

Task-Realization Models in Contextual Graphs

Patrick Brézillon

LIP6, Case 169, University Paris 6, 8 rue du Capitaine Scott,
75015 Paris, France
Patrick.Brezillon@lip6.fr

Abstract. Enterprises develop procedures to address focuses in any case. However, procedures result often in sub-optimal solutions for any specific focus. As a consequence, each actor develops his own practice to address a focus in a given context, focus and its context being particular and specific. The modeling of practices is not an easy task because there are as many practices as contexts of occurrence. This paper proposes a way to deal practically with practices. Based on our definition of context, we present a context-based representation formalism for modeling task accomplishment by users called contextual graphs and its interest for the tasks of incremental acquisition, learning and explanation. Contextual graphs are discussed on a modeling in information retrieval.

1 Introduction

Context plays an important role since a long time in domains where reasoning such as understanding, interpretation, diagnosis, etc. intervenes. This activity relies heavily on a background or experience that is generally not made explicit but gives a contextual dimension to the knowledge and the activity. Thus, context is always relative to something: context of the reasoning, context of an action, context of an object, etc., something that we call focus in this paper.

In this paper, we present contextual graphs that are used in several domains such as medicine, ergonomics, psychology, army, information retrieval, computer security, road safety, etc. The common factor in all these domains is that the reasoning is described by procedures established by the enterprise. These procedures are adapted by actors that take into account the context in which they have to deal with the focus and thus actors create practices as contextualizations of the procedures. Livet [10] shows that for building a robot that put a nut into a bolt, the perfect splined trajectory (i.e. the procedure) is unrealistic because any small error in the positioning of the bolt with respect to the nut will irremediably block the operation. There is clearly a need to introduce capacity of light variations around the theoretical trajectory (i.e. allowing practices), to introduce some play for an adaptation to error of positioning of any kind. A practical reasoning is not a logical and theoretical reasoning for which the action leads to a conclusion. The practical reasoning has more a status of inductive probabilistic reasoning: the conclusion cannot be detached (i.e. take a meaning) from the premises.

Procedures are collections of secure action sequences developed by the enterprise to address a given focus in any case. These procedures are decontextualized for covering a large class of similar focuses (generally differing by their contexts of occurrence), not a narrow focus. As a consequence, actors prefer to plan their action in real time rather than to rely on procedures for two main reasons. Firstly, the procedure is not perfectly adapted to the situation at hand and can lead to improper actions or sub-optimal incident resolution strategies. Secondly, the actor can miss some important facts and notice them too late to adequately solve the incident. Thus, the modeling of actors' reasoning is a difficult task because they use a number of contextual elements. These pieces of knowledge, which are not necessarily expressed, result in more or less proceduralized actions that are compiled in comprehensive knowledge about actions. However, procedures are considered as useful guidelines for actors to be adapted for particular focuses. Thus, each actor develops his own practice to address a focus in a given context, and one observes almost as many practices as actors for a given procedure because each actor tailors the procedure in order to take into account the current context, which is particular and specific.

This is a general way to reach the efficiency that decision makers intended when designing the task [2, 3]. Such know-how is generally built up case by case and is complemented by "makeshift repairs" (or non-written rules, "magic book", etc.) that allow the actors to reach the required efficiency. This is a way of getting the result whatever the path followed.

If it is relatively easy to model procedures, the modeling of practices is not an easy task because they are as many practices as contexts of occurrence of a given focus. Moreover, procedures cannot catch the high interaction between the task at hand and the related tasks that are generated by the task itself.

Hereafter, the paper is organized in the following way. Section 2 presents our view on context with a presentation of a general view and our definition. Section 3 discusses the different aspects of the building of the proceduralized context that plays a central role in our approach. Section 4 presents the context-based representation called contextual graphs that relies on the ideas develop in the previous section, with its main characteristics, an example, the way in which the proceduralized context is managed in contextual graphs, and some other properties with a more general impact.

2 Context Characteristics

2.1 Context and Focus

We cannot speak of context out of its context. Context surrounds a focus (e.g. the task at hand or the interaction) and gives meaning to items related to the focus. The context guides the focus of attention, i.e. the subset of common ground that is pertinent to the current task. Indeed, context acts more on the relationships between items in the focus than on items themselves, modifying their extension and surface.

As a consequence, the context makes the focus explicit and the focus defines the relevant pieces in the context. On the one hand, the focus determines what must be contextual knowledge and external knowledge at a given step. For example, a focus on software development implies contextual knowledge such as the programming

language, the constitution of the designer team, etc., i.e. knowledge that could eventually be used when the focus evolves. Some knowledge from the designers' individual context could also be considered such as a previous experience with a given piece of software. On the other hand, the context constrains what must be done in the current focus. This could correspond to the choice of a specific method at a given step of a task. A software programmer will focus his/her programming activity in defining classes and methods when in an object-oriented project, he/she will define modules and functions if the project uses the functional paradigm. Indeed, some contextual elements are considered explicitly, say for the selection of the method and thus can be considered as a part of the way in which the problem is solved at the considered step.

The focus evolves along the execution of a series of actions. According to this dynamics of the focus, there is a dynamics of its context (some external events may also modify the context of the focus): focus and its context are intertwined.

2.2 A Definition of Context

For a given focus, Brézillon and Pomerol [3] consider context as the sum of three types of knowledge. First, there is the part of the context that is relevant at this step of the decision making, and the part that is not relevant. The latter part is called **external knowledge**. External knowledge appears in different sources, such as the knowledge known by the actor but let implicit with respect to the current focus, the knowledge unknown to the actor (out of his competence), contextual knowledge of other actors in a team, etc. The former part is called **contextual knowledge**, and obviously depends on the actor and on the decision at hand. Here, the focus acts as a discriminating factor between the external and contextual knowledge. However, the frontier between external and contextual knowledge is porous and evolves with the progress of the focus.

Second, a sub-set of the contextual knowledge is proceduralized for addressing the current focus. We call it the **proceduralized context**. The proceduralized context is a part of the contextual knowledge that is invoked, assembled, organized, structured and situated according to the given focus and is common to the various people involved in decision making.

It could arrive that it is not possible to build the right proceduralized context for a given focus because some information is missing. Once identified from the distance between the focus and the built PC, this missing knowledge can be acquired from the external knowledge and introduced in the sub-set of contextual knowledge on which is built the proceduralized context. This is a process of incremental knowledge acquisition that is intertwined with a process of practice learning as discuss in the next section.

2.3 Conditions of the PC Building

When an element of the contextual knowledge moves in the proceduralized context, this means that we consider explicitly its current instantiations. For example, when I go to my Lab., I look outside to the weather (contextual element of my travel). If it is raining (instance of the contextual element), then I take my umbrella (the action realized with the instantiation of the contextual element) because the subway is not close from my home (other contextual element). With the identification of the

contextual element concerned and its specific instantiation, the actor has executed an action (or method) different of the action known by the system (I take my coat if it is not raining in the example).

We consider the process of proceduralized context building (PC building) in two situations, namely a building by a unique actor or by at least two actors. The focus is associated with the contextual knowledge (CK), part of the actor's context that could be mobilized for the current focus; the other part of the context being the external knowledge (EK). A sub-set of CK is explicitly consider to address the focus. Elements of this CK sub-set are assembled, organized, compiled as a proceduralized context (PC) to address the focus.

Sometimes, the building of a proceduralized context fails for a given focus and new (external) knowledge is needed. Associated with the proceduralized-context building, there are simultaneously an incremental acquisition of new contextual elements and the learning of a new practice. Acquisition and learning occur in a specific context that is acquired and learned jointly with the new knowledge. Indeed, first, learned practice and acquired knowledge are more properly two aspects of the learning process, and, second, learning is more an assimilation process rather than an accommodation process because this corresponds generally to a process of refinement of contexts. Second, if the addition of a piece of external knowledge in the PC correspond to a knowledge acquisition, it is nevertheless a learning process because the piece of external knowledge is not simply added to the PC, but assembled and connected to the construction already existing. Thus, it is possible to explain and justify each practice or item in the contextual graph later.

This triple aspect—context growth by integration of external knowledge in the PC building, by integration of a new “chunk of knowledge” in the contextual knowledge, and context change by the movement between the body of contextual knowledge and proceduralized contexts—gives a dynamic dimension to context [2]. This dynamic component is generally not considered in the literature and explains why making context explicit in an application is a difficult task, except if we restrict context at what can be obtained by sensors like in context-aware applications.

In a usual situation, a sub-set of contextual knowledge (CK) pieces are chosen, assembled and structured in a proceduralized context (PC) for addressing the current focus. Once this PC has satisfied the focus, this chunk of knowledge goes back in CK. Thus, CK contains the PC, the initial sub-set of CK elements and the way in which the PC has been built. This allows relevant explanation on why the PC was necessary for the focus, how the PC has been built and what the PC contains.

2.4 Learning of Practices

Learning is a social process that involves building connections: connections among what is being learned and what is important to the actor, connections among what is being learned and those situations in which it is applied, and connections among the actor and other actors of the work group (e.g. see [1]).

2.4.1 Learning by Assimilation of External Knowledge

The Proceduralized Context (PC) built from the contextual knowledge sometimes does not correspond to the focus. As a consequence, there is a need to add some

pieces of External Knowledge (EK) in the proceduralized-context building. EK is the (contextual) knowledge not represented because never needed up to now. An EK piece is acquired when a new practice is introduced in the graph (learned by the system). This addition occurs once the discrepancy between the focus and the PC built is known and the missing knowledge identified. This generally occurs because a contextual element, which was not considered before, had the same instantiation all along the practice development and in the particular situation at hand takes another instantiation. With this specific instantiation of a contextual element, a new item (action, activity, method) has been used. The main point here is that an EK piece is introduced first in the sub-set of contextual knowledge used for the PC building, not directly in CK. Our approach brings a solution to the problem of the infinite dimension of context evoked by McCarthy [11]. As a consequence, each element, and especially a contextual element, in a contextual graph has been used at least once.

This situation corresponds to an incremental process of (1) acquisition of new knowledge pieces, and (2) learning of a new knowledge structure as a proceduralized context. If the addition of an EK piece in the PC correspond to a kind of incremental knowledge acquisition, it is also a learning process because the piece of external knowledge is not simply added to the PC, but assembled and connected to the construction already existing and corresponds to the learning of a new way to build a PC as discussed in Section 3.3 and Figure 2 below.

2.4.2 Experience Learnt After PC Use

Once a PC allows the evolution of the focus, the PC is not lost and goes into the body of contextual knowledge from which are coming its elements. This is not only the product that is stored, but also all the way in which this PC has been built, the reasons behind the choices, the alternatives abandoned, etc. The PC is totally integrated in the body of contextual knowledge and the learning is here an accommodation process. The PC that is stored could be recalled later either as a whole (as a part of a new proceduralized context, a chunk of knowledge a la Schank, [14] or the way in which it has been built will be reused for the new proceduralized context. This is a type of learning by structuration of the contextual knowledge, and the more a person is experimented, the more the person possesses available structured knowledge (i.e. chunks of contextual knowledge).

3 Representation by Contextual Graphs

3.1 Characteristics

A contextual graph is a context-based representation of a task execution. Contextual graphs are oriented without circuits, with exactly one input and one output, and a general structure of spindle. A path (from the input to the output of the graph) represents a practice (or a procedure), a type of execution of the task with the application of selected methods. There are as many paths as practices. Different solutions can be associated with the unique output, like in the following example chosen in information retrieval: abandon, copy, or save a page before to close the window, but all of them lead to the same conclusion: end of the exploration of the

page. A contextual graph is an acyclic graph because user's tasks are generally in ordered sequences. For example, the activity "Make the train empty of travelers" is always considered at the beginning of an incident solving on a subway line, never at the end of the incident solving. A more drastic divergence in the type of output (e.g. the execution of the task is stopped like "Error 104" in information retrieval) must be considered at a upper level in which the contextual graph at hand is a branch of an alternative (a contextual element such as "Are the conditions required for the task execution present? If yes got to the contextual graph otherwise does not consider this contextual graph).

Elements of a contextual graph are: actions, contextual elements, sub-graphs, activities and parallel action groupings.

- An **action** is the building block of contextual graphs. We call it an action but it would be better to consider as an elementary task. An action can appear on several paths (see the example of the action A5 in the example below). This leads us to speak of instances of a given action, as one speaks of the instantiation of a contextual element because an action (e.g. action A5 in Figure 2), which appears on several paths in a contextual graph, is considered each time in a specific context.
- A **contextual element** is a couple 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 already encountered. The recombination node is $[N, 1]$ and shows that even if we know the current instantiation of the contextual element, once the part of the practice on the branch between the contextual and recombination nodes corresponding to a given instantiation of the contextual element has been executed, it does not matter to know this instantiation because we do not need to differentiate a state of affairs any more with respect to this value. Then, the contextual element leaves the proceduralized context and (globally) goes back to the contextual knowledge.
- A **sub-graph** is itself a contextual graph. This is a method to decompose a part of the task in different way according to the context and the different methods existing. In contextual graphs, sub-graphs are mainly used for obtaining different displays of the contextual graph on the graphical interface by some mechanisms of aggregation and expansion like in Sowa's conceptual graphs [15].
- An **activity** is a particular sub-graph (and thus also a contextual graph by itself) that is identified by actors because appearing in several contextual graphs. This recurring sub-structure is generally considered as a complex action. Our definition of activity is close from the definition of scheme given in cognitive ergonomics [9]. Each scheme organizes the activity around an object and can call other schemes to complete specific sub-goals.
- A **parallel action grouping** expresses the fact (and reduce the complexity of the representation) that several groups of actions must be accomplished but that the order in which action groups must be considered is not important, or even could be done in parallel, but all actions must be accomplished before to continue. The parallel action grouping is for context what activities are for actions (i.e. complex actions). This item expresses a problem of representation of items at a too low level of granularity. For example, the activity "Make train empty of travelers" in the SART application [4, 12] accounts for the damaged train and the helping train.

There is no importance to empty first either the damaged train or the helping train or both in parallel. This operation is at a too low level with respect to the general task "Return back rapidly to a normal service" and would have otherwise to be detailed in three paths in parallel (helping train first, damage train first, both in parallel) leading to the same sequence of actions after.

3.2 An Example

Figure 1 gives an example of contextual graph (the definition of the symbols are given in the Table 1) and Figure 2 presents the Activity-1 in the contextual graph represented in Figure 1. This contextual graph represents the different practices that can be used during an information retrieval when one clicks on a link. Square boxes represent actions, circles represent contextual elements (large circles for contextual nodes and black circles for recombination nodes). Rectangular boxes represent activities that can be opened as a sub-graphs. (There is no parallel action grouping in this example.) A path is followed from the left to the right and corresponds to the crossing of a series of elements.

The structure size of such a contextual graph is easily controlled and the consideration of a new contextual element will add few elements and do not increase drastically the size such as in a tree representation where the addition of a contextual element leads often to double the size of the tree for very few changes. A contextual graph represents a specific procedure and all its variants (the practices) at a given level of representation. The number of practices is not infinite and thus the size of the contextual graph is controlled easily because we consider changes only at the same level of representation of the problem solving (otherwise parallel action grouping are introduced in the contextual graph). Moreover, a new practice is generally introduced in a contextual graph as a variant of an existing practice differing from the previous one by a contextual element that was not initially taken into account because its instantiation did matter any more, and by an action or an activity. As such, a contextual graph appears as the corporate memory of the task execution represented by this contextual graph.

Table 1. Exploration of a link target

Contextual element	
C1	What is the link target ?
C2	Is there a HTML version ?
C3	Is the page interesting ?
C4	Are there figures to retrieve ?
C5	Have I time now ?
C6	Is it for a course ?
C7	Can the content of the page be retrieved?
C8	Duration of the download ?
C9	Explore the whole site ?
C10	Is the site already known?
C11	Is the whole page interesting?
C12	Which type of search is it?

Action	Definition
A1	Copy the slide
A2	Paste the slide in my ppt document
A3	Note the idea for a future exploitation
A4	Explore the rest of the presentation
A5	Close the window
A6	Save the page at the html format
A7	Copy and paste the text in a Word document
A8	Open the target in a new window
A9	Download the document
A10	Store the document in the « To read » folder
A11	Open the document
A12	Look for keywords on a colored background
A13	Search where are keywords in the page
A14	Go the home page of the site
A15	Go to the next slide
A16	Look for new stuffs
A17	Select the interesting part of the text on the page
A18	Look for searched item (e.g. reference of a read paper)
A19	Search management (key words or search engine)
A20	Select the text of the whole page

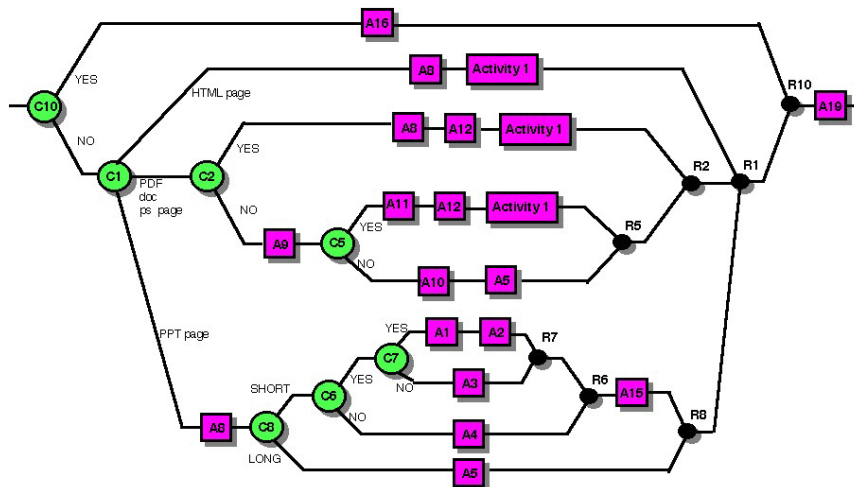


Fig. 1. Exploration of a link target found on a Web page

Contextual elements existing in the contextual graph belong to the contextual knowledge or the proceduralized context depending on the current focus. The context of an action, say action A6 at the top of Figure 2, is described in a static way with a fixed number of pieces of contextual knowledge (C9 and C11), and of an ordered sequence of instantiated contextual elements of the proceduralized context (C8 with the value “Short”, C12 with the value “Exploratory”, C3 with the value “Yes”, and C4

with the value “Yes”). Note that an action such as A5 that appears in several paths (i.e. practices) in Figure 2 is associated with different contexts according to its position in the graph.

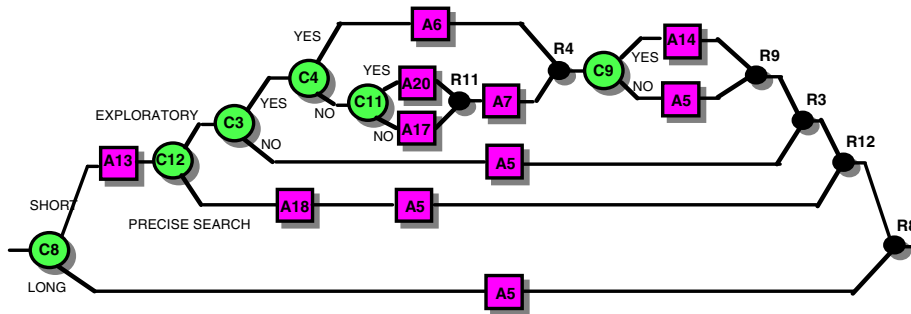


Fig. 2. Information exploitation on a link target (Activity 1 in Figure 1)

The context of a practice (say the upper path including the action A6 in Figure 2), is described in a dynamic way along the progress of the application of the practice (the current focus). For example, the practice represented by the upper path in Figure 2 goes through the steps described in Table 2.

Table 2. Status evolution of context pieces Ci during practice progress (with CK and PC standing respectively for contextual knowledge and proceduralized context with instantiations of the C in the PC)

Step	C8	C12	C3	C4	C9	C11
0	CK	CK	CK	CK	CK	CK
1	PC (short)	CK	CK	CK	CK	CK
2	PC (short)	PC (Exploratory)	CK	CK	CK	CK
3	PC (short)	PC (Exploratory)	PC (yes)	CK	CK	CK
4	PC (short)	PC (Exploratory)	PC (yes)	PC (yes)	CK	CK
5 (= 3)	PC (short)	PC (Exploratory)	PC (yes)	CK	CK	CK
6	PC (short)	PC (Exploratory)	PC (yes)	CK	PC (yes)	CK
7 (= 3)	PC (short)	PC (Exploratory)	PC (yes)	CK	CK	CK
8 (= 2)	PC (short)	PC (Exploratory)	CK	CK	CK	CK
9 (= 1)	PC (short)	CK	CK	CK	CK	CK
10 (= 0)	CK	CK	CK	CK	CK	CK

The dynamics of the context during the application of a practice corresponds here to the movement of contextual elements between the contextual knowledge and the proceduralized context. Note also that the context of the practice at step 4 corresponds to the context of the action A6.

3.3 PC Building in a Contextual Graph

When the system fails to represent the practice used by an actor (given as a sequence of actions such as A13-A6-A14 for the upper path), the system presents the actor the practice the nearest of the actor's practice, exhibits the differing part between the practices, and ask for an explanation. For example, in Figure 2, there was A13 followed, say, by action A7 according to the contextual element C3 (implicitly the type of search was supposed to be exploratory). However, an actor may decide to look for the phone number of a colleague on the Web and gives very precise keywords and has his answer in the first link provided by the browser. The actor then explains that the contextual element C12 "types of search?" must be introduced because the treatment is different from the previous one, i.e. an exploratory search. (The numbering of the contextual elements in Figures 1 and 2 is made in the order in which elements were introduced when we have modeled this task and represented it by this contextual graph).

The proceduralized context is knowledge that is explicitly used at the current focus (e.g. the action A6 in Figure 2 is executed because the contextual element C4 is instantiated with the value "Yes" that is thus explicitly considered in the focus. A proceduralized context is represented as an ordered set of contextual elements that fit into each other like a nest of dolls (Russian dolls). Figure 3 gives a context-based representation of the contextual graph in Figure 2 where the instantiated contextual elements (contexts C8, C12, C3, etc.) are replaced by oval for symbolizing the different contextual elements during their instantiation, and a path crossing different contexts along the numbered dots. A path entering a context means that the corresponding contextual element is instantiated (enters the proceduralized context as at dots 01, 02, 03, 04, 05 and 08 and thus enter the focus too) or decontextualized when leaving a context (as dots 06, 07, 09, 10, 11, 12 when leaving the proceduralized context). As a consequence, the practice leaves first the last contextual element entered. Thus, what is important is not so much the collection of contextual elements but the way in which they are ordered to allow the execution of a given action or a practice.

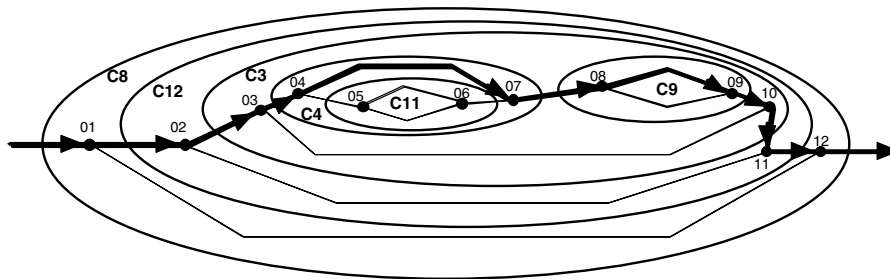


Fig. 3. A context-based representation of the graph in Figure 2 (See legend in the text)

This work, yet to be developed, is reminiscent of the work of Giunchiglia's team (e.g. see [6]). This is a new path to explore to developing contextual graphs as a formalism of representation of the knowledge and the reasoning.

3.4 Implementation Status

Right now, there is a software for visualizing contextual graphs and incrementally acquiring new practices. There are already a number of functions (zoom, handling of parts of a graph, change of language, etc.) for handling contextual graphs and the current development concerns the introduction of explanations. The use of this software in a number of applications shows that if the software itself cannot be separated from its interface (because of the incremental acquisition capability), the software is really independent of the applications already developed. A next step will be the integration of this piece of software in an intelligent support system.

Now, we study the re-writing of a graph in an operational programming language as, say, production rules. For example, there are two ways to write the path uses in the previous example (with the action A6 in Figure 2). The first one is to write the whole path as a unique rule, something like:

```

IF      C8      = (short)
        A13 is executed
        C12     = (exploratory)
        C3      = (yes)
        C4      = (yes)
THEN Execute action A6.

```

The second way is to have as many production rules as contextual elements, rules like:

```

RULE-11
  IF      C11     = (yes)
  THEN Execute action A7
  ELSE Execute action A17.
RULE-C4
  IF      C4      = (yes)
  THEN Execute action A6
  ELSE Check RULE-11.

```

This is only one way to obtain an operational representation of a contextual graph. Indeed, it would be also possible to use Bayesian networks, influence graphs, etc. for representing a contextual graph because we distinguish a model of a domain from its representation in a given formalism.

Contextual elements appear early in contextual graphs because actors want first to have a maximum of information. Then actors try to retrieve as much as possible known intermediary situations in order to reuse known strategies. A contextual graph is a kind of corporate memory and a return from experience. This could be the basis for the building of a more robust procedure (based on experience).

4 Conclusion

We propose contextual graphs for a uniform representation of elements of reasoning and contextual elements at the same level. This is different from the view of Campbell and Goodman in [5] for example (and Hendrix, [7] before for semantic networks) that consider context as a way to partition a graph. Moreover, context in our formalism intervenes more at the levels of the links between actions than actions themselves. Contextual elements being organized in contextual graphs in the spirit of “nest of dolls”, we have not a hierarchy of context because a given contextual element is itself contextualized and can appear encompassed in different other contextual elements. Rather, a contextual element is a factor of knowledge activation.

We show that contextual issues cannot be addressed in a static framework only and that eliciting and sharing contextual knowledge in a dynamic way is a key process in addressing and understanding context problems.

Moreover, one finds in the literature the management of a local context in reference to a global context. The literature on context-aware systems distinguishes two types of context: (1) the "local" context that is close of the focus of attention and highly detailed, and (2) the "distant" context that is general (with less details). This approach is also found in other domains. For example, van Dijk [16] presents a very similar position on political discourses with:

- A local or micro context (called situation), defined by a specific setting and specific participants, and
- A global or macro context, informally defined in terms of higher level societal structures, involving, e.g., groups, group relations (such as power and inequality), organizations, institutions or even whole states and nations.

van Dijk embeds this view by representing context as a mental model. Contextual graphs propose a continuous view on context instead of such a dichotomy micro versus macro, local versus global, internal versus external, etc. We plan to extend this notion of granularity in situation of collaborative work where we distinguish different types of context at different levels. Movement from one context to another one is ensured by a proceduralization of a part of the knowledge from the first context to the second context. It is a way to manage different users' context (far or global context) and the collaborative work context (the local context).

The introduction of the item “Parallel Action Grouping” (PAG) simplifies the representation of contextual graphs. However, if an activity is assimilated to a complex action, a PAG is more than a complex contextual element. An activity sums up a complexity at the same level of representation. A parallel action grouping generally represents (as a simplification) a complex entanglement of contextual elements corresponding to a low level of description of the problem solving modeled in the contextual graph. In the popular example of the coffee preparation given in UML manuals, it is said that we must take the coffee and the filter in one order or the other (or in parallel). However, according to the type of coffee machine (e.g. we must put it apart to fill the reservoir with water), the piece of the coffee machine where must be put the filter can be independent of the coffee machine, mobile on the coffee machine or fixed into the coffee machine. Each situation would be considered independently, but all situations will conclude on a unique action: “Put the coffee in

the filter.” Thus, instead of making complicated a contextual graph for representing its (natural) complexity, which is at a lower level of detail, we use parallel action groupings.

Information can be drawn from a contextual graph, such as the way in which it has been developed, which actors has developed a given part of the contextual graph. It is possible to have an evaluation of the distance between two practices (i.e. two paths in the contextual graphs). Contextual graphs are a formalism of representation allowing the description of decision making in which context influences the line of reasoning (e.g. choice of a method for accomplishing a task). This formalism has been already used in different domains such as medicine, incident management on a subway line, road sign interpretation by a driver, computer security, psychology, cognitive ergonomics, usual actions in a house (preparing hard-boiled eggs, change of an electric bulb, etc.). The extensions that will be given to this work concerns: (1) its introduction in an intelligent assistant system for providing suggestions to the actor, (2) the management of the database (operations on the items, regrouping contextual elements, etc.) in order to produce robust procedure; statistic on the development, use of a given path; and (3) the introduction of a module of explanation generation of different types and at different levels of details.

References

- [1] Barab, S.A., Scheckler, R. and MaKinster, J.: Designing System Dualities: Building Online Community. American Educational Research Association in Seattle, WA (April, 2001).
- [2] Brézillon, P.: Representation of procedures and practices in contextual graphs. *The Knowledge Engineering Review*, 18(2) (2003) 147-174.
- [3] Brézillon, P. and Pomerol, J.-Ch.: Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3), Paris: PUF, (1999) pp 223-246.
- [4] Brézillon, P., Cavalcanti, M., Naveiro, R. and Pomerol, J.-Ch.: SART: An intelligent assistant for subway control. *Pesquisa Operacional*, Brazilian Operations Research Society, 20(2) (2000) 247-268.
- [5] Campbell, B. and Goodman, J.: HAM: A General Purpose Hypertext Abstract Machine. *Commun. ACM*, 31(7) (1988) 856-861.
- [6] Serafini, L., Giunchiglia, F., Mylopoulos, J. and Bernstein, P.: Local relational model: A logical formalization of database coordination. In: P. Blackburn, C. Ghidini, R. Turner, F. Giunchiglia (Eds.), *Modeling and Using Context*, Springer Verlag, LNAI 2680, pp. 286-299 (2003)
- [7] Hendrix, G.: Expanding the utility of semantic networks through partitioning. *Proceedings of the Fourth IJCAI*, pp. 115-121 (1975).
- [8] Karsenty, L. and Brézillon, P.: Cooperative problem solving and explanation. *International Journal of Expert Systems With Applications* 4 (1995) pp 445-462.
- [9] Leplat, J. and Hoc, J.-M.: Tâche et activité dans l'analyse psychologique des situations. *Cahiers de Psychologie Cognitive*, 3 (1983) 49-63.
- [10] Livet, P.: *La Communauté Virtuelle: Action et Communication*, Editions de l'Eclat (1994).
- [11] McCarthy, J.: “Notes on formalizing context”, *Proceedings of the 13th IJCAI* (1993) Vol. 1, 555-560.

- [12] Pasquier, L., Brézillon, P. and Pomerol, J.-Ch.: Chapter 6: Learning and explanation in context.
- [13] Pook, S, Lecolinet, E, Vaysseix, G and Barillot, E, 2000. Context and interaction in Zoomable User Interfaces *AVI 2000 Conference Proceedings (ACM Press)* (2003) pp 227-231.
- [14] Schank, R.C.: *Dynamic memory, a theory of learning in computers and people* Cambridge University Press (1982).
- [15] Sowa, J.F.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA (2000).
- [16] Van Dijk, T. A.: Cognitive Context Models and Discourse. In Maxim Stamenov ed *Language Structure, Discourse and the Access to Consciousness* Amsterdam: Benjamins pp 189-226 (1998).