A Contextual Model of Turns for Group Work

Kimberly García(✉) and Patrick Brézillon

University Pierre and Marie Curie (UPMC), Paris, France
{Kimberly.Garcia,Patrick.Brezillon}@lip6.fr

Abstract. The Contextual-Graphs formalism has been conceived to represent task realizations in the way they are actually performed. The objective is to provide decision makers with a clear panorama of the different ways a task can be realized (i.e. practices), and the implications of choosing one way or another. The Contextual-Graphs formalism has been successfully used in many fields, such as medicine, biology, and transports, for representing task realizations involving a single actor. In this paper we explore the representation of a group task by analyzing the paper-submission example, from which a turns mechanism is proposed as a way to adapt the Contextual-Graphs formalism to support this type of tasks. Moreover, the types of interaction among actors involved in group task realizations are studied in detail based on a set of definitions introduced in this paper. We claim that the real understanding of group task realizations will not just help decision makers, but will also provide groupware designer with real requirements for building successful applications.

Keywords: Contextual model for CSCW · Turns mechanism · Cooperation · Collaboration · Coordination

1 Introduction

Cooperation and collaboration are two ambiguous notions that have different meanings across domains, and sometimes from one author to another. The difference between these two concepts seems related to the sharing of the goal in the interaction. Collaboration means to "work together", i.e. a joint development of a negotiated and consensual solution, thus it is a task realization in different ways. Cooperation means to "operate together", it is a negotiated division of the task realization among participants (a common goal but autonomous actions) and a pooling by the assembling of each subtask realization in a linear way, each participant has to handle a definite part of the shared task realization. Coordination is the technical organization of the different elements of a task realization to enable participants to work together effectively according to a plan. If coordination has a relatively well-accepted definition across disciplines and types of approaches, such as, cognitive (AI, psychology, etc.) or technology (CSCW, interface design, etc.) [2,7,9], it is not the case of collaboration and

H. Christiansen et al. (Eds.): CONTEXT 2015, LNAI 9405, pp. 243–256, 2015.
DOI: 10.1007/978-3-319-25591-0_18

cooperation. In this paper, we retain that collaboration refers to work together while cooperation is to operate together.

We are interested in bringing the concept of context to group tasks as the relevant knowledge, regarding the task, that each actor possess and shares with the rest of the group, as this would help understanding the way the job gets actually done, which will bring benefits to decision makers and to groupware designers. Commonly, in the technology research field, context is restricted to a set of physical measurements that denotes the state of the environment (e.g., location, and temperature). Our study is based on the Contextual-Graphs formalism [5], which offers a uniform representation of elements of knowledge, reasoning, and context. This formalism has been used by Fan et al. [8] as a contextual complement to scientific workflows in virtual screening, demonstrating that it is more important to model a task realization than its theoretical model. However, the Contextual-Graphs formalism was conceived for task realizations involving a single actor. Therefore, we want to extend it to support group tasks, which represents a different challenge, as each actor involved realizes a set of subtasks in an ordered way (e.g., an actor is able to perform a subtask after another actor has completed theirs), such an actor also needs to share with the rest of the group those elements that are considered important for the group task realizations (i.e. shared context). The interaction between actors evolves, since it can start with superficial communication, but within the task realization development, it could become a strong engagement of the actors. In order to represent such an interaction, we have explored different ways of modeling the example of the paper submission in a contextual model based on the Contextual-Graphs formalism. From this analysis, it was found out that it is necessary to create a mechanism that establishes turns between actors. During each turn, an actor realizes a set of subtasks, and updates the shared context so the next actor understands the state of the task and is able to contribute to the group task.

This paper is organised as follows. In Sect. 2, the Contextual-Graphs formalism as a foundation of the paper is described. Then, in Sect. 3, a series of definitions regarding the levels of interaction among people working together are presented. In Sect. 4, an analysis of the common known paper submission process [1] is introduced in order to find a suitable way to represent group tasks through the Contextual-Graphs formalism. After exploring some options, we propose a mechanism for turns to manage the flow of this type of group tasks in Sect. 5. Then, in Sect. 6, the practice tree implementation as a way to facilitate the visualization of the exchange of turns is introduced. Finally, the conclusions are presented in Sect. 7.

2 On the Contextual-Graphs Formalism

Pomerol and Brézillon [10] define context as the sum of: (1) the contextual knowledge, which is all the knowledge relevant for a person in a given decision problem, (2) the external knowledge, corresponding to the rest of the knowledge that is not important for the current situation, and (3) the proceduralized context, which is a part of the contextual knowledge that becomes important at a specific step of the decision problem. Based on this context definition, Brézillon [4] introduces the Contextual-Graphs (CxG) formalism for obtaining a uniform representation of elements of knowledge, reasoning and context. Thus, a contextual graph represents the realization of a task, each path is a practice developed by an actor in a particular context for realizing the task. A contextual graph represents the accumulated experience of one or several actors realizing the same task.

The elements of a contextual graph are: actions, contextual elements, activities and parallel action groupings. An action is the building block of contextual graphs. A contextual element is a pair of nodes, a contextual and a recombination one; the former has one input and N outputs (branches) corresponding to the N instantiations of the contextual element; the latter is [N, 1] and represents the moment at which the instantiation of the contextual element does not matter anymore. An activity is a graph by itself that is identified by actors because it appears in the same way in different problem solving processes. An activity is defined in terms of the actor, situation, task and a set of actions. Finally, a temporal branching for action grouping expresses the fact that several set of actions can be realized in parallel or in a sequential way, no matter the order.

A contextual element corresponds to an element of the nature that must be analyzed. The value taken by a contextual element when the focus is on it, its instantiation is considered as long as the situation is under analysis. The proceduralized context evolves dynamically during a practice development by addition (at the contextual node) or removal (at the recombination node) of a contextual element during the progress of the focus. Moreover, for group tasks, the shared context is formed by persistent (known by all the actors) contextual elements introduced by an actor, and eventually accepted by others after negotiation. Thus, contextual elements can be used as a way to manage turns among actors involved in a group task, as their instantiation would denote whose turn is next. The working context corresponds to all the contextual elements of a Contextual Graph and their instantiations [3].

3 Levels of Interaction

In the Computer Supported Cooperative Work (CSCW) research field, the group awareness need has been identified since early works as a requirement for applications supporting non-collocated teams [6], since variables such as presence, availability and activity of each member of the team need to be known to coordinate activities and achieve collaboration. Thus, many applications incorporated the availability and presence modes (e.g., busy, online, available, away), to let others users know if the interaction they require could be possible at a specific moment. Then, with the evolution of the field, the creation of applications for supporting not only non-collocated collaboration, but also collocated one, and the increasing attention to the Ubiquitous Computing field [12], arose a new way of thinking about context, since current authors have embraced and developed the view of context as any variable that characterize a situation [11]. Nowadays, the goal is to customize the response of an application based on the environment that surrounds the user and the user themselves. Thus, the most popular variables considered in a group application are presence, availability, activity, location, time and interests. This narrowed vision of context in CSCW is due to researchers concern on building applications limited to physical and environmental features. However, group tasks can be studied at another level, by focusing on human interaction and not just in the applications technical aspects.

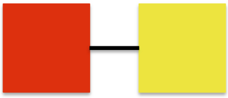
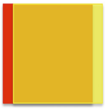
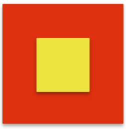
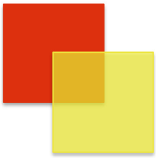
We propose a deeper analysis of the interaction among users involved in a group task. Thus, in Table 1, we illustrate through a set of definitions that there is a large spectrum of collaboration/cooperation, which always involves a degree of coordination. In Table 1 we present six levels in this spectrum, namely: the task level, the task realization level, the actor level, the activity level, the team level, and the planning level. For each level we identify the actors' commitment degree, then we propose a visual guide for each level of interaction, in which rectangles depict tasks (i.e., a task can consist of several actions) performed by a single participant at a specific moment. Finally, in the last column of the table, we use well-known collaborative applications as examples to illustrate that although they are all known as collaborative applications, they are targeted to provide support to one or several levels of interaction.

As mentioned before, most works in the CSCW field are focused on building applications, which is probably the reason why an agreement about the collaboration and cooperation concepts has not been reached yet. However, our objective is to provide support at a lower level, i.e. by understanding the way people actually realize tasks together. In Table 1, we see that it is important to differentiate the level of interaction among actors realizing a task. Since, the over simplification of using the term collaboration and cooperation to refer to any task group leads to lose the essence of the interactions, in which the shared context, the type of turns, and the degree of collaboration/cooperation varies significantly from one situation to another. Our conceptual framework allows to capture some contextual features related to the management of turns that otherwise would not be visible.

4 The Paper Submission Example as a Group Task

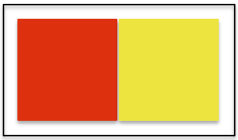
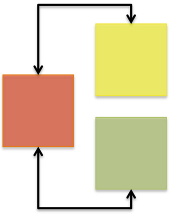
The submission of a scientific paper to a journal is a well-known process in the research community. In short, an author submits a paper to the editor of a journal. The editor may either accept the paper for reviewing, or reject it due to mismatch between the scope of the journal and the topics covered in the paper. An accepted paper is sent for evaluation to (at least) two reviewers. The reviewers read the submitted paper and provide their feedback before a deadline assigned by the editor. The reviewing process is an individual task that the reviewer performs based on his personal knowledge, expertise and point of view. Once the reviews are received, the editor makes a decision by comparing the reviewers' evaluations. This process can be long, so the editor must evaluate their options and time constraints in order to decide the tactics to choose. If

Table 1. Interaction between two actors' tasks

Level of interaction	Actors engagement	Applications corresponding to the type of interaction
1. Task level. Actors have independent tasks, but one task can start only after the completion of another. Interaction is limited to actors' tasks coordination.		An instructor using a learning platform, such as Blackboard, is able to upload an assignment for their students. The instructor will be able to give feedback and grade the student's work. It is a series of tasks that require coordination at all times.
2. Task realization level. Several actors realize (in parallel or at different times) the same task. Practices are accumulated in the same space. This "experience sharing" leads to direct or indirect collaboration.		A wiki is created by a user and fed by others in a synchronous or asynchronous way. Entries are shared in a common space by people in an indirect collaboration, except if someone edits their entry to contribute to the content.
3. Actor level. A helping actor is responsible for an activity in the task realization of another actor. It is a weak cooperation since the global goal belongs to a unique actor. Thus, a quick change of turns is required when the helping actor is involved in the task realization.		Question and answer websites and mailing lists help people to solve technical problems. By posting a question a user receives multiple answers from experts whose interest is just in helping others.
4. Activity level. Actors have independent tasks (and independent objectives) in a joint activity. In some cases, they could be competing but also cooperating to accomplish a common activity. Several actors' turns are activated at the same time, but in order to complete the joint activity, a coordination of turns is required at specific moments.		Crowdsourcing for software development often leads to this type of interaction. In a project, each participant has to realize a precise task as part of the common general task. The client is constantly making new requests. Thus, at specific moments, developers have to work together to perform the joint task, but as their personal goal is to maintain a good reputation in order to be hired again by the client, developers could compete to present the best ideas and work.

(Continued)

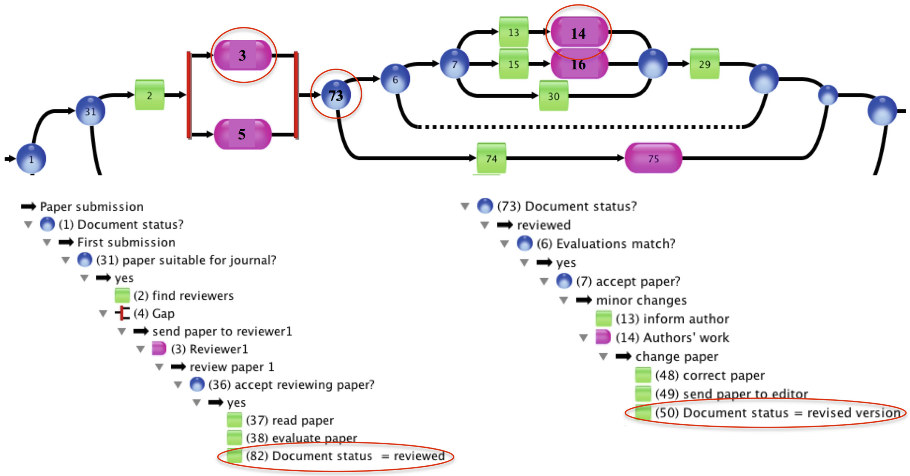
Table 1. *(Continued)*

Level of interaction	Actors engagement	Applications corresponding to the type of interaction
5. Team level. The focus is on the team task realization. There is a strong cooperation because, even when the focus is on a given actor at one moment, the others must coordinate their activities according to the actor "in focus". The shared context concerns all the team members who update it frequently. Interaction is dynamic, and turns are fast.		An application for collaborative edition of documents, such as Google Docs, supports this level of interaction. All members of a team working on a document at the same time coordinate their activities to edit a paragraph. Thus, the actor in focus changes constantly, increasing the need of dynamism of the shared context.
6. Planning level. Interaction between actors is not direct, a manager is responsible for the actor's interactions in their task realizations. The manager needs information about the schedules and actor's plans. It is a lose collaboration that could lead to cooperation once the team is constituted. The manager's turn can be paused while waiting for the termination of each actor's turn.		Google Agenda is an application supporting this level of interaction, by using it, a user is able to create an event, add comments and documents related to it. In case the event is shared and users are accepted to use the agenda, any of those users can provide feedback to interact with the people involved in it.

the paper is not rejected, the editor could: (a) conditionally accept the paper by demanding to the authors an improved version, which is verified by the editor or the reviewers, or (b) accept the paper with the minimal suggested changes. If the final editor's decision is to reject the paper, the author is notified with reviewers' comments. Otherwise, after receiving the new version, the editor sends the accepted submission to the publisher for publication. The paper submission process requires interaction among different actors with different roles: author, editor, reviewers and publisher. We choose to model the actors' interaction in the Contextual-Graphs formalism.

4.1 An Actor's Task as Part of Another Actor's Task

Figure 1 shows a fragment of a contextual graph centered in an actor's vision: the editor, who communicates to other actors whose tasks are embedded in the editor's graph. In Fig. 1 the activities 3 and 5 correspond to the reviewers'



```
➡ Paper submission
▼ (1) Document status?
  ▼ ➡ First submission
    ▼ (31) paper suitable for journal?
      ▼ ➡ yes
          (2) find reviewers
        ▼ (4) Gap
          ▼ ➡ send paper to reviewer1
            ▼ (3) Reviewer1
              ▼ ➡ review paper 1
                ▼ (36) accept reviewing paper?
                  ▼ ➡ yes
                      (37) read paper
                      (38) evaluate paper
                      (82) Document status  = reviewed
```

```
▼ (73) Document status?
  ▼ ➡ reviewed
    ▼ (6) Evaluations match?
      ▼ ➡ yes
        ▼ (7) accept paper?
          ▼ ➡ minor changes
              (13) inform author
            ▼ (14) Authors' work
              ▼ ➡ change paper
                  (48) correct paper
                  (49) send paper to editor
                  (50) Document status = revised version
```

Fig. 1. A partial view of an actor's task inside of another actor's task

tasks, while activities 14 and 16 are author's tasks. In this contextual graph, "Document status" represents the common goal as a shared contextual element. "Document status" may be changed by one actor and then used by another. In this case, when a reviewer has finished or decides not to be involved in the task realization, he will change the "Document status". Figure 1 shows in contextual element 73, that in case the reviewers have evaluated the paper, the instantiation of "Document status" becomes "reviewed" and the editor will be able to continue the decision process, by comparing the evaluations. In case a reviewer has refused to evaluate the paper, the Document status stays "to be reviewed", so the editor needs to find a new evaluator. The introduction of this shared contextual element brings some advantages, as any person could easily understand the task realization flow without having to know the general protocol, they just need to understand the common goal. Furthermore, the working context is always denoting the current status of the process, making easy to obtain information. However, this representation is sequential, since the editor's job is paused until the reviewers change the status of the shared context.

4.2 One Branch for Each Actor's Task Realization

Another way to represent interaction between the editor and reviewers as a working group is to consider all actors' task realization at the same level. In Fig. 2, each branch of the graph corresponds to an actor's activity. The branches correspond (from top to bottom) to the activities of the author, the editor, the reviewer, and the publisher. Thus, an actor is selected by the instantiation of the contextual element.

This representation becomes difficult to follow when a change of actors is needed, since the Contextual-Graphs formalism is not adapted to model group

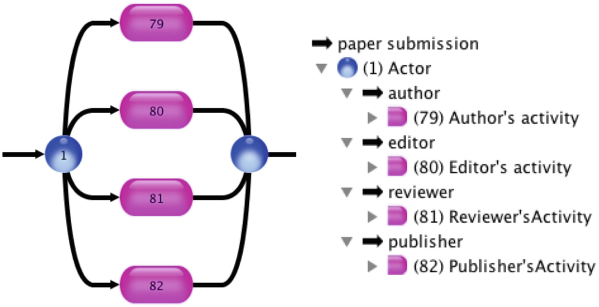


Fig. 2. A partial view of the paper submission problem presenting an actor in each branch

tasks. Thus, elements for guiding the representation should be introduced. Figure 3 shows the extended view of the editor and reviewer activities. Specifically, in action 8 the editor's task is to inform a reviewer about their new assignment. Once the request is sent, the next turn concerns the reviewer. This is realized by reapplying the global contextual graph of Fig. 3 in the new working context with reviewer as the actor in the focus. This representation, understandable in this well-known example, could become rapidly difficult to follow when jumping from one branch to another, without completing any branch. The representation is also inefficient, as it is difficult to know important information about the submission, such as its current status: Is the paper in the reviewers' hands? Has the editor received the evaluations but has not yet made a decision? Is there a conflict between reviewers? The sequence fashion of the representation and the lack of shared contextual elements make it really difficult to notice information rapidly.

4.3 Lesson Learned

Modelling a group task realization supposes the management of several actor's viewpoints in order to be able to follow the development of the task. In the framework of the Contextual-Graphs formalism, a contextual graph is the representation of a task realization. Thus, it is possible to represent the interaction between actors as an activity representing an actor's task realization as a clearly defined part of another task realization, as in Fig. 1. However, it is not efficient to use this approach when two actors interact several times during the group task realization, as it supposes that an actor sending a request waits for the completion of the recipient's task. It is an over-simplification of a group effort. The representation of an actor's activity on each branch of a Contextual Graph is not efficient either for a task involving more than two actors, as it is contradictory with the definition of a contextual element with exclusive instantiations. Moreover, many task realizations could get mixed in a single contextual graph, violating this philosophy, which states that a contextual graph represents a single task realization. Thus, a particular mechanism for turns management in addition

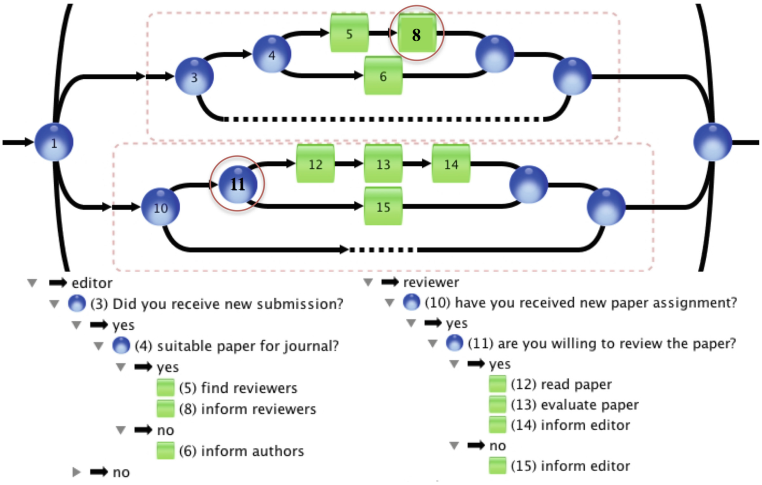


Fig. 3. A partial view of the branches representing the editor and reviewer's task realization

to the use of the shared context between the actors needs to be considered. The shared context will play the role of a virtual working context of the turns mechanism with contextual elements coming from the two working contexts associated with the task realization of the two actors.

5 The Turns Mechanism

To address the limits pointed out in the previous sections, we propose to manage turns with a specific contextual element. Figure 4 shows a "meta graph" for the paper submission example. This graph does not correspond to any actor's particular view, but to the management of the turns between actors. The different actor's views (e.g. the editor and the reviewer) are represented in individual contextual graphs shown in the activities. This representation introduces some reserved words to denote the name of the contextual elements in charge of the sequence of turns among actors. Such words are MANAGER, RECIPIENT and SENDER. The MANAGER is the actor responsible for the task realization at hand (the actor on focus); the RECIPIENT denotes the actor whose turn is next (the next actor on focus); and the SENDER is the actor whose turn is just ending (the actor releasing the focus). Thus, at the completion of the current task, the MANAGER informs the correct RECIPIENT, who in turn, will realize one task or another depending on the SENDER and the shared context. At the beginning of the graph in Fig. 4 the MANAGER becomes the last assigned RECIPIENT, thus the value of the contextual element 1 is known, and the corresponding branch is selected. By the end of a turn, the SENDER takes the value of the current MANAGER, since the next actor in turn needs to know the SENDER. Thus, a cycle is created through these contextual elements by instantiating the

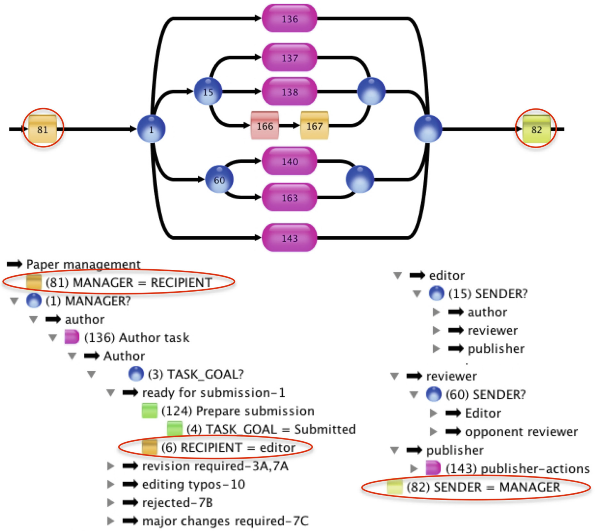


Fig. 4. A partial view of the Turns Mechanism model

initial working context of the current turn with the final working context of the previous turn (i.e. MANAGER = RECIPIENT).

This representation preserves and enhances the use of shared contextual elements. The semantic of these contextual elements are task dependent, and can adopt different values from one actor to another. The idea is to give a common ground to all the actors, for them to be able to know the current situation of the task realization and perform the corresponding task when their turn comes. A shared contextual element is instantiated by one actor and used by another. The TASK_GOAL is a reserved contextual element shared among the actors corresponding to the status of the document (e.g., submitted, reviewed, etc.). Figure 5 shows an example of changing turns. As shown in Fig. 5 at the reception of a new paper, the editor can decide if the paper is suitable for the journal or not. In case the paper is suitable, the corresponding actions are done in the editor's side, then the contextual element TASK_GOAL is instantiated to "to be reviewed" and the RECIPIENT is assigned as "reviewer"; which means that the turn of the editor has ended for now, and the focus should be placed on the reviewer, shown in Fig. 5 as a change from activity 137 to 140. Once the branch corresponding to this specific interaction between the author and the editor has been completed, the SENDER takes the value of "editor" in action 82 on Fig. 5, creating a loop in the reading of the graph, since the MANAGER instantiation will change and the next actor in focus should be found. In this case the focus will be on the reviewer (following the legends of Fig. 5) who detects that the editor has sent a paper to be evaluated, since the TASK_GOAL shared contextual element has been changed in the previous turn to "to be reviewed". The reviewer will change the TASK_GOAL instantiation to either accept or refuse

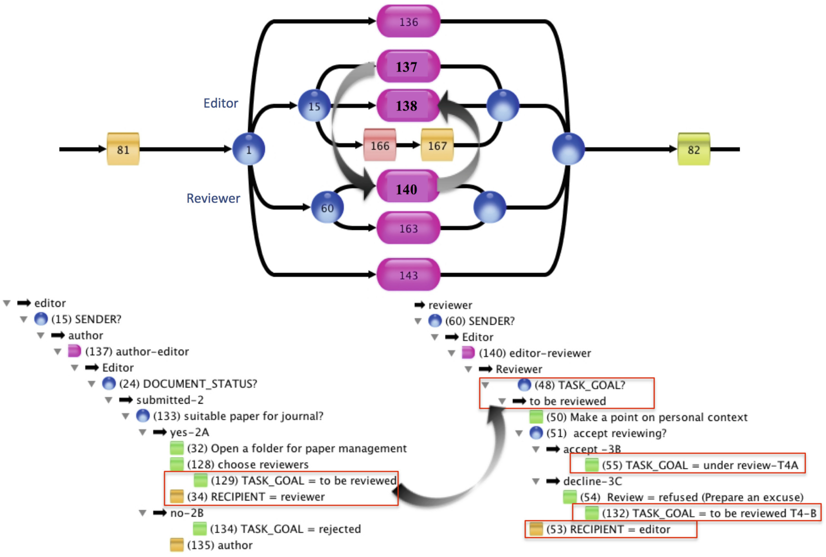


Fig. 5. A partial view of the Turns Mechanism shared context

reviewing the paper, and the RECIPIENT of the focus will be assigned to the editor. Thus, the loop in the reading of the diagram continues, this time from activity 140 to 138. Such a loop is broken when the RECIPIENT is equal to NIL (i.e. it is the end of the overall task).

6 The Practice Tree for Identifying Turns

The Contextual-Graphs software includes a Practice Tree View that helps decision makers to quickly identify the possible practices to develop, considering the information they possess. This view is also important for group tasks, since each tree branch corresponds to a turn. Figure 6 shows the practice tree of the contextual graph in Fig. 5 followed by the practice tree of the actor's activity 137. In the deployed view of activity 137 in Fig. 6, it should be noticed that a decision maker might need to have gathered a lot of information in order to choose the paths containing more contextual elements.

Figure 7 presents the author's practice tree, which is encapsulated in activity 136 in the general graph. Each branch finishes in a change of turns (e.g. author to editor, and author to publisher), meaning that the focus of attention moves from the editor to another actor. Branch 1 of Fig. 7 corresponds to the beginning of the group task, since the author sends a paper to the editor of the journal. The level of interaction in branch 1 is at "task level", regarding the types of interaction presented in Table 1, since the editor's turn is activated until a new paper submission has been received, the author instantiates the TASK_GOAL contextual element to "Submitted" just before their turn is over. The second branch

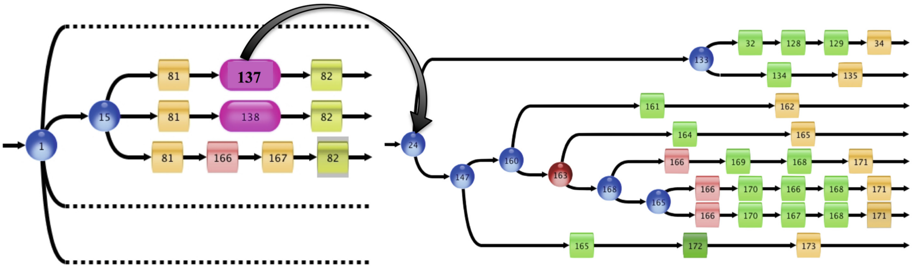


Fig. 6. Author's Practice Tree for decision making

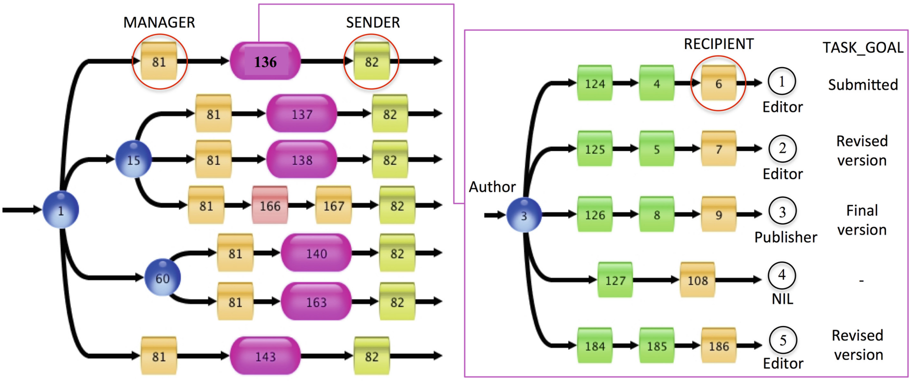


Fig. 7. Author's Practice Tree

corresponds to the author finishing his turn by instantiating the TASK_GOAL contextual element to "Revised version" and passing the focus of attention to the editor. Here the interaction is at the "activity level", since the editor task is different to the author, but they communicate to reach agreements regarding the submitted paper, e.g. the author attaches a document explaining the changes made to the revised version. Once the submission is accepted, on branch 3, the author and the publisher interact at the "team level", since they both work on the document to create the last version of the paper. When the author finishes correcting the typos, he instantiates the TASK_GOAL contextual element to "final version". Branch 4 does not create any further interaction, since it corresponds to the case in which the paper has not been accepted. Finally, branch 5 is really similar to branch 2, because it corresponds to the case in which the author has been asked by the editor to change the paper. The difference with branch 2 is that this time, the editor asks for a complete new version, not just some small changes. Thus, the interaction is at "activity level" and the TASK_GOAL contextual element is instantiated to "revised version". Although not all interaction levels presented in Table 1 can be found in the paper submission example, it is important to do not forget about them, since they are present in different types of group tasks, that have lead to the development of divers applications, such as the ones mentioned in Table 1.

By analyzing the levels of interaction detected in the paper submission group task, we see that, as the actors become more engaged in the group task realizations, the need of just coordination among them, evolves to collaboration and then to cooperation. In branch 1 of Fig. 7, the interaction between the author and editor requires coordination, as the editor tasks starts after receiving a new paper. Not presented in the author's interactions, but following the flow of the submission example, once the editor has sent the paper for reviewing, the interaction between the two reviewers denotes collaboration, as they both realize the same task in their own way. However, the interaction among the editor and the reviewers is at "planning level". Moreover, in Fig. 7, branches 2 and 5, the collaboration evolves to cooperation between the author and the editor, as their common goal is to obtain a new paper version, but each one has a different task to realize. While the author is in charge of providing a new paper version, the editor should answer the author's questions. Finally, once the paper has been accepted for publishing, the author and the publisher engage into a strong cooperation, as they both work in the same document to get it ready for publishing.

Beyond the Paper Submission Example. The paper-submission example is a task that involves several actors constantly interacting to reach a goal. However, group tasks are not limited to this type of team configuration, in which a leader coordinates the group actions and the shared context is reduced. Shorter and more dynamic group tasks (e.g. brainstorming sessions), as long as those including two groups working together at specific times might have several leaders coordinating the task realization of their own group. Thus, further discussion can be held regarding the passing of turns among groups, and the efficiency on the quick management of the shared context modifications.

7 Conclusions

Discussions on collaboration and cooperation must be considered in a large spectrum, going from a superficial approach focused on the built of applications, as found in the literature, to an approach aiming to understand the way people really work together, as proposed in this paper.

There is a need to extend the scope of the research on collaboration and cooperation upstream (experts working jointly to develop a contextual model satisfying everybody) as well as downstream (the contextual model as a medium of communication between experts). This means that it looks difficult to obtain a unique way to create contextual graphs for representing group task realizations. However, this paper explore some approaches. Through the analysis of the paper-submission example, we have notice the importance of the shared context, since it is a way to make compatible an actor's viewpoint with the rest of the group, because each actor is an expert in their own domain. Furthermore, the shared context, along with some reserved words, conform the turns mechanism we propose. Such mechanism helps experts to use the Contextual-Graphs formalism to model their group tasks and to analyze the type of interaction produced in each turn. Moreover, the understanding of a group interaction, will also help experts to make decisions regarding the way the task are realized, and application designers would be able to easily spot the real requirements for supporting a specific group tasks.

Acknowledgment. We thank the CONACyT (Consejo Nacional de Ciencia y Tecnología) for funding Kimberly García's post-doctoral fellowship at LIP6, UPMC.

References

1. Baresi, L., Garzotto F., Paolini, P.: Extending UML for modeling web applications. In: 34th Annual Hawaii International Conference on System Sciences, IEEE (2001)
2. Bedwell, W.L., Wildman, J.L., DiazGranados, D., Salazar, M., Kramer, W.S., Salas, E.: Collaboration at work: an integrative multilevel conceptualization. Hum. Resour. Manag. Rev. **22**(2), 128–145 (2012). Elsevier Science
3. Brezillon, P.: Contextualization of scientific workflows. In: Christiansen, H., Kofod-Petersen, A., Schmidtke, H.R., Coventry, K.R., Beigl, M., Roth-Berghofer, T.R. (eds.) CONTEXT 2011. LNCS, vol. 6967, pp. 40–53. Springer, Heidelberg (2011)
4. Brézillon, P.: Context modeling: task model and practice model. In: Richardson, D.C., Kokinov, B., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 122–135. Springer, Heidelberg (2007)
5. Brézillon, P.: Task-realization models in contextual graphs. In: Dey, A.K., Leake, D.B., Turner, R., Kokinov, B. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 55–68. Springer, Heidelberg (2005)
6. Dourish, P., Bly, S.: Portholes: supporting awareness in distributed work groups. In: Human Factors in Computer Systems CHI 1992, California, USA, May 1992
7. Elmarzouqi, N., Garcia, E., Lapayre, J.C.: CSCW from coordination to collaboration. In: Shen, W., Yong, J., Yang, Y., Barthès, J.-P.A., Luo, J. (eds.) CSCWD 2007. LNCS, vol. 5236, pp. 87–98. Springer, Heidelberg (2007)
8. Fan, X., Zhang, R., Li, L., Brézillon, P.: Contextualizing workflow in cooperative design. In: 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD-11), pp. 17–22. IEEE, Lausanne, Switzerland, June 2011
9. Gulati, R., Wohlgezogen, F., Zhelyazkov, P.: The two facets of collaboration: cooperation and coordination in strategic alliances. Acad. Manag. Ann. **6**, 531–583 (2012)
10. Pomerol, J.-C., Brézillon, P.: Dynamics between contextual knowledge and proceduralized context. In: Bouquet, P., Serafini, L., Brézillon, P., Benercetti, M., Castellani, F. (eds.) CONTEXT 1999. LNCS (LNAI), vol. 1688, pp. 284–295. Springer, Heidelberg (1999)
11. Salber, D., Dey, A.K., Abowd, G.D.: The context toolkit: aiding the development of context-enabled applications. In: The SIGCHI Conference on Human Factors in Computing Systems (CHI 1999), pp. 434–441. ACM, NY, USA, New York (1999)
12. Weiser, M.: The computer for the 21st century. Sci. Am. **265**(3), 94–104 (1991)