

# Context Modeling: Task Model and Practice Model

Patrick Brézillon

LIP6, University Paris 6  
104 avenue du Président Kennedy, 75016, Paris, France  
Patrick.Brezillon@lip6.fr

**Abstract.** Contextual Graphs are a context-based formalism used in various real-world applications. They allow a uniform representation of elements of reasoning and of contexts for describing different human tasks such as troubleshooting and interpretation. A contextual graph represents a task realization. Its paths represent the different ways of reaching this realization, each way corresponding to a practice developed by an actor realizing the task. In this paper, we revisit the classical distinction between prescribed and effective tasks, procedures versus practices, logic of functioning versus logic of use, etc. in the light of this formalism. We discuss the position of the practice model with respect to the task model using an example involving troubleshooting a problem with a DVD player and another example involving the collaborative construction of an answer, and place this within the context of some other applications developed in the formalism of Contextual Graphs.

**Keywords:** Contextual graphs, Reasoning, Task, Activity, Procedures and practices.

## 1 Introduction

Brézillon and Pomerol [4] defined context as "that which constrains something without intervening in it explicitly." We now consider it by extension as the focus of an actor. Several elements justify this definition, the three main elements being that (1) context is relative to the focus, (2) as the focus evolves, its context evolves too, and (3) context is highly domain-dependent. As a consequence, one cannot speak of context in an abstract way.

Next, we can show that the focus at a given stage allows the division of the context into external knowledge and contextual knowledge [1]. The latter constitutes a kind of tank where the contextual elements are to some extent related to the focus in a flat way, whereas the former has nothing to do with the focus. At this conceptual level, the focus acts as a discriminating factor on knowledge in a similar way as for social networks [2]. The focus evolves because a new event occurs (e.g. an unpredicted event) or as a result of a decision made at the previous stage of the focus. The notion of context impacts more on the relationships between knowledge pieces than upon the pieces themselves.

At present, our research on context is organized along two axes, namely models of reasoning represented in Contextual Graphs, and knowledge instantiation of a part of contextual knowledge, which is structured within a proceduralized context. The formalism of contextual graphs has been used in several real-world applications. We present two applications along the first axis, addressing the question of a troubleshooting problem with a device (a DVD player in this paper) and the collaborative construction of the answer to a question. These applications allow the explicit identification of differences between the behavior prescribed by the procedures (corresponding to the instructions) and the actors' observed behaviors (users faced with a problem with a device). This is completely in line with the prescribed and effective tasks identified by Leplat [7] and procedures versus practices [1], and can be found in a number of applications such as road safety (an aid for self-evaluation of drivers), in medicine (an aid for users to query reformulation in a grid environment) and in software engineering (for the assembly of software pieces).

Along the second axis, we discuss in a companion paper [3] the relationships between contextual knowledge and the proceduralized context in order to implement them in a computer system. To address the current status of the focus, the actor selects a subset of contextual knowledge called proceduralized context. In terms of contextual knowledge, the proceduralized context is an ordered series of instantiated contextual elements. The two keywords here are instantiation of contextual elements, which is also the link between the two axes, and comparison of the proceduralized context to a buffer between the focus and contextual knowledge.

In real-world applications, context appears as the “missing link” between domain knowledge and the focus. Brézillon and Brézillon [3] present a study on road safety, the representation of a simple crossroads in terms of a situation dressing. Domain knowledge contains elements like roads, lanes, traffic lights, countryside, city, lights, etc. To define a specific intersection, we must contextualize the domain knowledge (“Place” = “City”, “Traffic lights” = no, etc.). Thus, the contextual element “Place” is instantiated to “City” and this implies that some other domain elements become irrelevant (e.g. “Field of corn” is no longer an instantiation of “At the corner of the intersection”) and others must be instantiated (e.g. “Type of building at the corner”). This kind of dressing of the intersection corresponds to a contextualization of the situation. This contextualization, and thus we go back to the first axis on reasoning, leads to two types of inferencing rules. The first type concerns integrity constraints. For example, “Period of the day” = “Night” implies that the value “Sunny” is not relevant for the contextual element “Weather.” The second type is composed of rules about what a driver must do in a given context. For example, “Period of the day” = “Night” implies that the value “Car lights” must be “Switch on.” The latter rules constitute a kind of theoretical model of the behavior that drivers must enact in the specific context (i.e. dressing) of the situation, that is for the given situation dressing, the current focus. Thus, a student can have the same question (What do you do at the crossroads?) but will always have to reanalyze the situation in the light of the contexts generated randomly.

This paper presents the results of a study along the first axis, namely the difference between the task model and the practice model, the former being established by the company (those responsible for producing the highway code and laws in the previous example, or the engineers in the example of the DVD player hereafter), and the latter representing users' understanding of the task (from reading the instructions in the example of the DVD player). Note that the practice model in the DVD player application results from the collective work of a group of Masters students. There are clear differences with contextual graphs that can be produced by the students individually, but this is beyond the scope of this paper.

Hereafter, the paper is organized in the following way. The next section recalls the main characteristics of the context-based formalism of Contextual Graphs. After, we discuss the two types of model (i.e. task model and practice model) that are identified in a Contextual-Graph representation. The two following sections illustrate these notions within the application of troubleshooting a problem with a DVD player and of the collaborative construction of the answer to a question. After, we present briefly the type of results that we obtained in other applications.

## 2 Contextual Graphs

### 2.1 Context in Decision Making

In a previous piece of work on incident management for subway lines [9, 5], we showed that context-based reasoning has two parts: diagnosis and action. The diagnosis part analyzes the situation at hand and its context in order to extract the essential facts for the actions. The actions are undertaken in a predictable order to realize the desired task. Sometimes, actions are undertaken even if the situation is not completely analyzed (or even analyzed at all). For example, a driver puts a vehicle into gear before any action or situation analysis. Other actions are carried out before the proceduralization of a part of contextual knowledge. Thus, diagnosis and actions constitute a continuous interlocked process, not two distinct and successive phases in context-based reasoning. Moreover, actions introduce changes in the situation or in knowledge about the situation, and imply a revision of the diagnosis, and thus of the decision making process itself. As a consequence, there is a need for a context-based formalism for a uniform representation of diagnosis and actions.

Contextual graphs propose a representation of this combination of diagnosis and actions. (A contextual graph represents a process of problem solving or at least a step in the process.) Diagnosis is represented by contextual elements. When a contextual node is encountered, an element of the situation is analyzed. The value of the contextual element, its instantiation, is taken into account as long as the situation is under analysis. Afterwards, this instantiation does not matter in the line of reasoning that can be merged again with the other lines of reasoning corresponding to other instantiations of the contextual element. Thus, contextual graphs allow a wide category of diagnosis/action representations for a given problem solving process.

## 2.2 The Formalism of Contextual Graphs

Contextual graphs are acyclic due to the time-directed representation and guarantee algorithm termination. Each contextual graph (and any sub-graphs in it) has exactly one root and one end node because the decision making process starts in a state of affairs and ends in another state of affairs (not necessarily with a unique solution on all the paths) and the branches express only different contextually-dependent ways to achieve this goal. This gives the general structure of a spindle to contextual graphs. A path represents a practice developed by an actor, and there are as many paths as practices known by the system.

The elements of a contextual graph are: actions, contextual elements, sub-graphs, activities and parallel action groupings [1]. An action is the building block of contextual graphs. A contextual element is a pair of nodes, a contextual node and a recombination node; a contextual node has one input and N outputs (branches) corresponding to the N instantiations of the contextual element. The recombination node is  $[N, 1]$  and represents the moment at which the instantiation of the contextual element does not matter anymore. Sub-graphs are themselves contextual graphs. They are mainly used for obtaining different displays of the contextual graph by aggregation and expansion, as in Sowa's conceptual graphs [12]).

An activity is a particular sub-graph 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. More precisely, an activity is a sequence of actions executed, in a given situation, in order to achieve a particular task that is to be accomplished by a given actor.

A parallel action grouping expresses the fact (and reduces the complexity of the representation) that several sub-graphs must be crossed before continuing, but the order in which sub-graphs are crossed is not important, and they could even be crossed in parallel. The parallel action grouping could be considered as a kind of "complex context."

## 3 Task Model and Practice Model

### 3.1 What Is Represented in a Contextual Graph?

Contextual Graphs constitute a context-based formalism for representing reasoning. They have been used in a large spectrum of domains such as medicine, ergonomics, psychology, the army, information retrieval, computer security, road safety, the law.

In the decision making domain, actors identify an activity as a recurring structure in problem solving. This recurring sub-structure is a complex action in the spirit of the notion of a scheme given in cognitive ergonomics [13] where schemes are intended for completing sub-goals. Each scheme organizes an activity around an object and can call on other schemes to complete specific sub-goals. A scheme can be specified by a name, a goal and a contextual graph representing a decision-making process that allows its goal to be achieved in a context-sensitive way. Both contextual graphs and

schemes allow the representation of actors' activity and all its variants (procedures and practices), the integration of automatic learning and adaptation in a system, a clear representation of context in actors' reasoning, and the organization of the actors' activity itself.

More generally, enterprises establish procedures based on their experience in order to guide such reasoning. Procedures are collections of secure action sequences developed to address a given focus in any case. These procedures are decontextualized in order to cover a large class of similar focuses (generally differing by their contexts of occurrence), such as the procedure that a driver must follow when arriving at a crossroads, whatever the specificity of the crossroads and the current status of the driver. Such procedures describe the behavior that actors would adopt to address the focus, a kind of theoretical behavior for the actors (i.e. the task model).

Conversely, the practice model corresponds to the effective behaviors displayed by actors facing the focus in a specific context. Differences between the task model and the practice model arise mainly from a difference in the actors' perception due to different backgrounds. For example, everybody uses a refrigerator without difficulty (practice), but few people are aware of the concepts behind the functioning of a refrigerator (i.e. the second principle of Thermodynamics). Thus, if a problem occurs, the system is considered just like a black box, and the refrigerator is immediately brought to the repairer or thrown out and replaced by a new one.

### **3.2 Describing Task Execution...**

The degree of formalization of the task model depends on the nature of the domain. For example, the subway belongs to the domain of Engineering and different ways to buy a ticket in the subway can be modeled exhaustively. Passengers' characteristics (mood, size, etc.) do not matter. Conversely, road safety is a domain where the focus and its context present a large spectrum of possibilities. For example, a crossroads can have the form of X, T, Y or more elaborated topologies, and laws and rules try to capture only the main features of the crossroads, ignoring the number of exceptions and contextual variants (e.g. driving in the fog). Moreover, drivers may develop some practices not anticipated in the Highway Code. Between these two extreme examples, there are applications where the practice model developed by users is based on a logic of use radically (and sometimes deliberately) different from the task model developed by technicians and based on a logic of functioning. This is the case of the DVD player instructions that is presented in the next section.

### **3.3 ...Is not Executing the Task**

Generally there are different methods for the task realization. For example, there are several ways of travelling from Paris to attend a conference in Copenhagen. One method consists of going by car with colleagues because it is the cheapest way. Another method is to take the train because you have time and would like to take time to produce a bibliography and/or to stop in Kaiserslautern first to visit a friend. A third method is to take a plane. Thus, there are three methods for the task "Attend the conference in Copenhagen." The task model, which would describe the actions to

execute in this task, will retain a relatively high degree of generality (e.g. register, book an hotel, buy your ticket for the journey) and is not concerned with the choice of methods available to realize the task.

The choice of method depends on different contextual elements, and mainly on the values (instantiations) that these contextual elements have when the task must be realized. For example, I will pay for my plane ticket with an order form from my university or my credit card, depending on what I have in my bank account because I know that my university generally reimburses 3-4 months later (i.e. after the conference, not when I will be paying, 3 months before the conference). Generally, this level of detail is too fine for a procedure but could be essential for the actor.

A final difference between the task model and the practice model is when a new situation arises. This supposes the revision of the whole structure within the task model, when in the practice model this necessitates the addition of a few elements such as a new contextual element and a few actions. Thus, the practice model is incrementally enriched, but may move away from the task model.

## 4 Example: Troubleshooting a Problem with a DVD Player

This section presents the example of troubleshooting a problem with a DVD player, in the way that this troubleshooting would appear in the user manual for a DVD player [10]. There is a one-page aid for troubleshooting in a 40-page manual. We used the French version for this experiment, but we checked the French translation first against the English version.

### 4.1 Modeling the Task

We first analyzed the troubleshooting page in terms of contextual graphs by associating an alternative with a contextual element. For example, instructions are written such as “If the TV is not switched on, then first switch on the TV”. This instruction can be translated directly into a contextual element “TV power” with two possible instantiations, “On” and “Off”. The entire contextual graph for the troubleshooting page is presented in a simplified view in Figure 1 and the detailed part of “Video problem” is shown in Figure 2.

This knowledge structure is called the engineering viewpoint in a top-down way because the task model (i.e. how to initially troubleshoot a problem with a DVD player) relies heavily on domain knowledge and its structure in terms of mechanics, power, audio, video, etc. This is a “logic of functioning” that is represented as exclusive alternatives in the contextual graph because each step in the engineer’s reasoning is based on a hypothetical-deductive approach. The user model in this troubleshooting presentation impacts on the level of detail to be provided. For a novice with a supposedly very low level of understanding, one sees “If you do not see a red light at the front of your device, then check if the device is connected to the power source,” and to an “expert” with specialized knowledge (e.g. “Change AV”) without explaining what AV means.

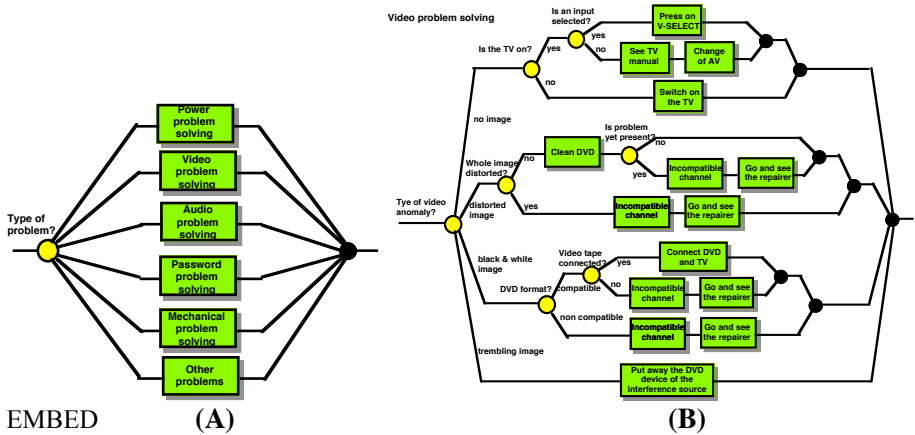
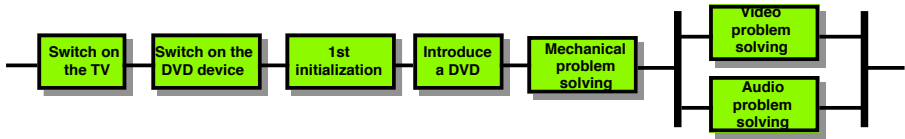


Fig. 1. Contextual graph for the troubleshooting task for a DVD player (A) and details of the video problem solving process (B)

### 4.2 Development of a Practice Model

“See a movie on DVD” is a common task for a user. This (normal) task is accomplished regularly by users, and each time with success. Users move from their normal task to the troubleshooting task when a problem occurs at one stage of this routine task. Thus, troubleshooting concerns a given stage of the normal task, knowing that previous stages of the normal task are accomplished correctly.

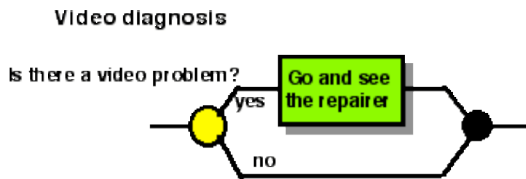
The instructions for troubleshooting the DVD player were used by a group of seventeen Masters level students (studying context modeling at Paris 6 University). The goal of the experiment was to express in the contextual-graphs formalism their collective understanding of the instructions, not to compare their individual understanding, which was the goal of another experiment, “Buy a ticket for the subway in Paris”, presented elsewhere. All the students were aware of how to use a DVD player, although with different perceptions and experience. The result, which is represented in Figure 2, is heavily experience-based. For example, it is natural for them to switch on the TV first before playing a DVD because “I know that the contents of the DVD (i.e. the menu and the movie) are shown on the TV screen.” Second I know that the DVD reader is on because the TV and DVD reader are on the same plug. As a result, firstly they contextualize (i.e. use contextual cues) within their environment, and, secondly, they begin a troubleshooting task at the stage of their “See a movie on DVD” task where a discrepancy with what they expect appears. They follow a logic of use, but it is not a bottom-up approach which would contrast with the top-down approach in the task model. For these reasons, the practice model is sequential, and potential problems appear sequentially (mechanical first, etc.) and are ramified when different experiences and/or contexts intervene at each stage.



**Fig. 2.** Contextual graph of the DVD player troubleshooting task realized by users

Figure 2 presents the practice model developed by the students. Whilst the task model has a parallel architecture (see Figure 1A), the practice model has a sequential architecture (see Figure 2). There is also a strong difference at a finer level of description. Users have bought their DVD player, and this operation is not made frequently. Moreover, engineering parameters that are too subtle for the user (e.g. “Is the image completely or only partially distorted?”) may lead to the user carrying out the wrong action, which could lead to the destruction of the device. Users then prefer to sum up the technical instructions in one action “Send to the repairer”, which, indeed, is also the engineer’s conclusion in most cases.

Users know that any action that is not correctly understood or carried out may lead to serious damage to the device, which will then need a more extensive intervention. Thus, users will decide to go and see the repairer. How serious a problem is for the user is a matter of personal interpretation. However, the risk of permanent damage to the device generally stops users from intervening personally, especially if the advice in the manual is written in incomprehensible language. For all these reasons, the solving of the video problem in practice, as represented by the contextual graph in Figure 3, is quite different to the engineer’s approach (Figure 1B).



**Fig. 3.** Detail for the “visual signal? problem” stage of Figure 2 in the practice model for comparison with the same subtask represented in Figure (1B)

Unlike the engineering engineer’s concern, users pay attention to whether the device is under warranty or not, whereas this is not considered in the user manual. Users are also perfectly aware of what to do the first time they install the DVD player. This knowledge relies on their experience and similarity between the new device and devices already installed. A DVD player is like a VCR, it needs a power supply, an image appears on the TV screen, which thus must be switched on and connected to the DVD player, the remote control needs a battery (generally not provided with the device...) and user-device interaction will take place through the TV screen. There is also new information to learn, generally when a stage in the task is new, like the first time the user watches a movie on several DVDs (e.g. The Lord of



the Rings) or on a double sided DVD. All other problems will cause users to go and take the device to the repairer.

### 4.3 Lessons Learned

The paradigm “Divide to conquer” is not applied in the same way in the engineer's logic of functioning and in users' logic of use. In the former, the problem is divided up according to its nature, i.e. domain knowledge (the causes of a mechanical problem are different to the causes of an audio problem). In the latter, the problem is divided chronologically along the normal temporal sequence of actions to be carried out (one first switches on the TV and then the DVD player and implicitly one checks whether there is a power supply problem).

The shared language between engineers and users is very restricted, especially in technical domains. For example, AV1 (audio-video channel 1) and V-SELECT do not belong to the users' language. This is an argument for users to take the DVD player directly to the repairer. Indeed, technical terms are introduced in the rest of the manual and supposed to be shared by the engineer and users, when users read separately (1) the manual for the installation of the device and to learn about its functioning, and (2) the part of the manual concerning troubleshooting on a different day, when it is needed. Moreover, users do not need an extensive knowledge of mechanics, video, etc. to see a movie on a DVD. This is a striking gap between the two viewpoints.

## 5 Example: Collaborative Answer Construction

### 5.1 The Experiment

In this application, we addressed the question “How can collaboration improve comprehension of documents?” The goal was the design of a model for the task of collaborative construction of an answer to a question, such as “How does an oyster produce a pearl?” “What is a hereditary disease?”, “What is sparkling water?”, etc. Two results of this study are: first, the importance of constructing a shared context (expressed as the sub-task of collecting contextual elements) to constructing the answer, and, second, the relative position of cooperation and collaboration to each other. We focus here on the representation of the construction process in a contextual graph, whereas the whole study is presented in [6]. Here, the task model is constructed from practices that are developed and observed by actors.

Eleven groups of two people participated in the experiments, and had sixteen questions to answer. We thus had 176 MP3 files of 90 seconds each to analyze. The two main findings (at least for our purpose here) concern:

- The *Dialog model*, which contains four “building blocks”: (a) Repeat or reformulate the question, (b) Find an example, (c) Gather domain knowledge (collection), and (d) Construct the answer either by looking for characteristics or by assembling explanation elements (integration).
- The *constructed-answer model*, which possesses four paths: (a) Neither partner knows the answer, (b) Neither partner knows the whole answer but each has

elements with which to provide an explanation, (c) Co-construction of the answer, (d) One of the partners knows the exact answer and provides it.

Each path in the constructed-answer model represents a practice, such as (like?) a combination of the building blocks identified in the dialog model. Figure 4 represents the contextual graph for the constructed-answer model where all these elements are meet, and Figure 5 gives the detail for the “Activity-1” activity in Figure 4.

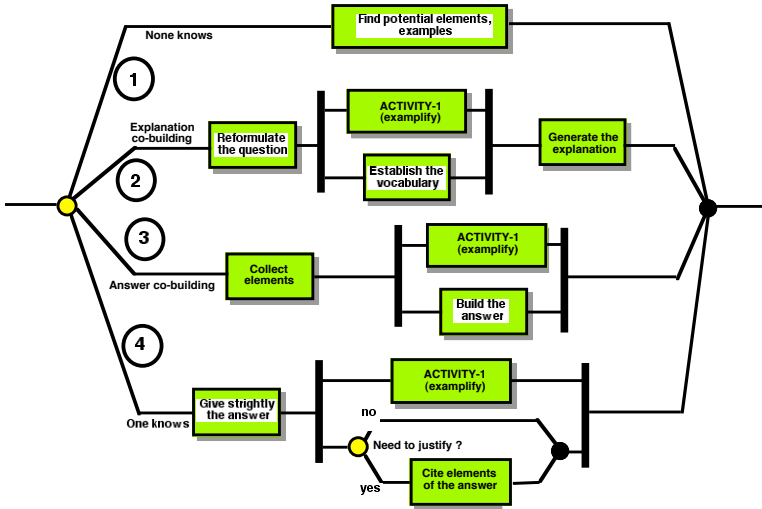


Fig. 4. Contextual Graph of the dialog with four ways to build collaboratively an answer

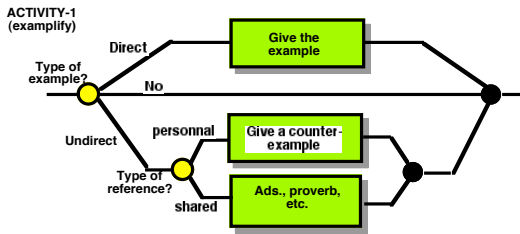


Fig. 5. Details of the “Exemplify” activity in Figure 4

On path 1, neither partner knows the answer. They have no elements of the answer at all. However, they try to utter some rough ideas (for example, a parallel with a known topic or the recall of an advertisement on TV) in order to trigger a constructive reaction from the other.

On path 2, neither partner knows the answer but they think they have elements to generate at least an explanation. Generally, a participant leads the interaction by proposing elements to the other, or asking him/her questions. Explanation generation

is a kind of justification or validation to themselves of their general understanding of the question, without trying to construct an answer.

On path 3, both partners have a partial view of the answer, know some of the elements of the answer and assemble them with the elements provided by the other. (In terms of context, this corresponds to the construction of a proceduralized context.) They hold the same position in the construction of the answer, and there is no need for explanations between them or for an external observer. This is a situation of maximal cooperation. However, without external validation, the quality of the answer is rather variable.

On path 4, one of the partners knows the exact answer, provides it immediately and spontaneously, and spends his/her time afterwards explaining the answer to the other participant. Here cooperation is unidirectional, like the flow of information.

Indeed, there is a relatively continuous spectrum between the path where one participant knows the exact answer (Path 4) and the situation where none of the participants knows it (Path 1).

## 5.2 Lessons Learned

Several observations could be made from these typologies:

- It was difficult to develop the task model further because there are too many combinations (including the absence of one or several blocks) that do not have the same importance for the description of the problem. However, the different paths of the task model represent four practices that constitute (for the moment) four task sub-models for a collaborative construction.
- Participants in the experiments had problems finding the right granularity for their answers. They could also know the answer but not its elements. Thus, a primary difference between the task model and the practice model is that the granularity of the representation matters a lot for actors and may explain variants with the task model. For example, it was observed that at times the answer was given but not using the right words, such as in the example of the use of “gas” instead of “CO<sub>2</sub>.”
- Repetition of the question occurred when participants in the experiments wished to be sure they had understood the question correctly, i.e. to be able to find some relationships between elements of the questions and contextual elements of their mental representation of the domain. This is also a way of having time to retrieve contextual information in their individual contexts.
- Participants collected contextual elements to construct the answer and determine the granularity for the answer they might construct. Once the level of detail was identified, the selection of pieces of contextual knowledge to use in the proceduralized context was automatic. The level of detail in the answer depended more on what they could assemble than on the level of detail that was indicated implicitly in the question.

## 6 Other Applications Developed in Contextual Graphs

In the last five years, contextual graphs have been used in several applications.

The first application was for subway monitoring in Paris [5, 9]. The goal was to represent the different ways in which an operator monitored a subway line in an accident situation. The main findings concerned (1) the usual difference between procedure and practice (i.e. prescribed and effective tasks), and (2) the role of activity for describing actor interaction during task realization. In the latter situation, the actors working together have a representation of the task at different representation levels. There is no real task model in this application because procedures are built from experience, as a "compilation" of past successful practices. Indeed, in some sense the task model should be an abstraction of the practice model.

The second application "Buy a Ticket for the subway in Paris" is close to the previous one. We studied more specifically the identification of practices in the practice model from the knowledge that users had of the subway (ranging from an American boy leaving his country and coming to Paris for the first time, to an experienced user living and working in Paris). Here the task model is known in an exhaustive way, and we hoped that the practice model was totally included in the task model. We nevertheless discovered that users organized differently the "building blocks" of the task model and even added new ones (a young American student was looking for a taxi to go to the subway station).

The third application focused on the different ways a user can make use of the contents of a web page [1]. The focus here was on the practice model, not on the task model. The reason was that users' objectives for information retrieval were beyond the simple task model, which could be summed up as (a) click on the link, (b) look for the keywords, and (c) copy the interesting part. Depending on what the user wants to do with the information, the user can copy just the text, a figure, or the entire page if he/she does not currently have the time to read the web page. In a similar way as for the first application, the task model can only be built a posteriori, from the practice model.

The fourth application was developed as part of a national project with seven partners in order to identify how it was possible to improve web sites presenting French scientific and technical culture on museum web sites [14]. In this application, we entered the domain of engineering, as we did for the second application above, a domain where first the task model can be described in a relatively exhaustive way (the goal was to identify all the paths from the home page to the page where the answer to a question was found), and second the practice model can be identified as a sub-model of the task model (other findings from this study will be discussed elsewhere).

Other applications have been carried out (computer security, for the US army, legal, the Revocation of the Edit of Nantes in History), and others are currently under development: road sign interpretation, drivers' behavior modeling, bug management, software design, medical image retrieval on a grid.

## 7 Conclusion

A contextual graph represents the different methods by which a task can be realized. We have shown that each application leads to special aspects in knowledge representation (the level of detail), man-machine interaction (the need to make different viewpoints compatible), and we have demonstrated the central role played

by the human actor, especially when the actor interacts with a system. Other aspects are for future study.

In the application for the subway, a contextual graph represents the different ways in which the person in charge of a subway line solves a problem. The notion of activity has been introduced for representing parts of problem solving (i.e. a sub-graph) appearing recursively in contextual graphs (e.g. “Clear the train of passengers”). Such an activity belongs within the task of the person in charge of the subway line, such as the action to ask the train driver, but this activity is a complex task for the driver (with information on the train, stop at the next station, etc.). Here interaction between two actors is described in a coherent way at different levels of representation. The contextual graph where the activity is replaced by an action represents the task of the person in charge of the line, whereas the same contextual graph with the activity replaced by the corresponding sub-graphs of actions represents the train driver’s task.

Shefferd [11] discusses the example of the task “Monitoring a given patient” in medicine that is accomplished by different actors (physicians and nurses) collaboratively, but where each actor is responsible for only a part of the whole task. The tree representation of the Hierarchical Task Analysis can be translated (relatively) easily into a contextual graph. This is another approach to the representation of collaborative work in contextual graphs where the global goal can be divided in distinct subgoals for different actors.

At a political level, the question is “Can we say everything in a practice model when the actors know that the Head of the company does not permit certain actions (e.g. for safety reasons, the subway employee must switch off the power before going onto the tracks. For personal reasons, the employee does not do it to avoid writing an incident report.)

At a strategic level, collaboration can be considered as a minimal expression of cooperation: One leads the interaction and the other only feeds in information (or only agrees), reinforcing the statement of the other. Contextual graphs seem to offer an opportunity for studying and modeling collaboration and cooperation among a group of people. We plan to study the different types of interaction between two groups of actors (more precisely, in the Quidditch game in Harry Potter, in the French team of handball, and interaction between car drivers and pedestrians at a crossroads in a town.)

An explanation is given to: (1) justify a known answer, (2) progress in the co-construction of the answer by sharing elements and their interconnection; (3) when participants are not sure of the granularity required in the answer (e.g. partners speak of ‘gas’ instead of ‘CO<sub>2</sub>’ for sparkling water). The explanation (given instead of an answer) is frequently less precise than an answer (generally at a macro-level), and is often for use between the partners. This gives a new insight into what explanations must be, their role and their direct intervention as part of the task at hand.

This series of lessons learned from the use of contextual graphs as a uniform representation of elements of reasoning and contexts must finally be situated within the second part of our work concerning context as an interface to tailor domain knowledge for a given focus. This second part of the work is presented in a companion paper [3] within the framework of the “situation dressing” of a crossroads. Here the link is that we need to consider contextual elements and their instantiations,

and then we have to deal with integrity constraints (at night there is no sun) and with inference rules describing the prescribed behavior of actors in the specific context (provided by instantiations) of the situation. The set of inference rules is certainly related to what we call here the task model.

**Acknowledgments.** The work presented in this paper is ascribed in the ACC project, which is supported by PREDIT and the French Ministry of Transport, mainly in terms of the funding of a Ph.D.Thesis. We also want to thank the members of the ACC project, especially T. Artières, P.Gallinari and Ch. Tijus. Information on the ACC project can be found at [www-poleia.lip6.fr/~jbrezillon](http://www-poleia.lip6.fr/~jbrezillon). Finally, I would like to thank Jacques Leplat for exciting discussions about his views on prescribed and effective tasks.

## References

- [1] Brézillon, P.: Task-realization models in Contextual Graphs. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) *CONTEXT 2005*. LNCS (LNAI), vol. 3554, pp. 55–68. Springer, Heidelberg (2005a)
- [2] Brézillon, P.: Contextualizations in social networks. In: Schultz, S., Roth-Berghofer, T., Brézillon, P. (eds.) *Revue d’Intelligence Artificielle (Special Issue on Applying Context Management)* 19(3), 575–594 (2005b)
- [3] Brézillon, J., Brézillon, P.: Context modeling: Context as a dressing of a focus. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) *CONTEXT 2007*. LNCS (LNAI), vol. 4635, Springer, Heidelberg (2007)
- [4] Brézillon, P., Pomerol, J.-Ch.: Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, vol. 62(3), pp. 223–246. PUF, Paris (1999)
- [5] Brézillon, P., Cavalcanti, M., Naveiro, R., Pomerol, J.-Ch.: SART: An intelligent assistant for subway control. *Pesquisa Operacional. Brazilian Operations Research Society* 20(2), 247–268 (2000)
- [6] Brézillon, P., Draï-Zerbib, V., Baccino, T., Therouanne, T.: Modeling collaborative construction of an answer by contextual graphs. In: *Proceedings of IPMU*, Paris, France, (May 11-13, 2006)
- [7] Leplat, J.: *Regards sur l’activité en situation de travail - Contribution à la psychologie ergonomique*. Presses Universitaires de France, Paris (1997)
- [8] Leplat, J., Hoc, J.M.: *Tâche et activité dans l’analyse psychologique des situations*. *Cahiers de Psychologie Cognitive* 3, 49–63 (1983)
- [9] Pomerol, J.-C., Brézillon, P., Pasquier, L.: Operational knowledge representation for practical decision making. *Journal of Management Information Systems* 18(4), 101–116 (2002)
- [10] Scott: *Owner’s manual*. Lecteur plt de DVD/VCD/MP3/CD, Model N° DSX510 (2005)
- [11] Shefferd, A.: *Hierarchical Task Analysis*. Taylor & Francis, London (2001)
- [12] Sowa, J.F.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove (2000)
- [13] Vergnaud, G.: Concepts et schèmes dans la théorie opératoire de la représentation. *Les Représentation, Psychologie Française* 30(3), 245–252 (1985)
- [14] WebCSTI: *Rapport d’avancement des Travaux Lutin* (2005), [http://www.lutin.utc.fr/pdf/rapport\\_intermediaire\\_WebCSTI\\_Oct\\_2005.pdf](http://www.lutin.utc.fr/pdf/rapport_intermediaire_WebCSTI_Oct_2005.pdf)