

Elaboration of the Contextual Graphs representation: From a conceptual framework to an operational software

Patrick Brézillon¹

¹ University Pierre and Marie Curie (UPMC), Patrick.Brezillon@lip6.fr

ABSTRACT. The paper presents the current status of research on context modeling and management that emerges from several real-world applications since 25 years. We now have a robust conceptual framework and its implementation as a piece of software called Contextual-Graphs (CxG) platform. The context modeling and management were limited to few relevant aspects of context in each application. Working on several applications imposed to develop a robust conceptual framework and software that are independent of applications. The CxG formalism is the syntax and the domain models are the semantics. The CxG formalism relies on four elements (action, contextual element, activity and executive structure of independent activities) and about 200 models were developed as contextual graphs. This success comes from the fact that the CxG formalism is used for modeling human behaviors during task realization, not task models. The robustness of the CxG formalism offers challenging perspectives in crucial domains like weak-signal management in emergency rooms or crisis or security problems.

KEYWORDS. context modeling, context management, contextual graph, contextual element, human reasoning, Context-aware applications, Subway monitoring, Battlefield simulation.

1. Introduction

Context plays an important role since a long time in such domains where reasoning intervenes as in understanding, interpretation, diagnosis, etc. The reason is that the activities of reasoning rely heavily on a background or experience that is generally not make explicit and gives a contextual dimension to knowledge. Indeed, everybody uses context in his daily life as Mr. Jourdain used prose for speaking without being aware of it [46]. However, there is no consensus on what context is [10, 2].

The research presented in this paper relies on research about context modeling and management that emerges from several real-world applications since 25 years. The context modeling and management were limited to few relevant aspects of context in each application:

– Enology is a domain where the wine making process is strongly dependent of far event like the weather few days before the harvest had a great impact on the quality of the fermentation. Contextual elements far from the focus may be the first causes of a problem.

– In power plants, the functioning of control systems in an extra high voltage substation depends of the region where the substation is. Control-system management in lowland is not the same as in a valley embedded in the mountains.

– Car drivers often try to interpret the Highway code. Procedures are developed for helping people, but people prefer to transform procedure to fit the specificity of their personal context.

– Medicine is a domain poorly formalized and experts behave generally as artists because there are too many contextual elements to take into account.

– Battlefield is a domain where decision at the strategic level relies on ground information at the operational level.

– Decision-making is a two-step process, with, first, identification of the context and, second, decision making. Solving a problem requires to know the context of the problem.

- Interaction during a collaborative work relies essentially on a shared context in the group.
- Assembling information from different heterogeneous sources supposes context management for making compatible information pieces coming from different sources.
- The context-based configuration of a platform for an organization showed the need to adapting the tool to users and not users to the tool. Case of a platform for an organization (with size but also information like growth of the enterprise).
- Training supposes to contextualize the topic through different concrete example.
- Modeling the functioning of an organization for helping employees to react correctly in incidental situation supposes a contextualization of the work for monitoring the quality of work.

The goal of this paper is to make a synthesis of 25 years on modeling and managing context in real-world applications in the formalism of the Contextual Graphs. Our work relies on several applications in different domains that lead us to study specific aspects of context in each application. This approach obliges to maintain a coherent conceptual framework for studying context relatively independent of the domain. The goal was the modeling of users' behaviors while realizing a task, not the design and development of a model of the domain task. For this reason, we use different terms for speaking of the end-users depending on the applications: the term operator is used in technical domains; actors and experts for less formalized domains like medicine. The success of our approach comes from the play of two frameworks, namely a conceptual framework and an operational framework, leading to a software piece called Contextual Graphs (CxG) that is used directly in a number of different real-world applications (the results will be presented in a companion paper). The key of our success is an operational definition of context obtained by observing users in action with the importance of elements often far from the focus. It is with the SART project that we began to make more concrete our ideas and arrive to the Contextual-Graphs formalism and its implementation.

Hereafter the paper is structured in the following way. Section 2 presents rapidly the SART project that was the starting point of our study on context modeling and management by pointing out context as the key difference between procedures and practices. Section 3 explains the origin of the Contextual-Graphs formalism and why it was important to consider Contextual Graphs as representation formalism whereas a specific contextual graph is a model of a task realization (and not a domain model). Section 4 introduces the key notions of the conceptual framework, that is, the notion of contextual element and the distinction with its instantiation for a specific situation in a given model. Section 5 describes the main characteristics of the CxG platform, the piece of software born in the conceptual framework.

2. The SART project

2.1. Introduction

The SART project (1996 – 2002; SART is the French acronym for support system for traffic control) aimed at developing an intelligent decision support system to help the operators who control a subway line to react to incidents that occur on the line [6, 48]. Operators contextualize their operational knowledge and store it in an adapted computational structure, which will be easily understood by other operators and efficiently used by computer systems [10]. Possible functions of such systems are acquiring knowledge incrementally from operators; simulating train traffic on the line, possibly with incidents; changing the line model on operator's request for helping the operator to test alternative issues; proposing variants for an incident solving; training new operators not familiar with a given line; etc. All these functions are highly context-dependent.

Incident management on a subway line is a difficult task mainly for two reasons. First, the overall strategy of the company is to rapidly return to a normal operating mode. This global goal underlies any

incident solving. Second, parameters that intervene when an incident occurs are numerous and come from several heterogeneous sources. Operators who monitor the process in the subway control room had just a few minutes (or seconds) when an incident occurs to analyze the situation, collect contextual information, identify the context in which the incident occurred, and make a decision [12].

2.2. Procedures at RATP

The subway company in Paris (RATP) faces daily numerous incidents since 1900. Most of the incidents are well known (object on the track, lack of power supply, etc.). Thus, the company has established mandatory procedures for incident on the basis of its experience for supporting all the concerned actors. The procedure and the knowledge involved in the incident "Sick traveler in a train" supposes that the driver stops at the next station because travelers are safer in a station than in a tunnel. At a deeper level, the reason is that the driver has to avoid stopping the train a long time in a tunnel because some travelers may have behavioral troubles such as claustrophobia and could leave the train to wander about on the railway (and thus may generate another type of incident such as "Traveler on the railway").

Security sake and the willing of incident-solving standardization pushed the head of the company to compare reports written on incident solving for establishing secure procedures for each type of incident with the elimination of details and particularities existing in operators' practices that are irrelevant for the incident solving. Thus, a procedure is supposed to be a safe action sequence for solving a given type of incident in any situation, and more exactly in any context. For this reason, procedures rely mainly on the natural structure of the domain knowledge. This decontextualization by abstraction leads to robust procedures covering a large class of similar incident solving, but a counter effect results often in sub-optimal solutions for any particular incident solving. As a consequence, procedures are useful guidelines for operators, but they adapt them (more or less officially) for each new incident situation [12]. Thus, a part of the operational knowledge is not captured in procedures.

For simple incidents, a unique procedure is sufficient for solving the incident. For example, the procedure developed for solving the incident "object fallen on the track" implies the procedure: power cut-off, an employee of the RATP goes down on the tracks to retrieve the object, and once back on the platform, the power is re-established. However, there is no global procedure for complex incidents, only a set of partial procedures, each procedure helping to solve a part of the incident. For example, when a train cannot move in a tunnel, there are procedures to evacuate travelers at the nearest station, for clearing the damaged train by another train, etc. Some procedures are sequential, but others may be accomplished in any order or in parallel. For example, when a train must push a damaged train, both trains must be empty of travelers but the order in which travelers of the two trains are evacuated is not important and mainly depends on the context of each train. What is important is that the two actions must be accomplished before the damaged train clearing.

2.3. Practices at RATP

In high technical and heavily dynamical process regulation domains, operators who are responsible for monitoring the process have to rapidly react if an incident occurs. They have few minutes (and often few seconds) only to forge a representation of the issue, gather information on the situation, analyze the incident and undertake the corrective actions. The consideration of many contextual elements may lead to a large number of operational strategies, even for the same incident [14]. Operators prefer to plan their actions in real time rather than to rely on procedures for two main reasons. First, the selected procedure is not always perfectly adapted to the situation at hand and can lead to improper actions or sub-optimal incident solving strategies. Second, following a procedure may lead to miss some important facts and notice them too late for adequately solve the incident. For this reason, operators develop practices based on a use-oriented organization of the domain knowledge and their experience and, thus, use procedures as frames to construct a genuine strategy tailored to the specificity of a given situation.

Operators develop practices by contextualizing the procedure. A practice is the assembling of actions in a sequence that is performed by the operator to treat a particular incident in the specific context at hand. 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. It is a way of getting the result whatever the path followed [15].

For identifying the specificity of the situation at hand, operators postpone actions in order to collect as soon as possible a maximum of contextual information to have a general picture of the best strategy to apply. Thus operators elaborate a scenario mainly based on this contextual knowledge. One operator said us: "When an incident is announced, I first look at the context in which the incident occurs." The reason is that operators want to have a clear idea of future events on the subway line they are responsible in a first time. The purpose of this look-ahead reasoning [51] is to reduce, as far as possible, the uncertainty in the scenario. A scenario is a sequence of actions intertwined with events that do not depend on operators but constrain their actions. The scenario building corresponds to a contextualization of a part of the contextual knowledge that operators will make operational for assembling actions in a sequence. The built scenarios allow the access to common intermediary situations already encountered. They can thus reuse common strategies to solve the incident. As a consequence, the same action sequences are often found in a number of practices.

Contextual knowledge is always evoked by an event, such as a signal of power cutoff on a section of the subway line. Then, the operator reminds (more or less automatically) pieces of contextual knowledge such as type of day (e.g., working day), the period of the day (e.g., afternoon), the traffic state (e.g., rush hours) and the section load (e.g., very busy). Thus, the operator has a clear picture of the context in which the incident occurs. Indeed the operator solves the incident in a global way [23], that is, at a global level for acquiring contextual information and ensuring as much as possible a normal exploitation of the rest of the line, and at a local level for the incident solving itself.

More formally, given a practical unit of reasoning:

IF P1, P2, ... THEN C1

In a logical and theoretical reasoning (a procedure), the action leads to the conclusion C1 that is added to the base alone.

In a practical reasoning, the conclusion cannot be detached from the premises, i.e. cannot take meaning out of premises: C1 is added to the base linked to P1, P2, etc. This means that a piece of knowledge like C1 is defined in a context of use that is lost in the procedure, which is supposed to be applicable to a large class of problems when a practice is a contextualized version of the procedure in a specific context. As a side effect, a "best practice" is a practice associated with a specific context.

Nowadays, the multiplication and the specialization of procedures exist in a number of domains. This is a general trend overtaking the railway control process, which is also observed in aeronautics. In this later field, operators always adapt the actual procedures to the current situation. De Brito and Boy [29] explain that the aircraft operators prefer to re-plan their actions instead of following non-adapted procedures, even if the new plan is inspired of these procedures. Britanik and Marefat [25] also propose a plan merging methodology that merges partial-order plans based on the notion of plan fragments. Plan fragments encapsulate actions with their context. Hayes-Roth & Hayes-Roth [38] proposed a so-called opportunistic approach to planning. This non-hierarchical planning assumes that a plan is executed with the help of some kind of mental blackboard where pieces of information, relevant cues and possible sub-goals are stored. They claimed and showed that planning happens asynchronously and is determined by the momentary aspects of the problem. No fixed order of operations exists, the plan execution and the steps to be taken grow out of the problem stage at hand.

3. Modeling a problem including context

3.1. Choice of the representation formalism

During the modeling phase in the SART project, we tried to understand the troubleshooting strategies applied in some incidents, such as lack of motor power on a train. We start our modeling from manuals (knowing and know-how) used as a shared context by operators and train drivers during the incident identification and solving. These manuals contain the description of the procedures established by the company for incident solving in different situations, procedures that are expressed like “IF <situation>, THEN <procedure>”. The production-rule formalism thus was used as a first representation of the natural way in which experts make their decision (i.e. first they identify the situation and second they apply the action sequence).

Expertise in manuals is in a form close to the production-rule formalism with (1) description of a situation, and (2) the decision to make leading to a series of actions to execute. Table 1 gives an example of this expertise in the manuals for the incident “lack of train power”.

IF C1.1	THEN Actions 1, 6, 2 & 5
SI C1.2, C2.1 & C3.1	ALORS Actions 2 & 5
SI C1.2, C2.1 & C3.2	ALORS Actions 3, 7, 4 & 5
SI C1.2, C2.2, C4.1 & C5.1	ALORS Actions 1, 8, 7, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.1, C5.2 & C6.1	ALORS Actions 1, 17, 8, 7, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.1, C5.2 & C6.2	ALORS Actions 1, 12, 7, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.1, C5.3 & C6.1	ALORS Actions 1, 17, 8, 7, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.1, C5.3 & C6.2	ALORS Actions 1, 10, 7, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.2 & C5.1	ALORS Actions 1, 8, 9, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.2, C5.2 & C6.1	ALORS Actions 1, 17, 8, 9, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.2, C5.2 & C6.2	ALORS Actions 1, 12, 9, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.2, C5.3 & C6.1	ALORS Actions 1, 17, 8, 9, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.2, C5.3 & C6.2	ALORS Actions 1, 10, 9, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.3 & C5.1	ALORS Actions 1, 8, 11, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.3, C5.2 & C6.1	ALORS Actions 1, 17, 8, 11, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.3, C5.2 & C6.2	ALORS Actions 1, 12, 11, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.3, C5.3 & C6.1	ALORS Actions 1, 17, 8, 11, 13, 14, 15, 5 & 16
SI C1.2, C2.2, C4.3, C5.3 & C6.2	ALORS Actions 1, 10, 11, 13, 14, 15, 5 & 16

Table 1. Example of production rules for representing expert knowledge

The first rule means [48]: IF “the immediate reset of the train” (C1) = Possible (C1.1), THEN “Monitor the traffic out of the incident zone” (action 1), “Reset the train” (action 6), “Resume the service for travellers” (Action 2), and “Drive the train to the garage” (action 5). The differences concern more the identification of the situation than the process itself, the latter being generally expressed by a sequence of actions with only few specificity at the beginning of the sequences.

Indeed, the rule representation was not well accepted by operators because such a representation is not compatible with their mental representation of an incident solving. Most of action sequences are identical and well known by operators. Thus, rules differ by few (relevant) items and operators have a feeling of redundancy because repetition darkens the readability and understanding of the rules. Indeed, the specific parts of rules, which are at the beginning, are highly contextual. Rules represent different ways for arriving at the same final situation from different initial situations. On the practical side, rules with symbols like in Table 1 are not easily readable and their translation in natural language can lead to long texts. Moreover, it is difficult to maintain and update such a ruleset (with a risk of combinatorial explosion if we want to refine the context of the incident solving). The rules are supposed to be independent (the flat representation paradigm of a rule set) when they represent interdependent strategies, often a strategy being a refinement of another one (as shown by the few differing items between rules).

This limits the rule representation due to the lack of explicit context consideration in the representation. Thus context is often assimilated to the situation, although external events (e.g. “Immediate reset of the train”) may intervene too. The situation descriptions in the IF part of rules are supposed to be independent when they are interdependent. As a consequence, (1) C1 with the value C1.1 define a situation that is different from the situation defined by C1 with the value C1.2; and (2) the contextual elements C2 and C3 are considered only if C1 has the instantiation C1.2, and not C1.1. This confusion is due to the non distinction of the contextual elements and their instantiations.

Then we changed the rule set formalism for a decision tree representation based on two types of elements: actions, which are directives to do an action, and event nodes, at which a path is selected depending on the knowledge about the event. The main reasons for changing the representation are:

- A direct translation of a rule set in a decision tree representation,
- An explicit representation of the ordering of nodes, and
- A progressive identification of the situation at hand.

Figure 1 shows the decision tree representation of the procedure for “train lack of power” solving in Table 1 (the meaning of the boxes is not important here but can be found in [48]).

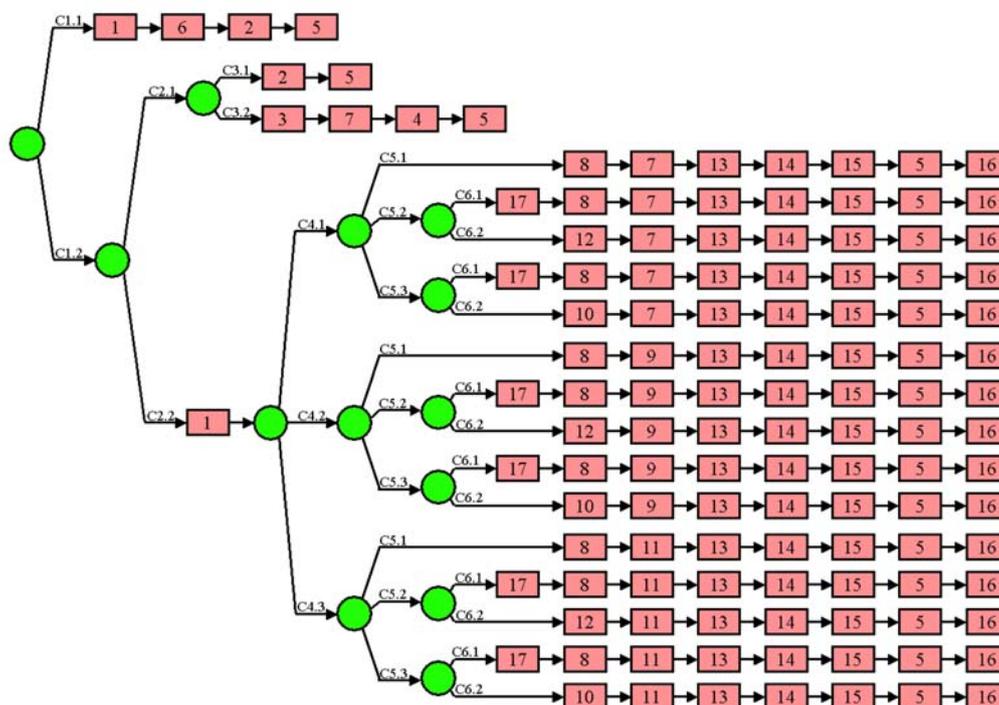


Figure 1. Tree representation of the procedure for solving “lack of train power” incident [48]

We can now precise the difference of meaning between the terms strategy, activity and practice. However, first recall that the task is what an actor must do, and the activity is the (physical and mental) behavior that the actor exhibits for realizing the task [55]. A task has a formal definition, but the notion of activity reveals practical aspects of the task realization in addition to the task model. Each activity is described in terms of scripts, each script being run in a specific context. Thus, an activity takes into account the result as well as the process leading to the result. A strategy is established at the level of the decision-making process. The activity is the application of the strategy by an operator at the tactical level, and the practice is the description of this activity at the operational level. Thus, one speaks effectively of the same thing, but at different decisional levels.

3.2. Evolutions of the tree representation

Our tree representation is inspired by decision trees, but differs on three points. First, a decision tree (representing a procedure) is often established in a top-down way (more general elements are considered first), while our tree (representing practices) is built in bottom-up way by incremental addition of practices when the incident solving occurs in a new context that implies a new strategy. Second, our trees have no decision node, only “event” nodes (the green circles on Figure 1) where an event is analyzed to select the path corresponding to its value at that moment. Third, there are no probabilities on events because we describe, with the maximum of parsimony, all the contexts encountered when a decision was made on different incident solvings. Then the first step is to make context explicit.

We replace the term “event node” by the term “contextual node” and the term “nature's state” by the term “context” [11]. The change implies a clear separation of the contextual node and its values. This means that when undertaking an action, operators consider that, due to the contextual information they identified, the state of nature between the root and the action undertaken is the true state of nature at that moment of the decision making. The nature of each event at an event node carries on a part of the uncertainty of the situation. There are two ways to manage uncertainty, first, by assessing some probabilities for each event, which is the usual view in decision theory, second, by considering that two events are possible depending on the context. Our representation provides an answer for any possible context whatever the probability is.

Indeed, each path in the tree representation corresponds to an effective practice developed by an operator for the incident solving in a particular context at hand, and this explains the small differences of 1 or 2 actions between practices. Thus, our trees model a task realization (i.e. an activity or a practice), which depends on the context, while a decision tree corresponds to the (unique and theoretical) model of the task.

In the SART project, the global goal of operators is “to reestablish a normal traffic as rapidly as possible, respecting security rules” and the branches of the tree express only different means to reach the final goal according to the states of nature [51]. Thus, the result depends heavily on the context, and several strategies can be used for solving the incident in a given situation. As a consequence, the number of branches may be very important [9].

For finding the right strategy to apply in a given context, operators prefer to gather a maximum of (contextual) information before making their decision and, thus, postpone most of the actions to the end of the branches of the tree. Operators use contextual knowledge to circumscribe a subset of possible states because the whole number of possible states is potentially very important [10]. The problem of the operator is to get an adapted answer as soon as he knows what the context is. Thus, the operator first must diagnose the exact situation from the contextual information available and, second, apply the corresponding action sequence, i.e. a practice as a kind of contextualized procedure). Each branch actually describes a contextual knowledge, which becomes more and more accurate as long as the branch is followed. We are interested by the possibility for the operator to determine, as soon as possible, which strategy (path in the decision tree) to apply and what the next action to undertake is [11]. It is important to note here that it is an exclusive choice of a strategy for each particular context.

However, the tree structure presents a high redundancy of item because practices differ on few items only. Thus, such a representation does not allow the representation of highly contextual decision-making in complex applications. A second change concerns a simplification of the tree representation at the action level in order to obtain a manageable structure for representing operational knowledge on incident solving on subway lines.

Generally, a number of paths in a tree terminate by the same sequence of actions (see the sequence 13, 14, 15, 5, 16 on Figure 1). It is the result of the postponement of the decision as far as possible in

two steps. First, operators try to replace uncertainty about events by the identification of the current context as precisely as possible. Second, by replacing action sequences by macro-actions, operators try to facilitate the mapping between context and action, each macro-action being supposed to be adapted to one or several contexts. Macro-actions are a way to proceduralize contextual knowledge and to introduce modularity in the diagnosis process by managing different modules accomplishing the same function in different contexts [7].

Macro-actions realize a kind of compilation of several actions originated from experience. A part of the knowledge about each action and their ordering become implicit in the compilation. However, for explaining a macro-action (and thus the whole reasoning involved in), an operator needs to decompile the macro-action and goes back to the ordering of actions in the sequence for retrieving the rationale. Such an operation is not always easy, especially when experience comes from previous generations of operators that appear in addition as an ordered sequence of instantiated contextual nodes before the macro-actions [52].

Concretely, the definition of the macro-actions relies on common sub-procedures known by the operators, such as “linking trains,” “return to the terminus without travelers”, etc. We will see later that a macro-action is a type of elementary activity. Figure 2 shows the simplification introduced by the use of macro-actions in the tree representation on Figure 1 where, for example, the action sequence 13, 14, 15, 5, 16 is replaced by the macro-action MA3.

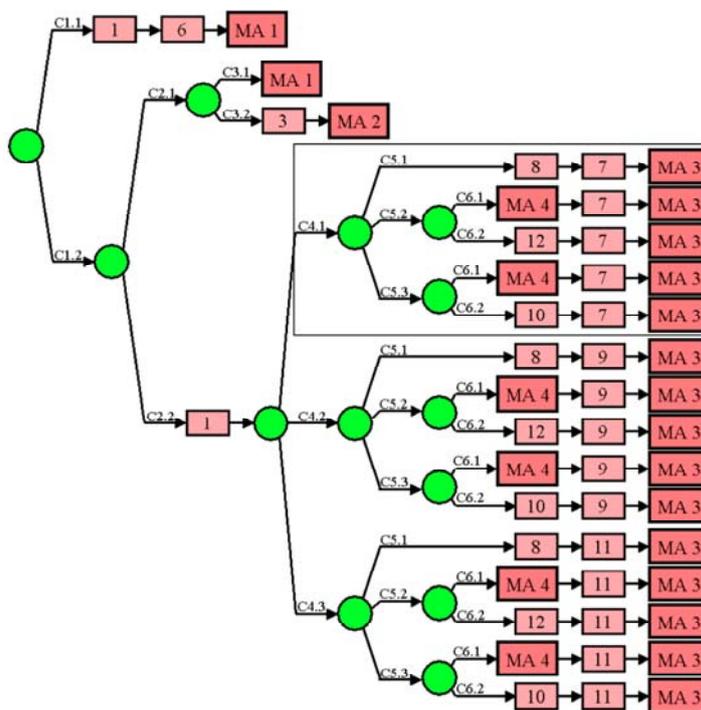


Figure 2. The tree representation with macro-actions [48], the rectangle corresponding to Figure 3

This first change allows the simplification of the description of a practice, but does not help the practice management itself (the combinatorial explosion of branches staying). The next change concerns the redundancy of items among practices as represented by the similarity of branches in Figure 2.

As seen on Figure 2, the replacement of action sequences by macro-actions does not reduce the number of occurrences of macro-actions in the tree. Thus a natural simplification of the tree representation (at a computational level) is to consider that all the branches, which finish by this same finale macro-action, can be merged just before the macro-action at a recombination node. The first important consequence is the abandon of the tree representation for a graph representation, and we will

speak now of path instead of branch in the representation. The graph representation can be built according to two viewpoints, namely logic of functioning and logic of use [54]. In logic of functioning, a designer prefers the use of the principle of parsimony of the representation (computationally speaking) and considers that all the branches ending with a macro-action can be merged before at a recombination node. In logic of use, operators look for preserving the understanding of the contextual information in the graph representation and thus associates a recombination node to the contextual node from where are coming branches and accepts a reduction of the number of occurrences greater than 1. We discuss now these two options on Figure 3 that represents the part of the tree in the rectangle on Figure 2.

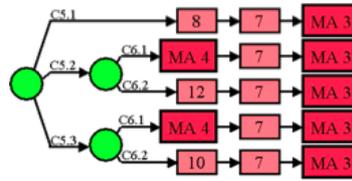


Figure 3. Part corresponding to the rectangle on Figure 2

According to the designer’s viewpoint (i.e. logic of functioning), all the branches ending on the same macro-action can be merged at a unique recombination node. In such an alternative, the recombination node is associated with the macro-action that thus exists once in the representation without consideration for the context of each path (i.e. the identification of the situation). It was the technical solution chosen initially [48], but also in the workflow community where contextual and recombination nodes are considered as two independent items [32]. Figure 4 shows the transformation of Figure 3 based on this logic of functioning (with the graph on Figure 2, 15 branches will be merged on a unique recombination node just before MA3).

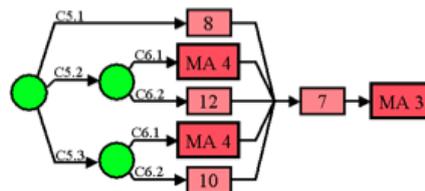


Figure 4. A graph representation in the logic of functioning [48]

According to operators’ viewpoint (i.e. in logic of use), one first notes on Figure 3 that there is the same action 7 also just before the macro-action MA3 on all the branches coming from the contextual node C5 on Figure 2. However, when looking to the other equivalent subtrees on Figure 2, action 7 is on the branch associated with the value C5.1, but it is action 9 on the branch that is associated with the value C5.2 and action 11 on the branch associated with the value C5.3. The natural conclusion is to associate the recombination node with the corresponding contextual node (the first occurrence of the contextual node 5 on Figure 2). Then, one concludes on the representation given on Figure 5 (drawn with the CxG software that is presented hereafter).

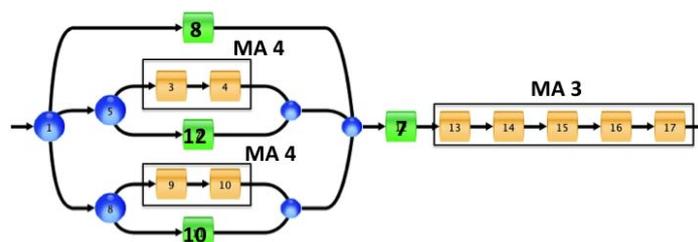


Figure 5. A graph representation in the logic of use (macro-action are in yellow).

The size of the structure is now under control and the consideration of a new contextual element will add few elements in the graph, but do not increase drastically its size as previously. Figure 6 presents the graph representation with the association of contextual nodes and recombination nodes of the tree representation presented Figure 2.

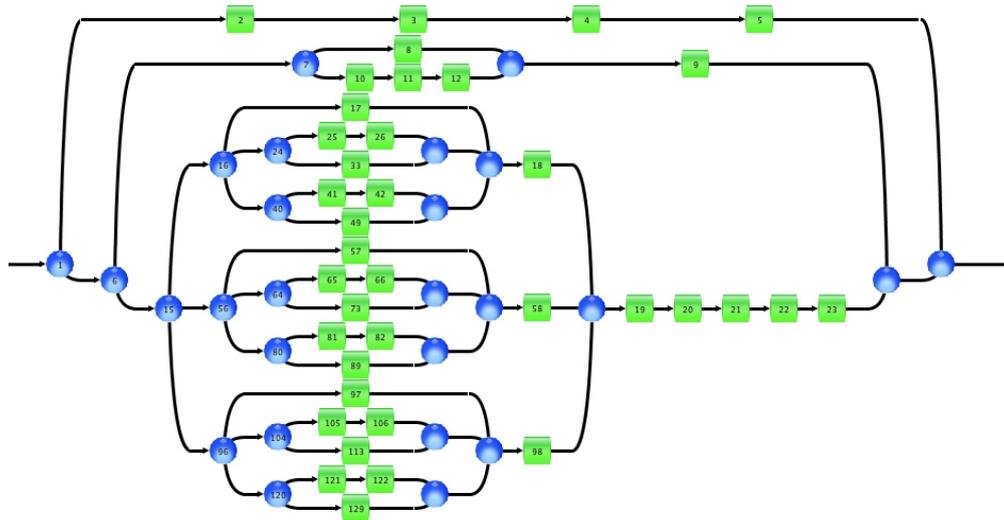


Figure 6. Graph representation equivalent to the tree representation on Figure 2

The perspectives opened by the logic of use are the following:

- The coupling of a contextual node and a recombination node at the implementation level allows the introduction of the notion of contextual element at the conceptual level.
- Pairs of contextual and recombination nodes appear in a contextual graph either one after the other, or one on a branch of the other, like in a series-parallel structure.

We discuss now these two points.

4. The conceptual framework

4.1. Contextual element

A pair {contextual node, recombination node} at the implementation level corresponds to a contextual element at the conceptual level. A contextual element may come from different heterogeneous sources that can be put in four categories, namely the user, the task, the situation and the immediate environment. A contextual element corresponds to an information piece that must be analyzed and that is translated by a question at the contextual node about nature's state. The value taken by the contextual element when the focus is on it—its instance—is taken into account as long as the situation is under analysis. The consequence is the selection of a branch of the contextual element and the execution of the corresponding activity.

Figure 7 shows how the term *Contextual Element* (CE) refers to pieces of data, information or knowledge that can be used to define the context [16].

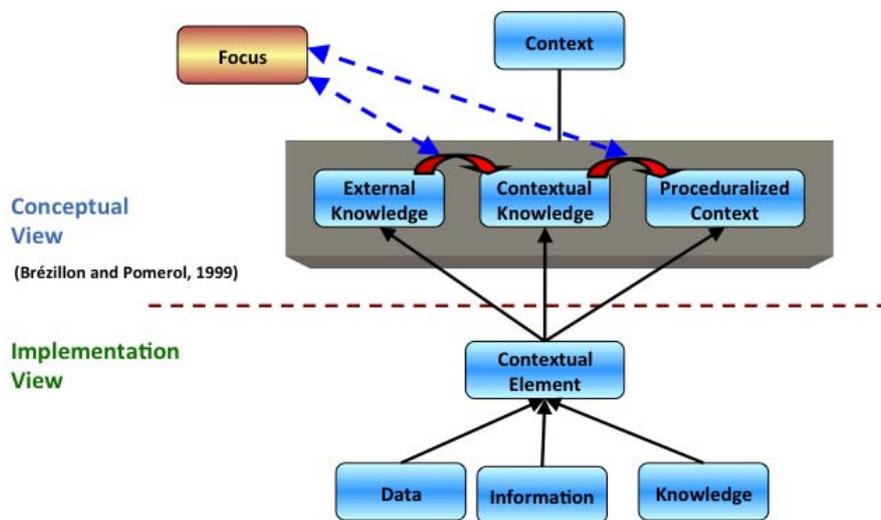


Figure 7. Position of contextual element in the conceptual framework

This separation is necessary because contextual knowledge comprises what is in the users' minds and thus is too abstract. To treat context computationally it is important to make this distinction between contextual data (e.g. location coordinates, identification and temperature), contextual information (e.g. weather is hot, nearby person) and contextual knowledge (e.g. understanding about the weather behavior in different regions).

4.2. Values and instantiation of contextual elements

Since the first representation in rule set (see Table 1), the instantiation is made explicit, but the rationale behind this choice is left implicit because the context is assimilated to the situation (i.e. its instantiation). Indeed, the values (e.g. C1.1 and C1.2 on Table 1) are more visible than the contextual elements themselves. When exiting a contextual element at the recombination node, the instantiation of the contextual element just exited does not matter anymore. A contextual element can have more than two values because the corresponding question is not necessarily a Boolean expression. The explicit introduction of contextual elements and the distinction with their values offer new perspectives in the representation: Values may be added incrementally when new practices are introduced ; Type of values also may varies ; and, Different ranges of values are possible. For example, the value of the contextual element « Temperature? » may have values like « hot » and « cold » and « warm » may be added later, but quantitative values like « 24,5°C » are possible instead of « warm » according to the focus. Now, if we take into account the actor's viewpoint, 24°C in winter is particularly hot in France while it is cold in Brazil because the temperature in winter at Rio de Janeiro is more around 32°C. The lesson learned here is to make clear that context must be modeled and managed from the viewpoint of a particular actor.

The distinction between a contextual element and its values is important for the reuse of experience because the difference between two contexts may come from (1) different instantiations of a given contextual element or (2) different contextual elements in the two contexts. Indeed, it is the contextual element that corresponds to the event node compared previously to the contextual node. This means that instantiation of the contextual elements needs to be done only in real time conditions, especially because the branches of a contextual element are exclusive (i.e. contextual elements in a graph belong all to the contextual knowledge, but only those on the path followed for developing the practice will belong to the proceduralized context).

The working context is the set of all the contextual elements in a contextual graph, their values and their current instantiation [22]. In a context-based representation, any practice that is developed in the working context of a focus of attention is the "best practice" because the practice is built at the same time it is used, thus taking into account all that occurs during this process. The working context plays a

central role in the modeling of collaborative work for introducing a cycle of use of a contextual graph (that is directed) and the possibility to modify instantiations of the contextual elements.

4.3. *Focus and types of context*

Brézillon and Pomerol [8] propose the following operational definition of context: "Context is what constrains the focus without intervening in it explicitly." The focus of attention depends on the evolution of a task realization (the incident solving in the SART project) as well as the progression of the actor's reasoning during the task realization. This definition has two main consequences: (1) one cannot speak of context in an abstract way, context is always relative to the focus, and (2) context and focus are interdependent. The context makes the focus explicit and the focus defines the contextual elements of the context that are relevant.

According to the focus, context has two parts, namely, contextual knowledge and external knowledge [15]. Contextual knowledge constitutes a kind of tank of elements that are related to the focus in a flat way. External knowledge concerns elements that are not important for the focus at the moment we consider it. However, the frontier between contextual and external knowledge is porous and evolves with the progress of the focus. Making context explicit supposes to consider actor's activity as a process, and not just the result of the activity. Such a conceptual framework supposes a uniform representation of the elements of knowledge, reasoning and context that are needed in a task realization.

Contextual knowledge is personal to an actor and it has no clear limit [44]. Contextual knowledge is evoked by situations and events, and loosely tied to a task or a goal [10]. It is something that is stored in long-term memory, and recalled as a whole, as a viable unit of a task appropriate to some step in a decision making [52]. The actor's focus is guided by the mental representation that the actor has of the task in its current step, the situation in which he is and the local environment in which resources are available. Contextual knowledge is more or less related to the mental representation of the operator and implicitly delimits the resolution space [19]. The mental representation expresses the individual context in which an actor interprets the focus. The mental representation depends on expert's experience with a task realization. Experience contains knowledge accumulated by the expert during his practical use of the domain knowledge along a number of realizations of the same task in different contexts. Thus, experience relies more on operational knowledge than domain knowledge and, thus, is highly contextual.

Numerous definitions exist on the Web across disciplines. Bazire and Brézillon [2] proposed a model of context from a corpus of 150 definitions (in 2005 and 250 in 2016) of context pick up on the Web from various disciplines such as computer science, philosophy, economy, business, HCI, etc. In our approach, half of the corpus (66 definitions) were used for context modeling, and the second half was kept for validation of the model (Selection of definitions has been made by classical statistical means). The most represented areas in the database are: artificial intelligence (39), documentation (27), cognitive ergonomics (9), cognitive psychology (6), business (4), philosophy (2), linguistic (4) and one in education sciences, another in medicine and the last one in neuroscience (some definitions are multidisciplinary).

Each definition is entered in the database according to a frame containing:

- the definition
- the object about which context is defined
- the domain from which the definition has been found
- the reference (URL or bibliographical reference)
- eventual comments or complements to the definition.

For example:

Definition 2:

Context is what constrains a step of a problem solving without intervening in it explicitly

Item: A step of a problem solving

Domain: Artificial Intelligence, Decision Support System

Source: Brézillon P. (1999) Context in problem solving: A survey. The Knowledge Engineering Review, 14(1): 1-34.

Comments:

This definition is similar to the definition “All that may influence a given process whom first causes are known” [30]. The corpus presents nevertheless a sufficient variety: "Context is what constrained a problem solving without intervening in it explicitly" [8]; "Something that growths and change as a function of the time , user's requests and the growth intelligence of the system" [45]; “An ordered set of modules which can change during the process of proving a formula” [3], etc. This variety arises from the fact that there is no absolute context, context being relative to something. As a consequence, context must be studied according to its use.

By abstracting key elements of the definitions we thought to develop a resulting definition able to cover a large class of definitions. We analyzed this corpus of definitions through two methods, namely LSA [41] and STONE [50], and we conclude that finally the content of all the definitions can be analyzed in terms of few parameters like constraint, influence, behavior, nature, structure and system.

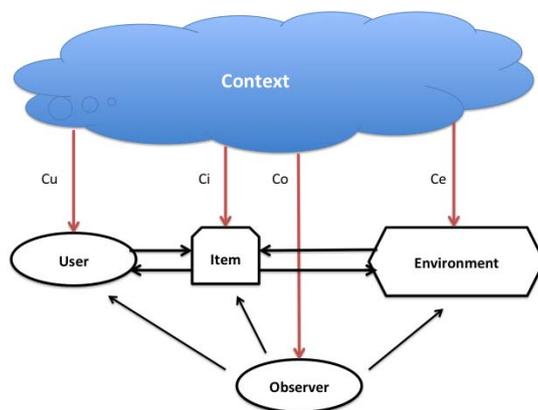


Figure 8. Our proposal for a model of context

The result of this study is shown on Figure 8 where context can be specified for a given situation by the answering of the following questions : Who? What? Where? When? Why? and How? “Who” indicates the subject, the agent of the action. “What” represents the object, the patient that sustains the action. “Where” and “When” give information about the spatio-temporal location of the considered action. “Why” gives the intentions, the goal (and eventually the emotions) of the subject. At last, “How” makes explicit the procedure needed to realize the action. Traoré et al. [58] presents a use of this model in an application in car-drivers’ modeling.

At each step of the actor’s reasoning, a sub-set of the contextual knowledge is proceduralized for addressing the current focus. This “proceduralized context” is built from elements of the (highly heterogeneous) contextual knowledge that are invoked, assembled, organized, structured and situated according to the given focus [22]. The proceduralized context is ready to be used for action. A proceduralized context is quite similar, in the spirit, to the chunk of knowledge discussed in SOAR [56,

39], and, in its building, to Clancey's view [26] on diagnosis as building of a situation-specific model. A proceduralized context is like a local model that accounts for a precise goal in a specific situation (at a given step). This notion is relative to each operator, to the current situation and to the moment at which the operator is realizing the task. This "chunk of contextual knowledge" is the contextual knowledge activated and structured to make diagnosis or decisions. If the contextual knowledge is background knowledge for the focus, the proceduralized context is foreground, that is, immediately useful at the given step of the focus [14]. In a distinction reminiscent to cognitive ergonomics [42], we could say that the contextual knowledge is useful to identify the activity whereas the proceduralized context is relevant to characterize the activity of a particular actor [17].

Indeed, a proceduralized context is more than a "chunk of contextual knowledge" that is stored in the contextual knowledge, because it is possible to retrieve its building, the reasons behind the choices (the contextual elements considered and their instantiations as contained in the proceduralized context), the alternatives abandoned (the values of the contextual elements not retained and the corresponding abandoned actions), etc. In a contextual graph, the proceduralized context of an action is the ordered sequence of instantiated contextual elements from the input of the contextual graph up to that action.

Proceduralized-context building is a kind of learning that results of an accommodation process. The proceduralized context could be recalled later either as a whole (like a part of a new proceduralized context) or explained in terms of the way in which it has been built and can be reused in the new proceduralized context. This type of learning is structured by the contextual knowledge, and the more a person is experimented, the more the person possesses available structured knowledge, i.e. chunks of contextual knowledge.

The dimension of context is infinite [44], but there are a limited number of contextual elements in a contextual graph because the number of practices is finished in a contextual graph. The main reason is that the number of methods that can be used for realizing a task is limited.

4.4. Related representation formalisms

In AI, context was first discussed by McCarthy [43] who argued that contexts should be treated as first-class objects with a formal set of axioms that govern their interactions with one another and with all the other features of logic. Contexts are rich objects in the sense that they cannot be completely described. Context permits to make explicit all what is implicit in a proposition. However, the logical machinery is only a small fraction of the effort involved in building a context-based system. The bulk of the effort lies in writing the axioms describing and interrelating contexts. The structure and content of these axioms--the lifting rules--are heavily dependent on the kind of use. The most common operation on contexts is to lift a formula from one context into another. Doing this requires relative (partial) decontextualization, i.e., the differences between the origin and target contexts had to be taken into account to obtain a formula with the same truth conditions as the original formula had in the origin context. The problem is augmented when we consider that there are different types of interdependent contexts. Giunchiglia [34] considers a set of discrete contexts (then, at a same level) and the relationships between them (through bridging rules for entering and leaving contexts). As a consequence, context makes reasoning local. Both of them consider context at the representation level.

McCarthy [44] defined a context as a generalization of a collection of assumptions. Contexts are formalized as first class objects (formal objects), and the basic relation is $ist(c,p)$. It asserts that the proposition p is true in the context c , where c is meant to capture all that is not explicit in p that is required to make p a meaningful statement representing what it is intended to state. Formulas $ist(c,p)$ are always asserted within a context, i.e., something like $ist(c', ist(c,p))$: c' : $ist(c, p)$. The lessons learned of this work are:

- a context is always relative to another context,

- contexts have an infinite dimension;
- contexts can not be described completely;
- when several contexts occur in a discussion, there is a common context above all of them into which all terms and predicates can be lifted.

There are several formalism like Contextual Graphs : production rules, UML, Bayesian networks, Petri nets, fluence graphs, etc. The difference between model and representation formalism can be illustrated on the well-known example in AI of a paper submission by an author of a journal from an initial « model » of Torres et al. [57] as shown in Figure 9:

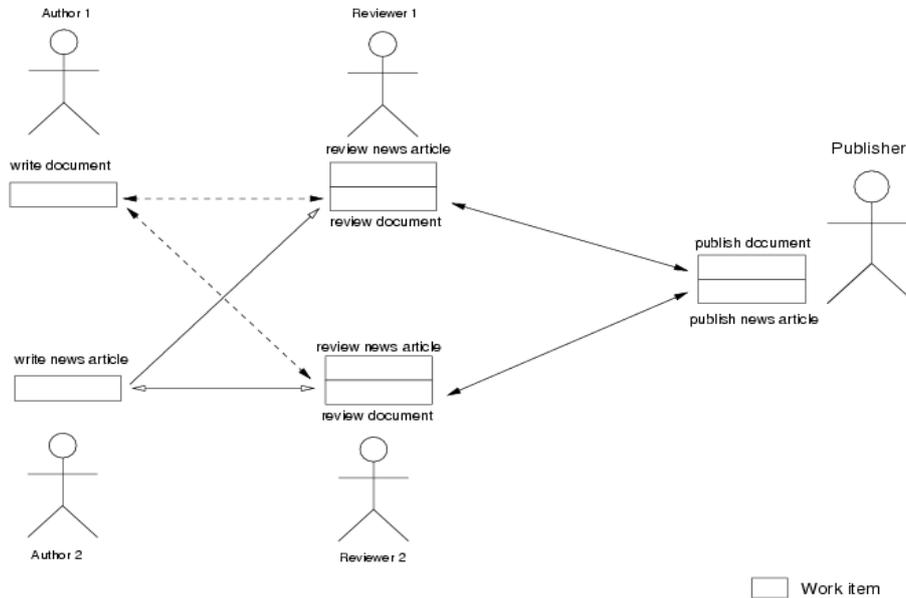


Figure 9. The paper-submission example [57]

Figures 10 and 11 give the representation of this model in Petri net and in UML respectively

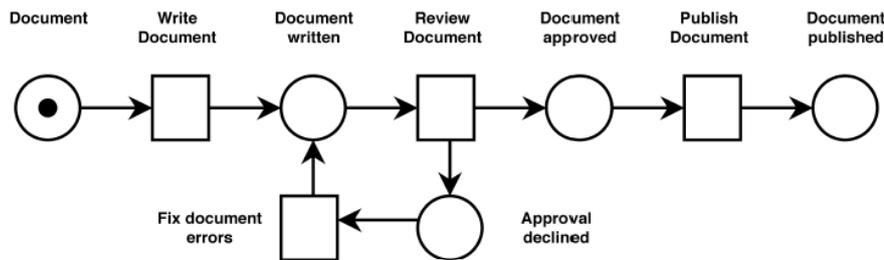


Figure 10. Petri-net representation of the submission-process example

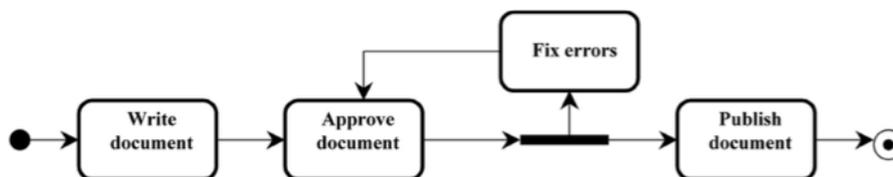


Figure 11. UML representation of the submission-process example

And the representation in contextual graph is given on Figure 12.

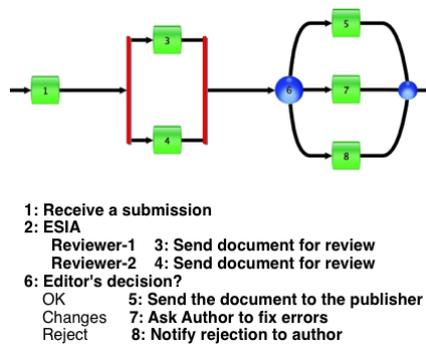


Figure 12. CxG representation of the submission example according to the editor's viewpoint (see explanation on the CxG formalism in section 5.1)

It is 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. However, each formalism has advantages and disadvantages, Contextual Graphs are strongly user-oriented, not the others, and translation from one formalism to another one is not biunivocal.

5. The operational framework

5.1. The Contextual-Graphs platform

The evolution from a tree representation to a graph representation is not a graphical simplification only. The notion of branch in the tree representation implied a progressive refinement of the practices along the branches. In the contextual graph, a path represents a practice too, but all the practices are now intertwined by the play of the contextual elements that structure the practices in a contextual graph [18]. Contextual elements structure experiences (practices) differently, on the one hand, from the knowledge bases of expert systems represented in a flat way because context is not represented explicitly, and, on the other hand, from knowledge organization in an ontology where links between concepts depend on the domain (is-a, kind-of, etc.) while elements in our context model concern the operator, the task, the situation and the local environment [24].

The contextual graph is a directed acyclic graph with exactly one root and one goal because operators have only one goal and branches express only different strategies, depending on the context, to achieve this goal. The representation of a contextual element as a contextual node and a recombination node makes that the structure of contextual graphs is close of the structure of series-parallel graphs. Formally, a directed graph (digraph) consisting of two vertices u and v and a single edge from u to v is a series parallel digraph [4, 5] with source u and sink v . Series-parallel graphs are used in a wide range of application, such as electrical circuits [31]. The main lessons learned from this application of series-parallel graphs are that (1) vertices in series-parallel graphs correspond to four items in contextual graphs (action, contextual element, activity and ESIA); and (2) electricity cross all the branches after a vertex, while the reasoning in a contextual graph when arriving at a contextual node (the equivalence of the vertex) is to follow one branch only. Thus, contextual graphs are not totally ascribed in the series-parallel formalism of representation used to model electric circuits. Contextual elements are organized in contextual graphs in the spirit of “nest of dolls” (matriochka) and always defined (embedded) with respect to another contextual element. Thus, contextual elements never overlap, and a contextual element is itself in a context (the onion metaphor in [6]).

We developed a software for exploiting the formalism of contextual graph. It presents usual functionality as: switching between different languages, identification of the user, enrichment and correction of all texts, an online help, different types of visualization, etc. [13]. Contextual graphs are acyclic due to the time-directed representation and guarantee algorithm termination. Each contextual

graph (and any sub-graph 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.

We use the syntax define in Figure 13 for all the figures representing a contextual graph [15].

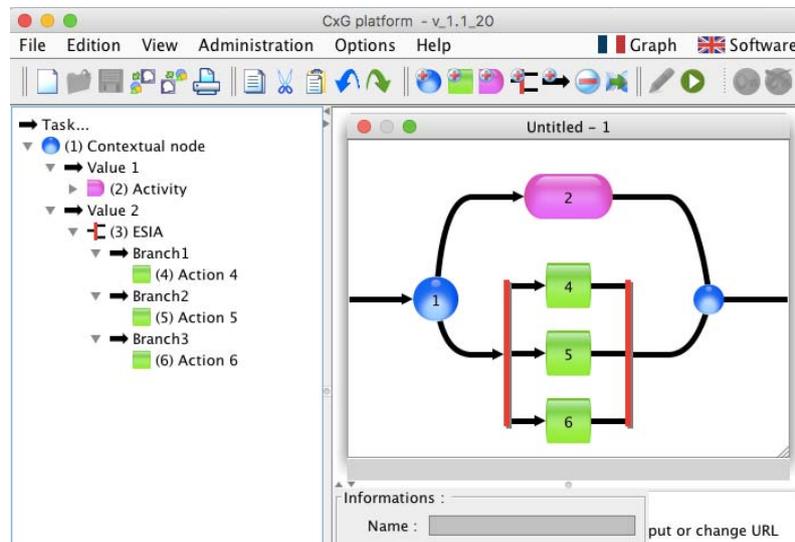


Figure 13. *The four elements of a contextual graph*

The software is written in Java and available at cxg.fr under GNU license [18]. It is an interface used by actors wishing to edit a contextual graph, reading practices for selecting the best one in his working context, browsing alternatives of a practice, exploring a contextual graph at a different granularity, analyzing contextual information attached to each item (date of creation, comments, etc.). The mechanisms of aggregation and expansion of activities (at a semantic level) allow the user to focus on one part of the contextual graph or another according to his focus of attention, but also with the possibility to hide some parts of the graph (at the syntax level). The user can define a part of a contextual graph as a sub-graph, and the system can then offer the possibility to aggregate it in a node in the contextual graph. It is thus possible to study parts of a reasoning and all the variants (the practices). This is particularly interesting to understand the variants between two practices, the role plays by a contextual element in the choice of an action instead of another, etc.

Design and development of the software is user-centered for an intuitive use by non-specialists in computer science and mathematics. Interesting functions of the software are the possibility to link an item to an external document (Word, PDF, Web page, etc.), to run an external piece of software, to have the trace of the practice development as a roadmap for an employee, etc. (This was used in the medical application, see [1]). The functioning of the platform itself and the management of items can be described in a contextual graph too as used in context-based intelligent assistant systems [22]. An activity—being itself a contextual graph—is stored as an independent XML file. In the SART project [53], operators used contextual graphs to replay how a colleague solved a given incident, look for alternatives, analyzing the working contexts, and even may study new practices.

5.2. Revisiting notions

We first must distinguish the notion of activity for the operator and for the implementation. At the implementation level, an activity is an extension of a macro-action, i.e. a sequence of actions without contextual elements. An operator's activity corresponds to the realization of a task integrating operator's preferences and experience, task characteristics, the situation in which the operator realizes

the task and the local environment, that is, a contextual sub-graph. An activity is similar to a complex action. Again, we distinguish clearly the task and the activity as [42], and we focus on how the task is realized, not on the task model.

The dual aspect of an activity—as an item in the main contextual graph and as a subgraph itself—offers interesting perspectives. An action may be simple for one actor and a complex activity for another actor. For example, when an operator decides that a train must be empty of travelers, it is a simple action for the operator, but a complex activity for the train driver who has to execute several commands and to establish a strategy to manage different possible scenarios. Thus, both the operator and the train driver give the same meaning at an action, although their interpretations may differ [14]. The two actors interact with views at different granularity levels, the operator considering the activity as a unit and the train driver considering the activity as a contextual sub-graph (i.e. a complex action for him).

Inside a contextual graph, it is possible to define sub-graphs as activities (this rule is applicable recursively). The notion of activity allows modularity in a contextual graph that, on the one hand, makes easier the management of a complex task (e.g. see [23]) by taking into account the natural modularity of the domain, and, on the other hand, open the door of the contextual-Graphs formalism for modeling collaborative work [33]. In the previous example, “make empty the train of travelers” is a particular activity of the incident solving “Evacuate a damaged train”, and of a number of other incident solving too because the traveler security is the mandatory policy of the company. At the implementation level, such an activity corresponds to a given sub-graph that appears several times (i.e. in different contexts) in the contextual graph in order to simplify the whole contextual graph.

Another advantage of contextual graphs is their ability to deal with large structure usually encountered in industrial applications. This representation is easily understood by the operators (and designers) and well accepted by them and their hierarchy because the representation is more activity-dependent than domain-dependent. The division into sub-graphs makes sense. This flexibility and modularity is another advantage of the contextual graphs introduced by the notion of activity.

Another limit of the tree modeling concerned the execution of independent activities. The order for executing such activities does not matter (the two activities can also be executed in parallel too), but both activities must be executed before to continue the development of the practice. For example, when a train must push a damaged train on a subway line, both trains must be empty of travelers but the order in which the trains are emptied is not important and mainly depends on the individual context of each train. However, both trains must be empty of travelers before to be assembled. A literal representation would include all the variants (2 branches for two actions, but 9 branches for 3 actions, etc.). As a consequence, the number of branches in the contextual graph (as in the tree representation) would increase without reason in a kind of combinatorial explosion. For avoiding the introduction of an artificial complexity in the representation, we introduce the notion of Executive Structure of Independent Activities (ESIA) for expressing in one item all the variants. (Initially, we used two other names to describe the implementation of such a structure, namely « Parallel action grouping » and « temporal branching ».)

Independent activities accomplish local goals to reach the global goal of the group without coordination with other activities because the way in which an independent activity is realized does not matter for the global task. Moreover, because activities in an ESIA are independent, contextual elements in these activities are independent of those in the rest of the contextual graph (only the results of the independent activities matter). Thus, there are the working context of the contextual graph and the working context that is local to the ESIA. Thus, with respect to the contextual graph, an ESIA is a building block like an action or an activity at the granularity level of the contextual graph. There is a unique result as output of an ESIA, the same for any ordering of the branches (an example with the voting system is given in the next section). An ESIA addresses a local goal of a contextual graph.

An ESIA generally represents (as a simplification) a complex entanglement of contextual elements corresponding at a lower level than the level of the incident-solving description modeled in the contextual graph. In the Contextual-Graphs formalism, an ESIA is represented as a set of red parallel vertical branches, each containing an independent activity. Externally, it is a building block of a contextual graph like an action, a contextual element and an activity. Figure 14 represents the example of the evacuation of a damaged train by a help train where an action is “Make empty your train of the travelers” (see Figure 6) that is applied to the damaged train as well as the help train.

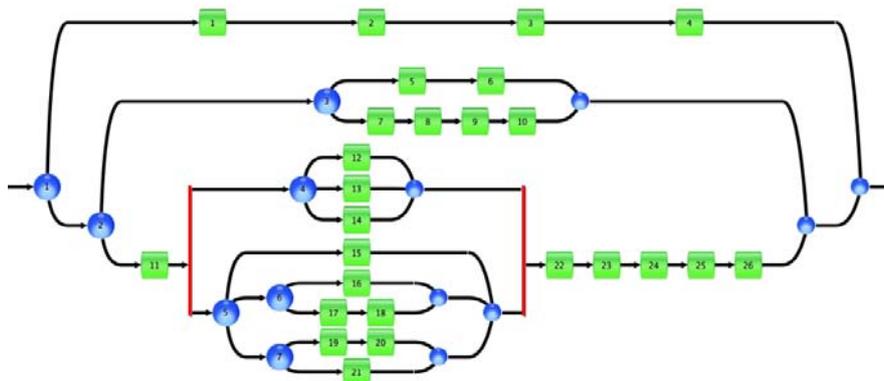


Figure 14. Contextual Graph for solving the “damaged-train evacuation” incident

The upper branch of the ESIA tells how to empty the help train, and the lower branch shows how to empty the damaged train of travelers. The actions of both activities are locally and independently carried out. To detect such ESIA in real-world applications, one has to detect sequences of actions that may actually be indifferently carried out in an order or another. For example, sometimes the operators decide to do a sequence "A-B" and at other moments the sequence "B-A," one can suppose that actions "A" and "B" are independent, but must be accomplished before a given step (the sequence of five actions on Figure 14). Such situations are easy to detect automatically from records [10].

Often, before to make a decision an expert tries to have a global view on the context of the focus based on several independent criteria (often subjective criteria). The expert gives more or less importance on each criterion (and possible combination of criteria) based on qualitative arguments instead of objective ones. This situation is modeled by a voting system where a key contextual element “vote” will be incremented of +1 (but also possibly +2 or -1 depending on the experts) if the criterion is present. Figure 15 presents a simple case of a voting system with two criteria in an ESIA item for representing the global evaluation of an expert. Before the ESIA, an action initialize the reserved contextual element “vote” to zero. A criterion is checked on each branch of the ESIA in an activity that retrieves the value of the criterion (either a visual observation or the result of a blood analysis). For example, in a project with ana-cyto-pathologists, a criterion concerns the “nuclear pseudostratification”. The activity aims to obtain this contextual information (i.e. presence or absence of pseudostratification). The contextual element following the activity is used to modify the reserved contextual element “vote” that may be incremented ($\text{vote} = \text{vote} + 1$) if the criterion is verified in Action 3. Once all the branches of the ESIA have been traversed, the key contextual element “vote” may have one of the values 0, 1, 2 as a global subjective estimation on the object in the focus at the contextual element 3 where $\text{vote} = 1$ is obtained either for criterion 1 or criterion 2 alone.

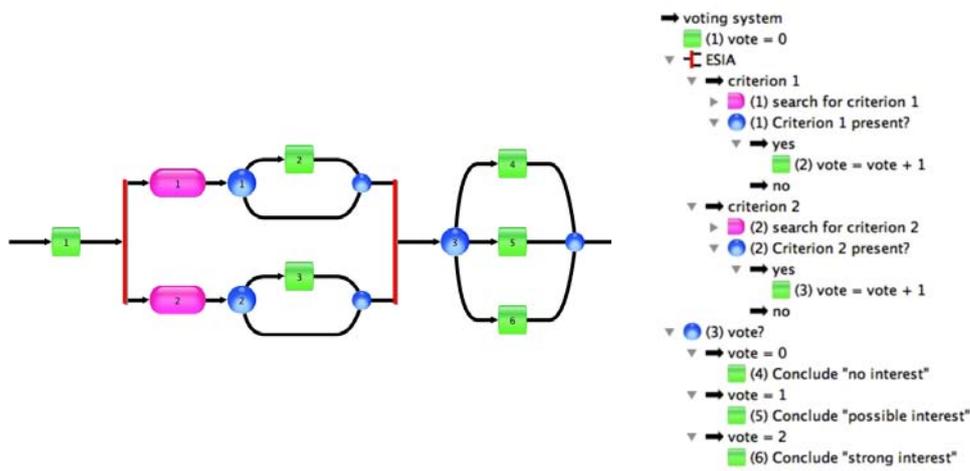


Figure 15. Example of use of the voting system

At the output of this contextual graph (the contextual element 3 “vote”?), the expert has three options for making his decision (no interest for this focus, a possible interest or a string evidence for the focus). This represents in a realistic explanation why different experts may reach the same conclusion by different appreciation of the criteria. The weight given for a vote may be different from a criterion to another criterion. For example, an expert may decide to give “+ 2” if he judges the corresponding feature more important than the other or finds relevant the combination of two features. Conversely, the lack of a feature may be meaningful for the expert who may give a weight “-1” to the vote. This voting system offers flexibility to address the variety of expert approaches.

5.2. Related works

In the PEPYS project, the idea of Lamming and Flynn [40] is to use the physical context to organize our memories for past events into episodes, and the location of the episode, who was there, what was going on, and what happened before or after, are all strong cues for recall. Then, the user's context acts as a retrieval key to provide a valuable key for indexing information automatically. Based on this idea, Newman et al. [47] describe an application called PEPYS, designed to yield descriptions of episodes based on automatically collected location data. The program—concretized by an active badge that each employee has—pays particular attention to meetings and other episodes involving two or more people. The episodes are presented to the user as diary generated at the end of each day and distributed by electronic mail. After reading their diaries, users remembered events that they had previously forgotten.

The CYC system [37] uses a context mechanism to simplify the construction of the knowledge base. The basic change is that formulas in the knowledge base are not just true or false; they are true or false in a context. This permits to make statements "about" contexts, although they cannot be completely described. A context is defined as a consistent set of propositional assumptions about which something can be said. Such a set forms a theory of some topic, e.g., a theory of mechanics, a theory of the weather in winter, etc. In that sense, a context is called a "microtheory." The scope of a context (the theory associated with the context) is the set of objects over which its predictions hold. There are as many contexts as sets of assumptions under consideration. Based on a statement made about an object in one context, something may be derived about that object in another context. The two contexts use different vocabularies and make different attributions of an object, but these attributions are about the same object. So, there might be some contexts in which P might not be stateable (in the vocabulary of that context) and there might be yet other contexts in which P is stated differently. The meaningfulness of a formula may depend on the context it occurs in. Different contexts permit to use different languages.

Compton and Jansen [27] address the long-term maintenance of expert systems. They attempt to capture the context by entering the expert's new rule directly as provided, including an 'IF LAST_FIRED (rule n°)' condition. That is, the new rule will not fire on a case unless the old rule, which produced the wrong interpretation, has fired first. Thus the new rules are tested precisely in the context in which the expert provided them, that is the portion of the expert system that comes before this rule is exactly the same as the expert system, which produced the interpretation coming before this rule, is exactly the same. Context is largely determined by the case that prompted the change to the knowledge base [28, 49]. Knowledge has not to be generalized when it is acquired. It is fundamental to record the context in which the knowledge is acquired [37, 59].

Turner [60] has developed a system--an adaptive reasoner--to make context explicit for autonomous underwater vehicles to tackle unanticipated events in complex environments. Context intervenes in at least five different ways: (1) make predictions about the situation; (2) modulate agent's behavior; (3) focus agent's attention; (4) influence an agent's choice of actions; (5) determine how an agent should handle unanticipated events. C-schemas contain information not only describing the context they represent, but also information prescribing how to behave in situations that are instances of that context. An agent's context manager retrieves the best c-schemas from its memory based on features of its current situation, then merges them to form a view of the current context, the current c-schema. Thus, relatively few contexts are represented as c-schemas, but they are combining as needed to adequately represent a particular situation. The major difference with case-based reasoning is how c-schemas are used: generalized cases are usually used as indexing structures, while c-schemas are problem-solving structures in addition to their role in memory organization. Context is mainly considered as a way to cluster knowledge for search efficiency, for representing counter-factual or hypothetical situations, for circumscribing the effects of particular actions to particular situations, and for directing an agent's focus of attention to salient features of a situation.

Context-based Reasoning [35] is based on three ideas. First, any recognized situation inherently defines a set of actions, procedures and expectations that properly address the situation. A situation is defined by the mission, a set of orders and a specific set of environmental conditions at that moment. These sets of actions, procedures and expectations are called *contexts*. Second, as the situation evolves, a new set of actions, procedures and expectations may be required to successfully manage the newly emerging situation. Therefore, a *transition* to a new context must be effected successfully. Three, identification of a future situation can be simplified if things that are likely to happen while under the current situation are limited by the current situation itself. This facilitates *situational awareness*. Technically, CxBR contexts are organized as a tree, albeit possibly with multiple parents. The hierarchy consists of three levels, the Mission Context (strategy level), the Major Contexts (tactical level) and the Minor Contexts (operational level). The Mission context is unique and serves to define rather than control the agent. Major Contexts are the main control elements, and while there may be several assigned to a particular mission, one and only one is actively controlling the agent at any one time. The Minor Contexts play a supporting role and there can be several levels of minor contexts, starting with the Sub-Contexts. CxBR and CxG do not work at the same level of granularity [36]. CxGs could also be used to represent minor contexts when their actions are complex.

A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information. It can be depicted as a sequence of operations, declared as work of a person or group, an organization of staff, or one or more simple or complex mechanisms. However, even as a high-level description of a problem, the use of a workflow in routine implies a huge number of variants to be adapted at various contexts of a given situation. In some sense, workflows are contextualized (like a practice) and thus not reusable directly for facing a change in the business environment because context is not made explicit in workflow description [20]. SWFs are a formalization of the ad hoc process that a scientist may go through to get from raw data to publishable results. Generally, scientists upload their SWFs in a repository for reuse by the community. However, because a SWF is successful

in a specific context, reuse of a SWF suppose a phase of contextualization to retrieve the SWF that may be a good candidate, decontextualization to extract the SWF model applicable in the new domain and recontextualization to build a successful SWF in the new context. This laborious and time-consuming phase of SWF adaptation is due to the fact that a SWF is similar to a practice, but there is no structure for SWFs equivalent to a contextual graph for practices. A coupling of SWF contextual graphs approaches is proposed in [18] for developing contextualized workflows [21].

When compared to BPEL (Business Process Execution Language), the items ESIA and contextual element in Contextual Graphs correspond to the structural activities <flow> and <if> in BPEL, but it is not a total match [32]. Firstly, <flow> activity in BPEL allows dependences between paralleled activities while ESIA in Contextual Graphs have branches containing independent activities. Secondly, the difference between contextual element and <if> depends on the way to make exclusive divergence. In BPEL, <if> only supports the Boolean expression <condition> to make branching. For example, the <condition> would be “If the value of A is bigger than 10”. However in CxGs, contextual element which is a question followed by several answers is any piece of information, knowledge and reasoning of the scientist, associated with a consequential action or activity to be taken. Unifying and assembling workflow models by using contextual graphs will: (1) promote a new way to drive BPEL; (2) hide the complexity of BPEL syntax from the user; and (3) introduce the highest layer of workflow abstraction to automate the workflow generation and validation. Thus we believe that it is possible that CxGs and BPEL work together to utilize the merits of both sides.

Despite of common elements in the triad of representation, BPEL limitation is obvious when we compared with CxGs.

–BPEL mixes up the representation and implementation levels, which confuse the user. For example, the notion of activity in BPEL is at the programming level, not the representation level.

–BPEL obscures the difference between “logic of functioning” and “logic of use” [16] for the users. For example, BPEL provides a large number of activities (9 basic activities and 9 structured activities) that are introduced for implementation purposes (logic of functioning, not in the logic of use for the users).

–BPEL syntax and semantics are not understandable by domain scientists. Conversely, CxGs focus solely on the representation level and provide a uniform way for representing elements of knowledge, reasoning and contexts.

As a lessons learned, BPEL uses a technical approach for managing challenges in workflow representation while CxGs uses a social approach to put scientists in the centre of workflow orchestration. Thus, the combination of CxGs and BPEL presents an opportunity to express scientists’ knowledge, reasoning and context in "rich" representation (i.e., CxGs), while automating the composition and execution of complex workflows.

6. Conclusion

In this paper, we present how the Contextual Graphs representation was elaborated. After several real-world applications (like Oenology), the SART application for the Parisian subway was the first application where it was mandatory to make the notion of context explicit in order to model practices and not procedures only. Modeling practices imposes to use a formalism allowing a uniform representation of elements of knowledge, reasoning and context. The Contextual Graphs formalism comes from the establishment of a robust conceptual framework where elements of the formalism had a clear semantics of high level. The direct consequence of the conceptual framework is its implementation in an operational framework called Contextual Graphs platform. During more than fifteen years, the CxG platform allowed the modeling of problems in twelve real-world applications

and more than one hundred of examples (realized by Master students during their course on Context Management at UPMC).

A contextual graph is the implementation of a task-realization model based on the notion of activity. The goal of such a representation is to “capitalize the accumulated experience” in order to reuse it for supporting other actors realizing the same task. If procedures are guidelines for task realization, contextual graphs are guidelines for activity management, i.e. task realization in contexts. We are working on this aspect in modeling group activity. The contextual graph and its working context offer a tool for activity management and contains the different possible outcomes for task realization (e.g. completed, abandoned, stand-by, etc.) depending on context, that is, the corresponding proceduralized contexts.

A system using such an “experience base” will be able to explain, anticipate by simulation, and complete its knowledge on the task realization and alert about the variants abandoned for the practice. An actor develops a new practice when a contextual element was not considered up to now because it was either not relevant or always instantiated at the same value. Thus, this contextual element was not included in the contextual graph. Without knowledge about this contextual element, the system fails to reproduce the actor’s reasoning developing the new practice, but the system may recover its failure by adding this contextual element for distinguishing between the known practice and the new practice. The capabilities of incremental knowledge acquisition and practice learning allow a system exploiting contextual graphs to become increasingly “intelligent” because it can benefit of the situation to learn incrementally new practices when it fails. In that sense, a contextual graph is a local base of experiences on the task realization and a system working on such experience bases will be context based intelligent assistant systems.

Contextual Graphs are particularly adapted to complex tasks where there are a large number of contextual elements about actors’ specificity, the task to realize, the situation in which an actor realizes the task, and the local environment where resources are available. The notion of contextual elements allows the handling of information coming from different heterogeneous sources like the user, the task, the situation and the local environment.

However, the story of Contextual Graphs formalism does not end at this point. Indeed, there are several promising paths to explore in different domains like medicine, security, and crisis unit. The modeling of group activity is a first path to explore, especially in emergency rooms and crisis cell. Another interesting path to explore is the capture and exploitation of weak signals through the notion of generic frame and focus dressing [17]. The generic frame is the set of all the contextual elements for addressing a task realization. Indeed, only a part of the contextual elements of the generic frame are relevant for addressing the focus in a given context of the task realization. A preliminary study based on the TACTIC project is promising. More common domain like training and knowledge management also are directly concerned by context, and the CxG platform would be a powerful tool to support modeling of the human reasoning in these domains too.

7. Bibliography

- [1] Attieh E., Brézillon P. and Capron F. (2013) Context-based modeling of an anatomico-cyto-pathology Department Workflow for Quality Control. In: P. Brézillon, P. Blackburn, and R. Dapoigny (Eds.): Modeling and Using Context (CONTEXT 2013), LNAI 8175, pp. 235-247.
- [2] Bazire M. and Brézillon P. (2005) Understanding context before to use it. Modeling and Using Context (CONTEXT-05), A. Dey, B.Kokinov, D.Leake, R.Turner (Eds.), Springer Verlag, LNCS 3554, pp. 29-40.
- [3] Bielikova, M. & Navrat, P. (1997) A multilevel knowledge representation of strategies for combining modules. *Proceedings of the Seventh International Conference on AI and Information-Control Systems of Robots*. World Scientific, Singapore: 155-168.
- [4] Bodlaender H.L., and De Fluiter B. (1996) Parallel algorithms for series parallel graphs. Springer Berlin Heidelberg.

- [5] Brandstädt A., and Spinrad J.P. (1999) Graph classes: a survey. SIAM, Vol. 3.
- [6] Brézillon P., Gentile C., Saker I. and Secron M. (1997) SART: A system for supporting operators with contextual knowledge. First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97), Rio de Janeiro, Brasil, Federal University of Rio de Janeiro (Ed.), pp. 209-222.
- [7] Brézillon P. and Pomerol J.-Ch. (1998) Using contextual information in decision making. In: Chapman and Hall. Widmeyer G., Berkeley D., Brézillon P. and Rajkovic V. Eds.: Context-Sensitive Decision Support Systems, pp. 158-173.
- [8] Brézillon, P. and Pomerol, J.-Ch. (1999a) Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, **62**(3), Paris: PUF, pp 223-246.
- [9] Brézillon P., Pasquier L. and Pomerol J.-Ch. (2000) Representing operational knowledge by contextual graphs. *Advances in Artificial Intelligence*, M.C. Monard and J.S. Sichman (Eds.). Berlin : Springer. Lecture Notes in Artificial Intelligence, N° 1952, pp. 245-258.
- [10] Brézillon P. (2002a) Modeling and using context: Past, present and future. Rapport de Recherche du LIP6 2002/010, Université Paris 6, France.
- [11] Brézillon, P., Pasquier, L. and Pomerol, J. Ch. (2002b). Reasoning with contextual graphs. *European Journal of Operational Research*, 136(2): 290-298.
- [12] Brézillon P. (2003c) Representation of procedures and practices in contextual graphs. *The Knowledge Engineering Review*, 18(2): 147-174.
- [13] Brézillon P. (2003d) Context dynamic and explanation in contextual graphs. In: Modeling and Using Context (CONTEXT-03), P. Blackburn, C. Ghidini, R.M. Turner and F. Giunchiglia (Eds.). LNAI 2680, Springer Verlag Verlag (<http://link.springer.de/link/service/series/0558/tocs/t2680.htm>). pp. 94-106.
- [14] Brézillon P. (2004a) Chapter 6: Learning and explanation in a context-based representation: Application to incident solving on subway lines. In: R. Jain, A. Abraham, C. Faucher and J. van der Zwaag (Eds.) *Innovations in Knowledge Engineering*. International Series on Advanced Intelligence, pp. 129-149, with Pomerol, J.-Ch. and Pasquier, L. (Book Chapter)
- [15] Brézillon P. (2005a) Task-realization models in Contextual Graphs. Modeling and Using Context (CONTEXT-05), A. Dey, B.Kokinov, D.Leake, R.Turner (Eds.), Springer Verlag, LNCS 3554, pp. 55-68.
- [16] Brézillon P. (2007) Context Modeling: Task model and model of practices. In: Kokinov et al. (Eds.): Modeling and Using Context (CONTEXT-07), LNAI 4635, Springer Verlag, pp. 122-135.
- [17] Brézillon J. and Brézillon P. (2008) Chapter 42: Context sensitive decision support systems in road safety. In: F. Burstein and C.W. Holsapple (Eds.), *Handbook on Decision Support Systems*, pp. 675-691. International Handbook on Information Systems, Springer-Verlag, Heidelberg. Series Editors: P. Bernus, J. Blazewicz, G. Schmidt, M. Shaw.
- [17a] Brézillon P. and Pomerol Ch.-J. (2010) Framing decision making at two levels. In : Respicio, A., F. Adam, G. Phillips-Wren, C. Teixeira & J. Telhada (Eds.) *Bridging the Socio-Technical Gap in Decision Support Systems. Challenges for the next Decade*. IOS Press, pp. 358-368.
- [18] Brézillon, P. (2011) Contextualization of scientific workflows. M. Beigl et al. (Eds.): *Modeling and Using Context (CONTEXT-11)*, Springer-Verlag, LNAI 6967. Springer, Heidelberg, pp. 40-53.
- [19] Brézillon P. (2012) Modeling activity management instead of task realization. In: "Fusing DSS into the Fabric of the Context", In: A. Respicio and F. Burstein (Eds.) IOS Press, Amsterdam, NL, pp. 51-62.
- [20] Brézillon P. and Pomerol J.-Ch. (2012) Context-oriented workflows for flexible use. 16th IFIP WG 8.3 International Conference on Decision Support Systems (DSS-2012). Annavissos, Greece, June 28-30, 2012.
- [21] Brézillon P. (2013) Operational workflows in Contextual Graphs : A proposal for BPM. CONTEXT-13 Workshop on Context for Business Process Management (CBPM'13), Annecy, France, October 28, 2013.
- [22] Brézillon P. (2014b) Chapter 4: A context-centered architecture for intelligent assistant systems. In: Faucher C, Jain LC (eds) *Innovations in intelligent machines - 4*. Springer International Publishing Switzerland, Volume 514, 2014, pp 103-127. DOI: 10.1007/978-3-319-01866-9_4.
- [23] Brézillon P., Attieh E. and Capron F. (2014) Modeling glocal search in a decision-making process. In : Gloria PhillipsWren, Sven Carlsson, Ana Respicio, Patrick Brézillon (Eds.): *DSS 2.0 – Supporting decision making process with new technologies*. *Frontiers in Artificial Intelligence and Applications*, 261, IOS Press, pp. 80-91. DOI : 10.3233/978161499399580

- [24] Brézillon, P. (2015) Modeling expert knowledge and reasoning in contextual graphs. In: Christiansen H., Stojanovic, I. & Papadopoulos G.A. (Eds.): Modeling and Using Context (CONTEXT-15), Springer, LNAI 9405, pp. 18-31.
- [25] Britanik, J.M. and Marefat, M.M. (1999) Hierarchically merging plans in decomposable domains. *IEEE Trans. on Systems, Man, and Cybernetics*, 29(1): 27-39.
- [26] Clancey, W.J. (1992) Model construction operators. *Artificial Intelligence*, 53, p. 1-115.
- [27] Compton, P. and Jansen, B. (1988). Knowledge in context: A strategy for expert system maintenance. *Lecture Notes in Artificial Intelligence*, J.Siekman (ed), Subseries in Computer Sciences, Vol. 406.
- [28] Compton, P., Yang, W., Lee, M. and Jansen, B. (1991). Cornerstone cases in a dictionary approach to rule maintenance. *Proceedings of the IJCAI'91 Workshop on Software Engineering for Knowledge-Based Systems*, August, pp. 24-40.
- [29] De Brito, G. and Boy, G. (1999). Situation awareness and procedure following. *CSAPC'99*, Villeneuve d'Ascq, Presses Universitaires de Valenciennes, pp 9-14.
- [30] Denis, M. & Sabah, G. (1993) In *Modèles et concepts pour la science cognitive. Hommage à J.-J. Le Ny*. (Grenoble: PUG).
- [31] Eppstein, D. (1992) Parallel recognition of series-parallel graphs. *Information and Computation* 98.1, pp. 41-55.
- [32] Fan X., Brézillon P., Zhang R., Li L. and Pomerol J.-Ch. (2012) A method for making contexts explicit in collaborative workflows. 16th IFIP WG 8.3 International Conference on Decision Support Systems (DSS-2012). Annavissos, Greece, June 28-30, 2012.
- [33] Garcia, K. and Brézillon, P. (2015) A contextual model of turns for group work. In: Christiansen H., Stojanovic, I. & Papadopoulos G.A. (Eds.): Modeling and Using Context (CONTEXT-15), Springer, LNAI 9405, pp. 243-256.
- [34] Giunchiglia, F. (1995). Contextual reasoning. *Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context*, Research report 93/13, LAFORIA, pp. 39-48.
- [35] Gonzalez, A.J. and Ahlers, R. (1998). Context-based Representation of Intelligent Behavior in Training Simulations. *Transactions of the Society of Computer Simulation* 15 (4) pp 153-166.
- [36] Gonzalez, A.J. and Brézillon, P. (2008) Integrating two context-based formalisms for improved representation of human tactical behavior. *The Knowledge Engineering Review*, 23(2): 1-21.
- [37] Guha, R.V. (1991). Contexts: a formalization and some applications. MCC Technical Report ACT-CYC-423-91, December.
- [38] Hayes-Roth, B. and Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science*, 3, 275-310.
- [39] Laird, J.E., Newell, A. and Rosenbloom, P.S. (1987). SOAR: an architecture for general intelligence. *Artificial Intelligence*, 33: 1-64.
- [40] Lamming, M. and Flynn, M. (1994). "Forget-Me-Not", Intimate computing in support human memory. *Proceedings of FRIEND21, International Symposium on Next Generation Human Interface*, Japan.
- [41] Landauer, T., Foltz, P. & Laham, D. (1998) An introduction to Latent Semantic Analysis. *Discourse Processes*. 25 : 259-284.
- [42] Leplat, J. and Hoc, J.-M. (1983) Tâche et activité dans l'analyse psychologique des situations. *Cahiers de Psychologie Cognitive*, 3, pp. 49-63.
- [43] McCarthy, J. (1979). Ascribing mental qualities to machines. In Ringle, M., editor, *Philosophical Perspectives in Artificial Intelligence*. Harvester Press.
- [44] McCarthy, J. (1993). Notes on formalizing context *Proceedings of the 13th IJCAI* Vol.1, pp 555- 560.
- [45] Minsky, M. (1986) *The Society of Mind*, Picador Publisher.
- [46] Moliere (1670). *Le Bourgeois Gentilhomme*, Acte 2, Scene 4.
- [47] Newman, W.M., Eldridge, M.A. and Lamming, M.G. (1994). PEPYS: Generating autobiographies by automatic tracking. *Research Report*, Rank Xerox.
- [48] Pasquier, L. (2000). Raisonnements basés sur le contexte: Contextes procéduralisés, graphes contextuels et schèmes d'action. *Research Report LIP6 N.2000-010*.

- [49] Paton, R., Shave, M., Bench-Capon, T. and Nwana, H. (1993). Domain characterisation in context. Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France.
- [50] Poitrenaud, S. (2001) Complexité cognitive des interactions Homme-Machine. Paris: L'Harmattan.
- [51] Pomerol J.-Ch. and Brézillon P. (1997) Multicriteria decision making: A cooperative process explainable in context. 13th International Conference on Multiple Criteria Decision Making. Cap Town.
- [52] Pomerol J.-Ch. and Brézillon P. (1999a) Dynamics between contextual knowledge and proceduralized context. Modeling and Using Context (CONTEXT-99). In: Lecture Notes in Artificial Intelligence, N° 1688, Springer Verlag, pp. 284-295.
- [53] Pomerol, J.-Ch., Brézillon, P. and Pasquier, L. (2002) Operational knowledge representation for practical decision making, *Journal of Management Information Systems*, 18 (4) pp 101-116.
- [54] Richard, J.F. (1983) Logique du fonctionnement et logique de l'utilisation. Rapport de Recherche INRIA N° 202.
- [55] Sarrazin, P., Biddle, S., Famose, J.-P., Cury, F., Fox, K. and Durand, M. (1996). Goal orientations and conceptions of the nature of sport ability in children: A social cognitive approach. *British Journal of Social Psychology*, 35, 399-414.
- [56] Schank, R.C. (1983) *Dynamic memory, a theory of learning in computers and people*, Cambridge, University Press.
- [57] Torres da Silva, V., Choren R., & Pereira de Lucena, C. J. (2005). Using the UML 2.0 Activity Diagram to Model Agent Plans and Actions. In: F. Dignum, R.M. van Eijk, & R. Flores (Eds.), *4th International Joint Conference on Autonomous agents and multiagent systems (AAMAS'05)*, Heidelberg : Springer LNAI 3859, 594-600.
- [58] Traoré, A., Tattegrain, H. and Mille, A. (2013) A trace analysis based approach for modeling components. In : P. Brézillon, P. Blackburn, and R. Dapoigny (Eds.): CONTEXT 2013, LNAI 8175, pp. 371–380, 2013.
- [59] Turner, R.M. (1993). Context-sensitive reasoning for autonomous agents and cooperative distributed problem solving. Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France, May.
- [60] Turner, R.M. (1997). Determining the context-dependent meaning of fuzzy subsets. Proceedings of CONTEXT-97, Federal University of Rio de Janeiro, pp. 233-242.