the contextual element 1 as shown on Figure 11-B. The same rule applies on each branch now of the contextual element 1 for actions 2 and 6.1 into the contextual element 7 on the upper branch, for actions, 3, 4 and 6.2 into the contextual element 12 on the middle branch, and for actions 5 and 6.3 into the contextual element 13 on the lower branch of the contextual element 1. Then, it is possible to suppress all the recombination nodes because there is no other item after. Figure 11-C shows the result as a Practice-Tree view (a formal presentation is given in [28]).

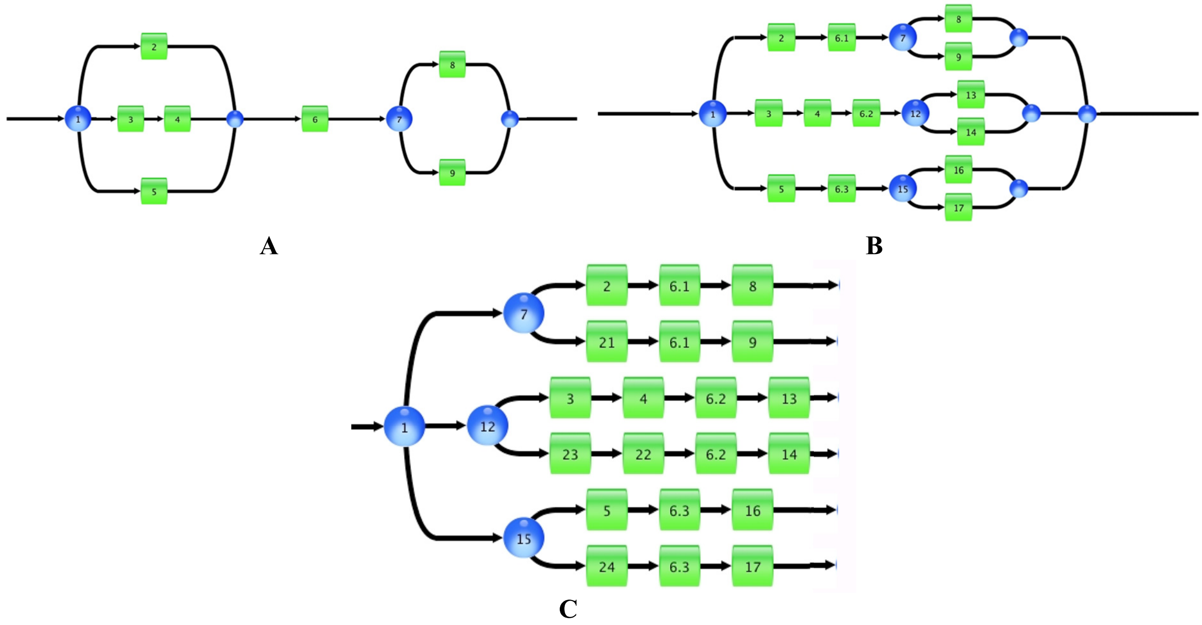

Figure 11. *Transformation of a contextual graph (A) into a Practice-Tree view (C) with an intermediary step (B)*

The tree view of contextual graphs provides clear visualization of a practice in two parts, namely, a specific context (i.e. the proceduralized context presented as an ordered sequence of instantiated contextual elements) and a sequence of actions (like a macro-action or even a workflow) to process, and the objects of the reasoning associated with. Thus, a Practice-Tree view allows to focus on the link between the contextualized situation (e.g. an image to be classified) and the conclusion (e.g. the decision made) by, first, presenting the rationality of the conclusion (the proceduralized context) and then the actions that should be performed. Figure 12 shows the example of mitosis identification [3] on Figure 12.

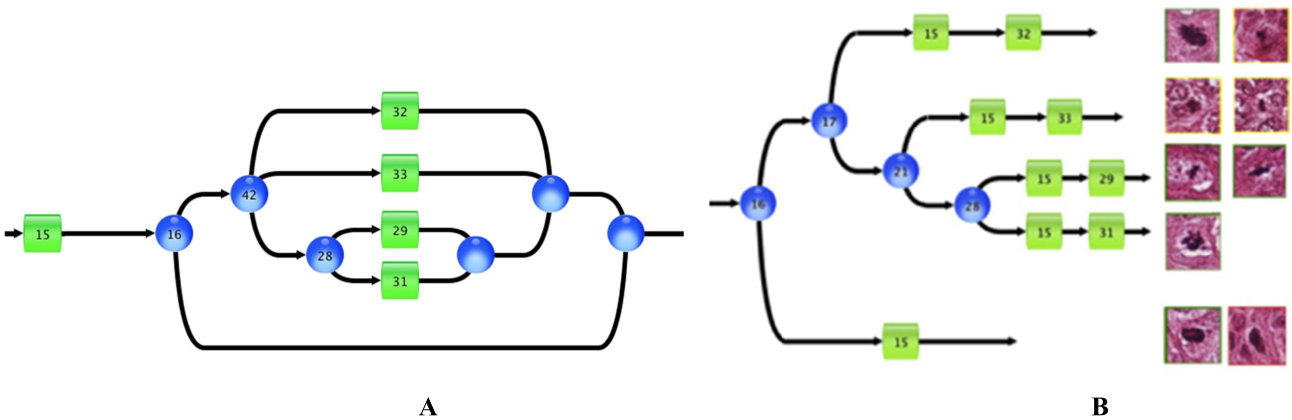

Figure 12. *Mitosis identification: graph (A) and tree (B) representations with the objects of the reasoning*

Figure 12 represents a model of mitosis identification represented as a contextual graph [3] with the initial graph representation (Figure 12-A) and the equivalent tree representation (Figure 12-B). Figure 12-B details the two parts that corresponds to the representation of an independent task as a decision-making (DM) process. It is thus possible to associated to each practice the list of the digital slides on which the practice applies. It is very important for Anatomical Pathology physicians because the Practice-Tree view can be used to organize their wall of images they refer to frequently.

By integrating the Practice-Tree view to the CxG software, we are able to show a task realization model in two different formalisms, graphs and trees. A CxG focuses on the global way a problem is solved in different contexts (i.e., the practices), providing a chronological representation of a task realization. Meanwhile, a Practice Tree focuses on the local link between the situation (e.g. an image to be classified) and the conclusion (e.g. a decision) by first presenting the rationality of the conclusion (the proceduralized context) obtained in the practice, and then the actions that should be performed.

3.2.2. *Limits of the formal approach*

The predominant structure of a series-parallel graph is something between totally series and totally parallel, and the width of the corresponding tree depends crucially of the nature of the predominant structure. Roughly speaking, the parallel structure of a graph leads to a tree with a number of leaves comparable to the number of paths in the graph. Conversely, the final number of leaves in a pure series structure is the product of the number of branches b_i in the N contextual elements, $i \in [1, N]$ in the contextual graph. As a consequence, the width of the Practice Tree could become rapidly uncontrollable if the number of sequential contextual elements (and of branches on them) is large [28]. Figure 13 illustrates the problem on two contextual graphs, one with a predominant series structure (G1) and the other with a parallel structure (G2). Both G1 and G2 structures both have 5 contextual elements and 11 actions.

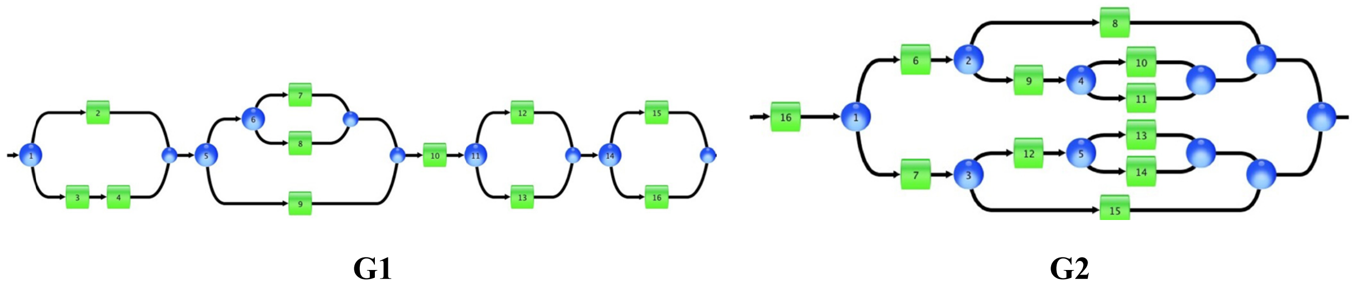

Figure 13. *The graph G1 has a predominant series structure and G2 a predominant parallel structure*

Even with the same number of items in G1 and G2, the recursive application of the previous rule for transforming the contextual graphs G1 and G2 into Practice Trees T1 and T2 respectively leads to different numbers of leaves for G1 and G2 as Figure 14 shows: T1 has 24 leaves while T2 has 6 leaves. Thus, T1 is four times greater than the width of the Practice Tree T2.

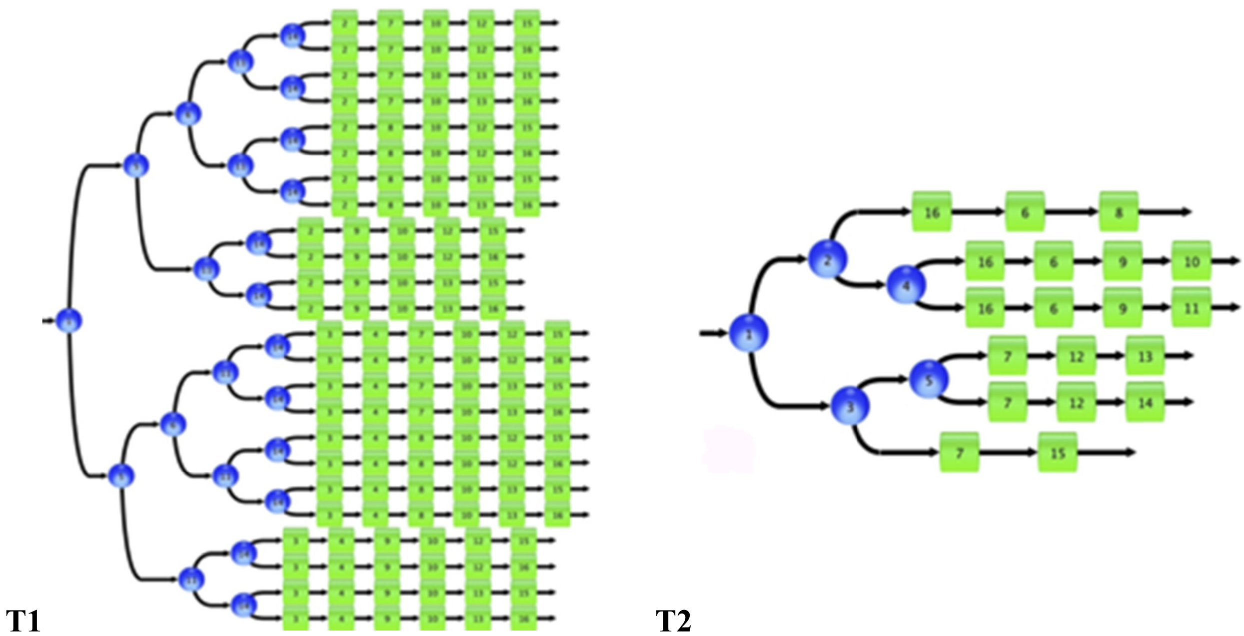

Figure 14. *Practice Tree T1 and T2 correspond to the contextual graphs G1 and G2 on Figure 13.*

In real-world applications, contextual graphs have complex structures generally with more than 200 contextual elements and 300 other items (actions, ESIA and activities), and the approach would lead theoretically to a Practice-Tree view with a magnitude of 8000 leaves (i.e. practices) or more. Thus the predominant structure of a contextual graph could be a crucial factor for the interest of a Practice-Tree view, because a predominant series structure leads to a wide Practice Tree that could be heavy to compute and useless for the user, because Practice Trees are valuable only if they can be quickly computed and easily explored. A Practice Tree that is too bulky is opposed to the objective of this view representation (i.e. a fast focus on a precise practice contrasted with the others).

However, the theoretical problem of tree width does not have the same importance in real-world applications because models come more from experts than novices. This claim relies on a study of the CxG models developed, on the one hand, by experts in a number of real-world applications, and, on the other hand, by Master students in numerous simple models made in a course on Context Management. We thus have a collection of about 200 contextual graphs built by users with different levels of experience (i.e., novice, experts, and intermediate). Models in the CxG formalism have a semantic that makes an important difference with other series-parallel structures because contextual graphs have a structure that is more parallel (i.e. expert graphs) than series (i.e. novice graphs). As a consequence, the Practice-Tree view associated with a contextual graph has a reasonable size because most of the contextual elements are interdependent. For example, one looks for the instantiation of the contextual element "Which protection against rain?" (umbrella or raincoat) only if the contextual element "What is the weather?" is instantiated to "rainy", not "sunny".

The model of the task "Give an order of recognition" (Figure 4) in the TACTIC project was elaborated by knowledge acquisition techniques from a panel of ten experts that used routinely the simulator for different purposes, but they were not the real end-users (military people). We ask each expert to represent in a mind map the knowledge they used for solving the same task "Give an order of recognition", first, in a general way and, second, in a specific simulation (see [31]). In the framework of the Practice-Tree view, we present in this paper a strategic expert (a holistic view of the software), a tactical expert (a simulator trainer) and an operational expert (a developer). In the following three figures, the depth of the representations (number of levels) corresponds to the number and details of the contextual elements that are considered. Grey items correspond to the items not considered for the specific scenario on which they had to work.

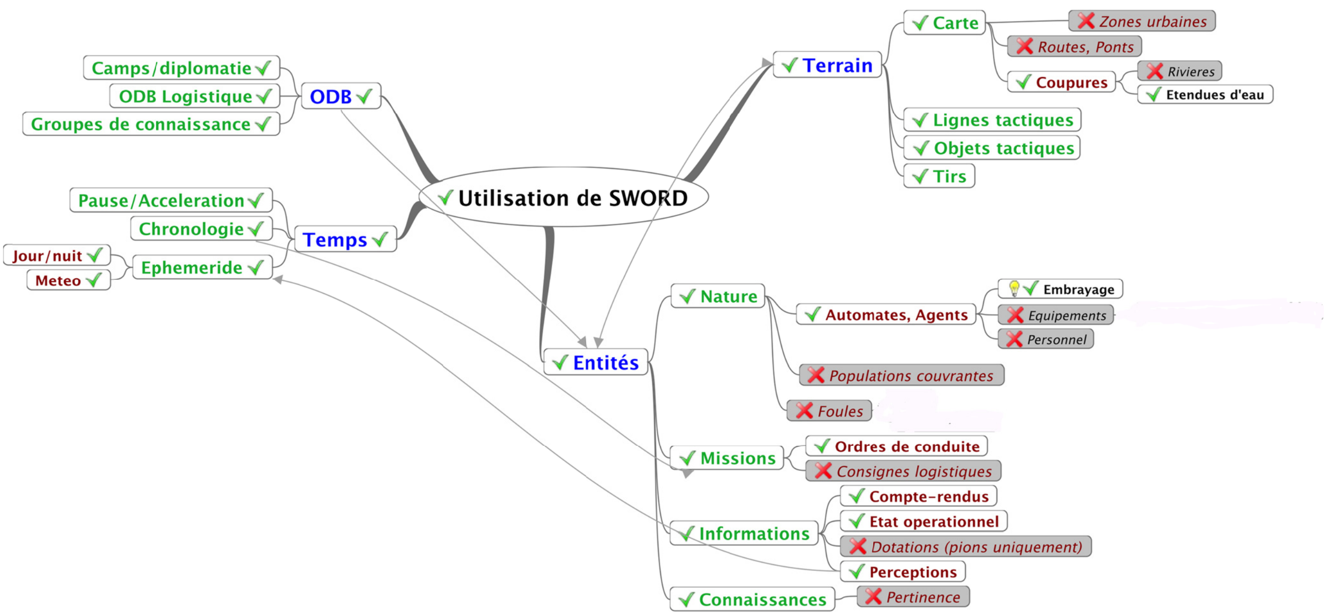

Figure 15. *Knowledge organization in a tactical view of the mission on Figure 4.*

Figure 15 shows the knowledge organization of the tactical expert, which is very close of the operator interacting with the simulator (the end-user). The focus is limited to the given mission and leaves all other problems at the periphery of his focus. The expert also introduces links between leaves of the mind map to express his strong compilation of the contextual elements (e.g. the item “perceptions” is linked to the item “Ephemeris” because the useful information on weather or day/night only address the question of the soldier’s vision around him. Such links corresponds in the contextual-graph model to establish a dependency between contextual elements (a compilation) and thus a parallel structure.

Figure 16 shows how the operational expert--one of the modelers of the simulator—organizes his knowledge in depth for analyzing what the execution of an action on the simulator is, not for analyzing what the action is on the battlefield (i.e. on the simulation). With respect to the operator, the operational expert is a « smart novice ».

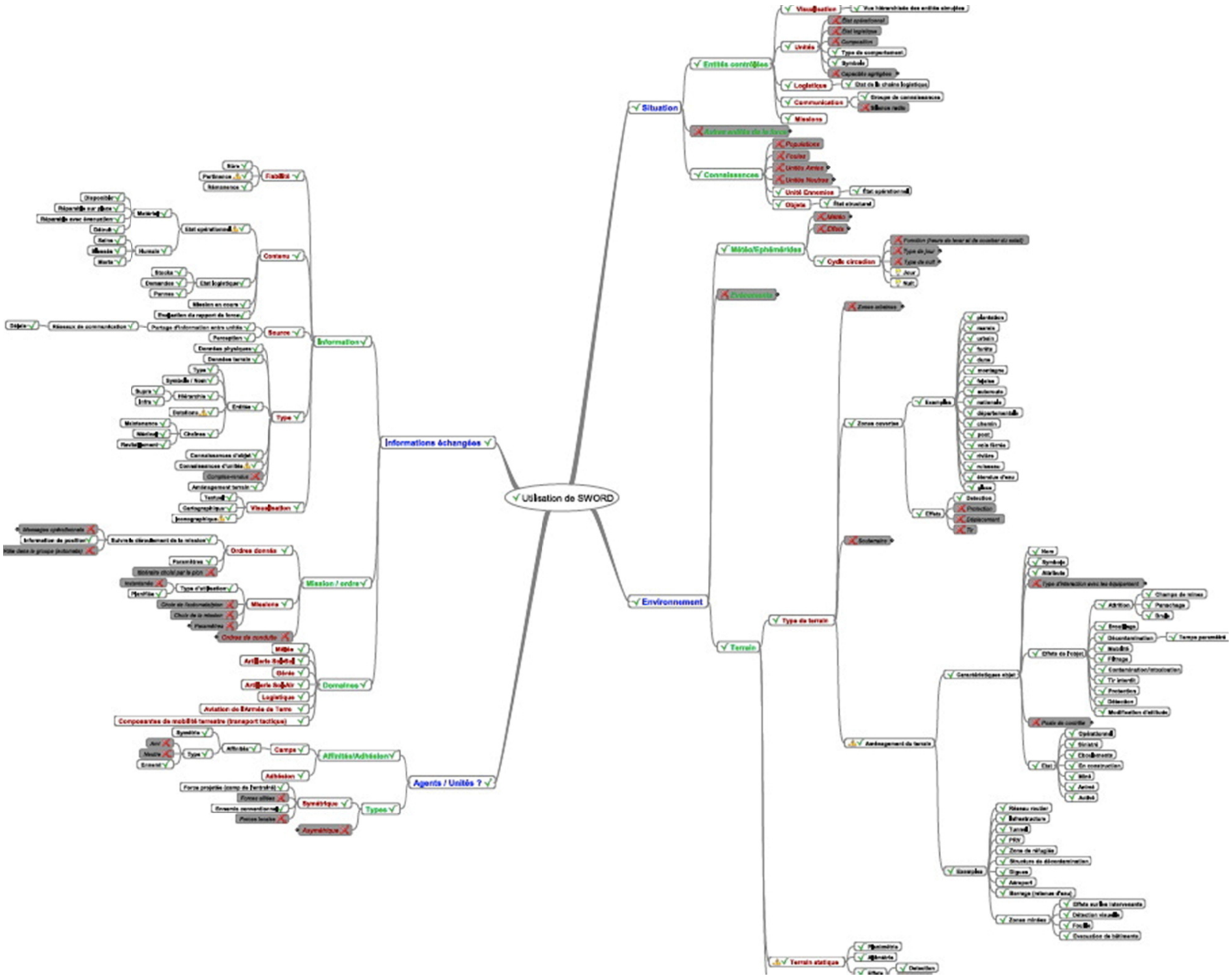

Figure 16. *Knowledge organization in an operational view of the mission on Figure 4.*

In Figure 17, the expert map of the strategic expert is more developed than the tactical expert’s map because the focus of this expert is not on the realization of the task at hand, but includes the possibility to deal, on the one hand, with a more decontextualized view of the problem (i.e. to keep in mind different contexts), and, on the other hand, to rely the specific task on other strategic problems, for example, taking into account other potential tasks that may interact with the problem at hand. At the limit, the details on the task realization are not interesting for the strategic expert. In some way, the strategic expert prunes at a high level to find more quickly what the real focus is.

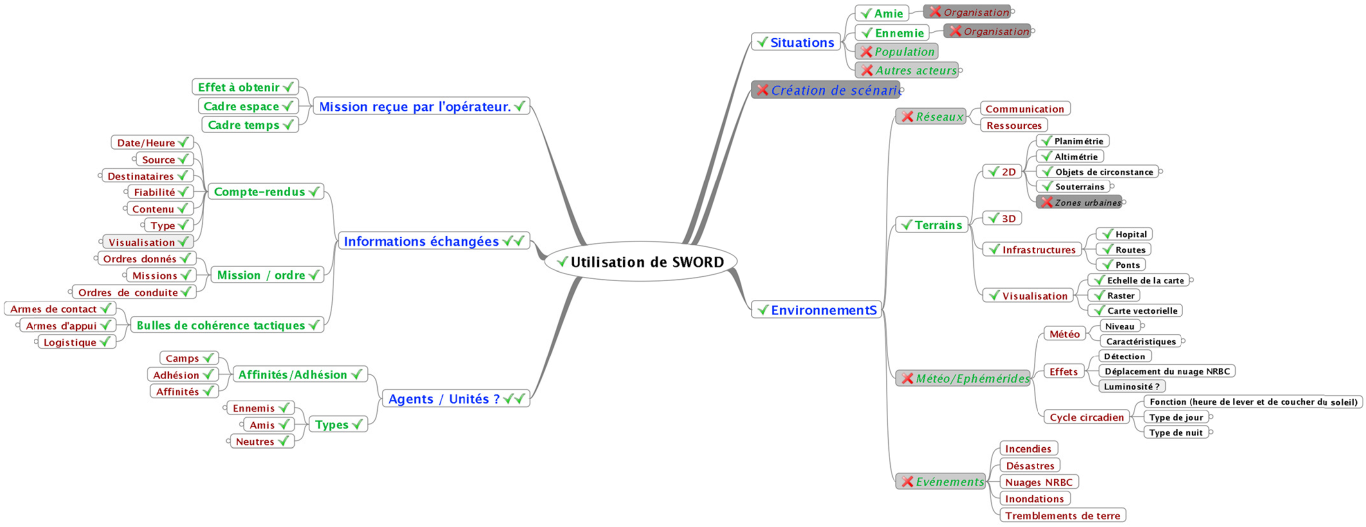

Figure 17. *Knowledge organization in a strategic view of the mission on Figure 4.*

Now if we compare these expert maps to the Practice Tree representations presented in Figure 14, we can associate a series structure with a large number of leaves to the novice view (i.e. the operational expert and to some extend the strategic expert), while the parallel structure with a small number of leaves corresponds to the expert view (i.e. the tactical view).

The main observations based on our experience in the TACTIC project are the following:

- Experts tend to create contextual graphs with a predominant parallel structure while novices are more prone to model their problems by developing contextual graphs with a predominant series structure. The main reason is that expert reasons on the basis of their experience (i.e. structured knowledge), when novices assemble on the fly “atomized” knowledge pieces.
- The quality of the task-realization model depends directly on the user’s knowledge about the task realization, the situation and the local environment. A weak user’s experience implies a narrow understanding of the problem, while an expert does not see the problem as a problem.
- The modeling of the task realization depends on the user’s experience. The expert thinks in an organized and structured way, avoids assumptions leading to unreal scenarios, and identifies actions that can be performed in parallel. The novice executes actions sequentially without paying attention to the logic of the task. The novice may add sub-tasks, or change the order of the subtasks, while the expert applies the general scenario of (1) collecting first contextual information and (2) applying the better strategy in the proceduralized context found.
- A user’s reasoning depends on the user’s experience. The expert has a holistic view of the task realization based on his experience and discriminates relevant and useless knowledge pieces. Thus, the expert makes “cognitive simulation” to look ahead in the task realization. The novice has a fragmented view of the task realization and, thus, has a reactive reasoning discovering the task when realizing it, sometimes missing some details, considering irrelevant aspects or establishing no links between the different steps. This is in line with Feltovitch et al.’s work on how doctors change their diagnostic behavior as they gain expertise [45]

The lesson learned here is that the Practice-Tree view is well adapted for representing experts’ reasoning (i.e., resulting in few branches). For example, the operator responsible of a subway line in Paris manages an incident with the train driver. The operator identifies the complete situation (i.e. the proceduralized context) and gives the roadmap for solving the incident to the train driver (i.e. the procedure to follow in this specific context). The combination of the contextual graph and the Practice-Tree view offers an efficient support for the training of novices.

3.2.3. *Practice-Tree view for group activity*

Figure 18 revisits the modeling of the task “Manage a unit” considered previously for illustrating the TACTIC Application. Figure 18-A is the tree representation of the contextual-graph legends presented on Figure 6 (without the arrows) and Figure 18-B is the tree representation of the contextual graph on Figure 7. The shade legend “select_mission-2” on Figure 18-A corresponds to the independent task O4 that is discussed previously. The branches of the contextual elements on Figure 18-A are not represented for simplifying the figure (and thus actions are not represented).

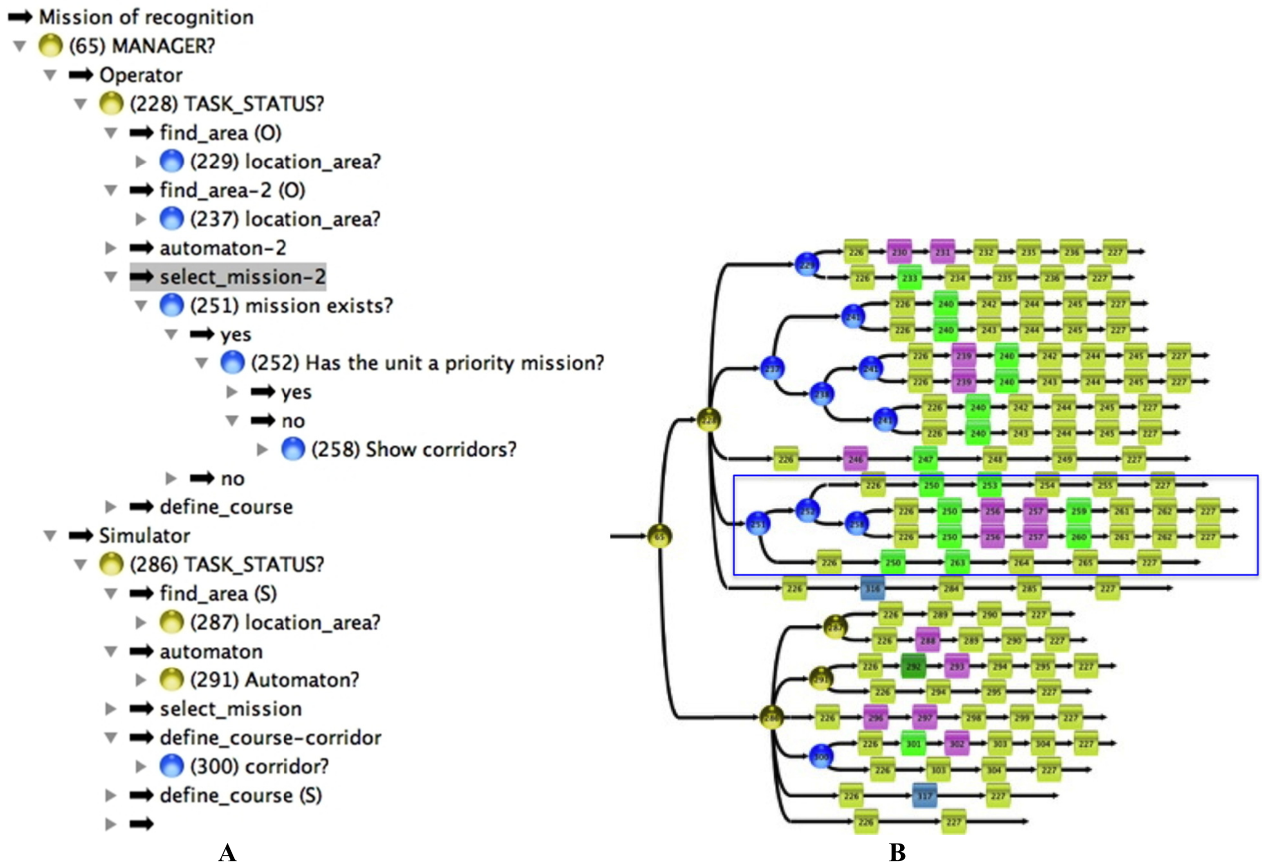

Figure 18. *The tree representation of the task “Manage a unit”*

A CxG representation based on independent tasks (Figure 18-A) is very close to the corresponding Practice-Tree view (Figure 18-B), although the objectives of the two representations differ: the CxG representation proposes a global visualization of the group activity at the tactical level with all the practices and their interdependencies in terms of independent task, while the tree representation makes explicit the details of each practice in a local visualization of the group activity at the operational level. For instance, the independent task O4 (global view) contains four practices (local view in the rectangle on Figure 18-B). This corresponds to what we call a glocal view [3]. Thus, we can follow the group activity at two levels, at the tactical level by following group-members’ interactions, and, at the implementation level, from the details of each intervention.

3.3. *Glocal visualization*

Brézillon et al. [15] define the term “glocal” as an association of global and local standards. The glocal search associates a phase of superficial exploration of a digital slide for finding a zone of interest with a phase of fine-grained analysis by zooming on the zone of interest. In the FlexMIm project [3], anatomo-cyto-pathologists follow procedures that include a gross examination (visual examination of organs), dissection and sampling of surgical specimens according to standardized protocols and then a microscopic examination of stained tissue sections. They also use techniques such as immuno-histochemistry, cytogenetic and molecular biology to identify abnormalities in cells or tissues. They now work from digital slides playing at two different levels by frequent switches between a low magnification (to identify a zone of interest on the slide) and high magnification (to evaluate relevant features according to criteria in the zone of interest). Practices are developed for the gross examination and others for the microscopic examination. The former are relatively standardized (e.g. systematic exploration of the image), but the latter rely on criteria for making a decision that are subjectively evaluated and the object of a voting system. The two types of reasoning (global and local reasoning) have different natures because they are at two different levels. However, “glocal reasoning” may have another meaning if one considers the local reasoning as a specialization (or contextualization) of the global reasoning in the specific context. (For example, international enterprises have a global strategy that is applied locally by an adaptation to local rules.)

In the TACTIC project, the reasoning of the operator is led between a “depth-first” strategy (the local reasoning) and a “breadth-first” strategy (the global reasoning). The "depth-first" strategy goes to the finest possible granularity on a line of reasoning in order to identify the current status of a situation and to anticipate the course of events. It assumes that we know what to do and how to get there quickly. This strategy allows studying the technical feasibility of an approach as well as the needs in terms of resources, and, in a second step, gradually expands this approach. Figure 15 shows this type of reasoning in an expert map. Conversely, the breadth-first strategy is applied when it is necessary to consider all possible situations first (e.g. to be aware of a unexpected event in another area of the battlefield). The breadth-first strategy is observed in expert maps of operators evolving at a strategic decisional level that maintain important contextual elements, even if not directly necessary in the realization of their tasks. Figures 16 and 17 present two expert maps. For example, "environment" was considered only as a part of "situation", even if environment is the main source of contextual elements on the battlefield map. The reason is that the operator considers environment only through what is needed in the task realization, that is, operational knowledge that is required. The operator thus is interested by an external event more through its effect on his task rather than by the origin of this event.

The glocal approach leads to two types of enrichment of the experience base: (1) by addition of a new independent task (at the global level); and (2) by addition of a new practice in an independent graph (at the local level). The first type corresponds to learning by accommodation by which the mental representation of the user is enriched by a new independent task. The second type corresponds to learning by assimilation in which a new practice is added to an independent task because a more specific context has been encountered. Compared to an expert, a user may be a novice either by lack of knowledge on independent tasks to realize (or often on the way to combine these independent tasks) or by lack of sensitivity of context and specific practices inside an independent task.

With the introduction of the notion of independent task, the graph and tree representations are similar, with a reduced number of possibilities. For example, the tree representation of the task “Manage a unit” of a normal contextual graph like on Figure 5 possesses 144 leaves, while the number of leaves is 23 in the revised formalism on Figure 18-B. Making the simulator explicit in the group activity allows a clear representation of the effective practices. This implies that even an activity led by a unique actor will benefit by a representation based on independent tasks.

4. Conclusion

In organizational settings, groups do not work in isolation like groups in a laboratory setting. These groups need to manage their boundaries and adapt to their organizational environment. As a consequence, the organization is a contextual variable influencing group behavior [29], and the group is a contextual variable affecting individual reactions [9]. The organization of an enterprise often is described in terms of roles and tasks associated with roles. Brézillon et al. [16] pointed out that a collaborative activity begins by building the shared context in which an element is introduced by an actor and accepted by the others; that is, integrated more or less directly in their mental representations (see [16] for an application on the collaborative construction of an answer to a question). Building a shared context corresponds to the first step of a collaborative process of decision-making [19]. An important lesson here is the need to follow a "task-realization oriented" approach rather than a simple "user-oriented" approach, thanks to making context explicit. In the TACTIC project, we study the possibility of mapping between the actor's mental map and the simulator's "mental map". The situation seems relatively simple in battlefield simulation because it was possible, on the one hand, to help the operator to express their mental model as an expert map, and, on the other hand, to identify the mental map of the simulator as three databases. However, it is not sure that this will be as straightforward in other applications (e.g. between human actors).

The set of changes in our initial conceptual framework ensures a logical shift of our paradigm of Contextual Graphs from the representation of an actor activity to a group activity. Indeed, a group activity is expressed as the cyclic traversal of the contextual meta-graph in different contexts, while the realization of a task by an actor corresponded to a unique traversal of the contextual graph for realizing a whole task. As a consequence, the actor's viewpoint stays at the level of member activity, as previously, but now there is the viewpoint of a group as an entity at the level of the contextual meta-graph. The notion of activity at the cognitive level is translated in terms of task realization—and not of task model—with a representation in terms of independent (sub)tasks at the level of each member activity. Making these two levels explicit brings an explanation to the opposition of the Engineering viewpoint (a set of discrete contexts) and of the Cognitive viewpoint (a unique evolving context) [14]. Thus, the notion of group activity takes a more operational meaning than the notion of collaborative work because it relies on a model of task realization and shared context. The notion of shared context is central of any real operational modeling of a group activity. Such a modeling is possible only in a formalism that allows a uniform representation of the elements of knowledge, reasoning and contexts. The contextual-graphs formalism is one of these formalisms. It was limited to a unique actor, but has been extended for a collaborative-work model.

Other key elements of our approach are the notion of turn mechanism to model interaction during the deployment of a group activity. Indeed, group-member interaction is expressed as the cyclic traversal of the contextual meta-graph by reapplying the model of the group activity in different contexts, which are attached to turns. The working context is attached to the task realization of the turn at hand and the shared context to the group activity. The CxG formalism allows the representation of any type of interaction between group members, and even its evolution along the task realization: what is collaboration at one moment may become cooperation at another one.

We propose a modeling of group activity at two levels. At a conceptual level, group activity is modeled as a sequence of interactions between the group members that is expressed as a sequence of turns successively taken in the group-members' activities, and the shared context is the medium of group-member interaction. Successively, each member takes the lead of a turn and the group activity thus corresponds to a sequence of turns that is built dynamically. At the implementation level, a group activity is represented as a contextual meta-graph composed of members' activities that are designed as independent tasks managed by a turn mechanism relying on the shared context.

A group practice is simultaneously built and developed by the dynamic association of task realization and interaction management. The contextual meta-graph (i.e. the model of the group practice) is structured by simulation parameters that, first, manage member interactions through the selection of the temporary manager of the group activity, and, second, build the turn sequence by a cyclic traversal of the contextual meta-graph and the shared context. Thus, a group activity is simultaneously built and developed by combining task realization and interaction management.

Although it was not the object of this paper, it appears from this study that the lack of consensus on the terms of collaboration and cooperation comes from the fact that (1) These terms evolve dynamically during interaction; (2) A group activity must take into account, upstream, the setup of the group activity and, downstream, the use of its results; (3) The type of turn management depends essentially of the type of group activity; and (4) The shared context plays a central role in a group activity. To be fully operational, an actor must know the request and the context in which this request is made. The more the shared context can be developed, the more efficient will be the decision group.

However, the most challenging perspective concerns the introduction of the CxG-based simulation as a tool for modeling group activity. Indeed, it looks difficult to model group activity without such a tool, which associates the cyclic traversal of a contextual meta-graph and its shared context, keeping the focus on the manager's activity at any moment without loss of the whole picture of the group activity, thanks to the possibility to associate the context-based evolution of the group activity with the interaction among the group members for simulating.

The revised CxG formalism was elaborated for modeling group activity (and thus group-member interactions). However, the revised CxG formalism also can be used for modeling an actor activity. Based on the notion of independent task and simulation parameter, the revised CxG formalism leads (1) modeling actor's activity in terms of independent tasks (becoming the building blocks of the modeling), (2) introducing the cyclic use of the traversal of the contextual graph, and (3) making explicit the relationship of the working context in the actor activity. As a consequence, this new approach naturally allows the introduction of notions such as loop, stop, reasoning revision, etc. in the modeling actor's activity, and does not rely on ad hoc functionalities of the representation formalism. Indeed, the extended formalism can be applied for modeling an actor activity because independent tasks are not necessarily sequential (see a study applying the CxG formalism for workflow management in [20]). It is a way to support the structuring of novice's reasoning in a task realization, thanks to the TASK_STATUS simulation parameter. The simulation parameters propose (even if MANAGER is a person) a unique framework for representing the expert's realization as well as the novice's realization of the task. It is thus easier to identify the learning problems of the novice (in the structure or missing knowledge) and therefore help him to improve hos experience.

References

[1] Alarcon R.A., Guerrero L.A. Ochoa S.F., Pino, J.A., "Context in Collaborative Mobile Scenarios", In: P. Zaraté, J.-L. Soubie, M.R.S. , Borges (Eds.), *Workshop on Context and Groupware*. Aachen : CEUR-WS Proceedings, Vol. 133. http://ceur-ws.org/Vol-133/ , 2005.

[2] Anderson J.R., *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press. ISBN 0-19-532425-0, 2007.

[3] Attieh E., Capron F., Brézillon P., "Context-based modeling of an anatomo-cyto-pathology Department Workflow for Quality Control", In: Modeling and Using Context (CONTEXT-13), pp. 235-247. Springer Berlin Heidelberg, 2013.

[4] Banks J., Carson J., Nelson B., Nicol D., *Discrete-Event System Simulation*. Prentice Hall. p. 3. ISBN 0-13-088702-1, 2001.

[5] Bedwell W.L., Wildman J.L., Granados D., Salazar M., Kramer W.S., Salas E., "Collaboration at work: An Integrative Multilevel Conceptualization", *Human Resource Management Review. Elsevier Science, 22(2)*, pp. 128–145, 2012.

[6] Benitez-Guerrero E., Mezura-Godoy C., Montané-Jimenez L.G., "Context-Aware Mobile Collaborative Systems: Conceptual Modeling and Case Study", *Sensors*, *12*, pp. 13491-13507, 2012.

[7] Beyer H., Holtzblatt K., *Contextual Design: A Customer-Centered Approach to Systems Designs*, Morgan Kaufmann Series in Interactive Technologies, 1997.

[8] Bikson T.K., Eveland J.D, *The interplay of work group structures and computer support*. Rand Note N-34-29-MF. Rand Corporation, 1991.

[9] Black K.W., "The small group—tightrope between sociology and personality", *Journal of Behavioral Science, 15*, pp. 283-294, 1979.

[10] 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, pp. 56–68, 2005.

[11] Brézillon P., "Context Modeling: Task model and model of practices", In: B. Kokinov, D.C. Richardson, T.R.

Roth-Berghofer, & L. Vieu (Eds.) *Modeling and Using Context (CONTEXT-07)*. Heidelberg: Springer Verlag, LNAI 4635, pp. 122–135, 2007.

[12] Brézillon P., "Context-based development of experience bases", In: P. Brézillon, P. Blackburn, & R. Dapoigny (Eds.), *Modeling and Using Context (CONTEXT 2013)*, Heidelberg : Springer, LNAI 8175, pp. 87–100, 2013.

[13] Brézillon P., "Contextual modeling of group activity", In: P. Brézillon et al. (Eds.): *CONTEXT 2017*, LNAI 10257, Springer International Publishing, pp. 113-126, 2017.

[14] Brézillon P., Abu-Hakima S., "Using Knowledge in its context: Report on the IJCAI-93 Workshop", *AI Magazine*, Spring, **16**(1), pp.87-91, 1995.

[15] Brézillon P., Attieh E., Capron F., "Modeling glocal search in a decision-making process". In: G. PhillipsWren, S. Carlsson, A. Respício, P. Brézillon (Eds.): *DSS 2.0 – Supporting decision making process with new technologies.* Frontiers in Artificial Intelligence and Applications, 261, IOS Press, pp. 80-91, 2014.

[16] 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, pp. 11-26, 2006.

[17] Brézillon P., Pomerol J.-Ch., "Using contextual information in decision making", In: Chapman and Hall. Widmeyer G., Berkeley D., Brezillon P. and Rajkovic V. Eds.: *Context-Sensitive Decision Support Systems*, pp. 158-173, 1998.

[18] Brézillon P., Pomerol J.-Ch., "Contextual knowledge sharing and cooperation in intelligent assistant systems", *Le Travail Humain, 62(3)*, pp. 223-246, 1999.

[19] Brézillon P., Pomerol J.-Ch., "Framing decision making at two levels",. In : A. Respicio, F. Adam, G. Phillips-Wren, C. Teixeira & J. Telhada (Eds.), *Bridging the Socio-Technical Gap in Decision Support Systems. Challenges for the next Decade*. Amsterdam: IOS Press, pp. 358-368, 2010.

[20] Fan X., Brézillon P., Zhang R., Li L., Pomerol J.-Ch., "A method for making contexts explicit in collaborative workflows". 16th IFIP WG 8.3 International Conference on Decision Support Systems (DSS-2012). Annavissos, Greece, June 28-30, 2012.

[21]Carlsen S., "Action Port model: A Mixed Paradigm Conceptual Workflow Modeling Language", In: *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems*, 1998.

[22] Clancey W.J., "Simulating activities: relating motives, deliberation, and attentive coordination", *Cognitive System Research, 3(3)*, pp. 471–499, 2002.

[23] Elmarzouqi N., Garcia E., Lapayre J.C., "CSCW from Coordination to Collaboration", In: W. Shen, J. Yong, Y. Yang, J.-P. Barthes, & J. Kuo (Eds.), *11th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2007)*. Berlin: Springer 5236, pp. 87–98, 2007.

[24] Eppstein D., "Parallel recognition of series-parallel graphs", *Information and Computation*, 98(1), pp.41-55, 1992.

[25] Finta L., Liu Z., Mills I., Bampis, E., "Scheduling UET- UCT series-parallel graphs on two processors", *Theoretical Computer Science*, 162(2), pp.323-340, 1996.

[26] 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, pp. 243-256, 2015.

[27] Garcia K., Brézillon, P., "From a contextual Graph to a tree representation", In P. Brézillon et al. (Eds.): *CONTEXT 2017*, LNAI 10257, Springer International Publishing, pp. 27-40, 2017.

[28] Garcia K., Brézillon, P., "Model visualization: Combining contextual graph and tree representation", *Expert Systems With Applications*, 99(1), pp. 103-114, 2018.

[29] Gladstein D.L., "Groups in context: A model of Task group effectiveness", *Administrative Science Quarterly*, 29(4), pp. 499-517, 1984.

[30] Gulati R., Wohlgezogen F., Zhelyazkov P., "The Two Facets of Collaboration: Cooperation and Coordination in Strategic Alliances", *Academy of Management Analysis 6*, pp. 531–583, 2012.

[31] Kabil A., Brézillon P., Kubicki S., "Contextual Interfaces for Operator-Simulator Interaction", In: H. Chistiansen, I. Stojanovic, & G. Papadopoulos (Eds.), *Modeling and Using Context (CONTEXT-15)*, Heidelberg: Springer, LNAI 9405, pp. 483-488, 2015.

[32] Leplat J., Hoc J.-M., « Tâche et activité dans l'analyse psychologique des situations », *Cahiers de Psychologie Cognitive, 3*, pp. 49-63, 1983.

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

[34] Nembhard D., Yip K., Shtub A., "Comparing Competitive and Cooperative Strategies for Learning Project Management", *International Journal of Engineering Education*, pp. 181–192, 2009.

[35] Paternó F., *ConcurTaskTrees: an Engineered Notation for Task Models*, The Hand-book of Task Analysis for Human-Computer Interaction, Lawrence Erlbaum Associates, pp. 483-503, 2004.

[36] Pomerol J.-Ch., « Scenario development and practical decision making under uncertainty », *Decision Support Systems* 31 2001, pp. 197–204, 2001.

[37] Pomerol, J.-Ch., Brézillon P., Pasquier L., "Operational knowledge representation for practical decision making", *Journal of Management Information Systems (JMIS), 18(4)*, pp. 101-116, 2002.

[38] Richard J.F., *Logique du fonctionnement et logique de l'utilisation*, Rapport de Recherche INRIA No 202, 1983.

[39] Rosa M.G.P., Borges M.R.S., Santoro F.M., « A Conceptual Framework for Analyzing the Use of Context in Groupware", In W. Shen, J. Luo, Z. Lin, J.-P. Barthes, & Q. Hao (Eds.), *CSCWD 2006*, Heidelberg: Springer Verlag, LNCS 2806, pp. 300-313, 2006.

[40] Salas E., Wildman J.L., Piccolo R.F., "Using simulation-based training to enhance management Education", *Academy of Management Learning & Education*, 8, pp. 559–573, 2009.

[41] Sarrazin P., Biddle S., Famose J.-P., Cury F., Fox K., Durand M., « Goal orientations and conceptions of the nature of sport ability in children: A social cognitive approach », *British Journal of Social Psychology, 35*, pp. 399-414, 1996.

[42] Sokolowski, J.A., Banks, C.M., *Principles of Modeling and Simulation*. Hoboken, NJ: Wiley. p. 6. ISBN 978-0-470-28943-3, 2009.

[43] Trickett S.B., Trafton J.G., " "What if...": The Use of Conceptual Simulations in Scientific Reasoning", Cognitive Science 31, pp. 843–875, 2007.

[44] Van Welie M., Van Der Veer G., *Groupware Task Analysis*. Handbook of Cognitive Task Design, Human Factors and Ergonomics, pp.447-476, 2003.

[45] Feltovich P.J., Johnson P.E., Moller J.H., Swanson D.B., LCS: The role and development of medical knowledge in diagnostic expertise, In: W.J. Clancey & E.H. Shortliffe (Eds.), *Readings in Medical Artificial Intelligence*, Reading, MA: Addison-Wesley, pp. 275-319, 1984.

