Modeling and using context for system development: Lessons learned from experience

P. BRÉZILLON and J.-Ch. POMEROL

LIP6, Case 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France
E-mail: {Patrick.Brezillon, Jean-Charles.Pomerol}@lip6.fr

ABSTRACT: Over ten years a community on context has emerged. There is a now series of conference on context, a web site and a mailing list. The number of web pages with the word "context" has been multiplied by four in the last five years. Being among the instigators of the use of context in artificial intelligence and in decision support system, we present in this paper the evolution of our thoughts over the last ten years and the results obtained about context representation. This experience leads us to introduce the notion of contextual graphs which are used in an application called SART. Moreover, this paper sums up and gives some comments on the papers we published during these last years.

RÉSUMÉ : Depuis une dizaine d'années, une communauté s'intéressant à la notion de contexte a émergé. Il existe maintenant une série de conférences sur le contexte, un site web, et une liste de diffusion. Le nombre de pages web où se trouve le mot contexte a été multiplié par quatre ces cinq dernières années. Etant parmi les initiateurs de cet intérêt pour l'utilisation du contexte en intelligence artificielle et en aide à la décision, nous présentons dans cet article l'évolution de nos réflexions sur la représentation et l'utilisation du contexte dans des systèmes informatiques. Cette expérience nous a conduit à proposerr une représentation basée sur le contexte (graphes contextuels). Cette représentation est maintenant utlisée dans le cadre de l'application industrielle SART. De plus, nous présentons une synthèse de nos travaux ces dix dernières années.

KEYWORDS: DSS and AI, context, context-based representation, intelligent assistant system, contextual graphs.

MOTS-CLÉS: SIAD et IA, représentation du contexte, Systèmes d'assistance intelligents, graphes contextuels.

1. Introduction

Context plays an important role since a long time in domains where reasoning intervenes such as in understanding, interpretation, diagnosis, etc. The reason is that these activities of reasoning rely heavily on a background or experience that is generally not explicit but provides a necessary contextual dimension to knowledge. Everybody uses context in his daily life just as Mr. Jourdain spoke in prose without being aware of it. However, the concept of context misses a clear and unanimous definition.

Context is, on the one hand, a part of world representation. This is one of the current acception of context as a means to describe situations. On the other hand, context is related to knowledge. As regard knowledge, artificial intelligence, in the frameworkof system development, develops for several years representations and more recently some developers rise the question of context representation. It is therefore not surprising that knowledge and context representations present some convergences.

In Decision Support Systems (DSSs), ook-ahead reasoning leads to unmanageable decision trees due to the large number of possible actions and events. In decision practice, people reduce the complexity of the tree by using as much as contextual information as they can. As a consequence, the reasoning moves from look-ahead to diagnosis (In what context are we?) and relies on macro-actions.

Hereafter, we present how context is viewed in Artificial Intelligence (section 2) and in decision making field (section 3). Section 4 gives some insights on context that wwe learnt in system development, and Section 5 presents the representation by contextual graphs, which have been used in for our real-world application.

2. AI representations

2.1 *Various knowledge representations*

In an expert system-like representation, knowledge is gathered as production rules. These rules are pieces of knowledge of the form "if *conditions* then *conclusions*." They are recorded in large rule bases difficult to update. The rules are structured pieces of knowledge, which are supposed easily understood by domain experts. However, the lack of structure of the rule-base impedes the comprehension (even for the experts of the domain) and the maintenance of the knowledge. Some works have been done on rule-bases structuring, namely by splitting of the rule bases into several rule packets, each containing a subset of the rules applied to solve a specific sub-problem (Brézillon, 1990). Clancey (1983, 1995) proposed to add screening clauses to the condition part of the rules so that they are activated only in some kind of context, this amount to add in the preconditions of the rule some clauses constraining the triggering to a certain context. This is burdensome because the designers must anticipate all the possible contexts to define the preconditions of the rules.

The decision tree approach (Raïffa, 1968) tries to represent the decision process step by step. This is obtained by the presence of two types of nodes: the event nodes and the decision nodes. At an event node, paths are separated according to an event on which the decision maker has no influence. On a decision node, the person makes a choice. This approach might be a way to structure rule bases. For each new element analyzed in the conditions, a new event node is created. For each new value of an existing contextual element, a new branch is created, and so on. Rule after rule, a tree is constructed. The leaves give the conclusion rule. The main problem with this structure is the combinatorial explosion. The number of leaves exponentially increase with the deep of the tree. The addition of a contextual element may easily double the size of the tree.

Case-Based Reasoning (CBR) is a kind of analogy reasoning. To solve a current issue, one selects the most similar problem in a problem base and one adapts the solution to the problem at hand. Instead of adapting prior solutions, it is proposed to store and reuse the context of use, i.e., the trace of how those solutions were derived (Leake, 1996). The main advantage of this reasoning is its great power of generalization and its maintenance. However, it fails to provide explanations on the obtained solution.

In Artificial Intelligence, the lack of explicit representation of context is one of the reasons of the failures of many Knowledge-Based Systems (KBSs). For example, Vanwelkenhuysen and Mizoguchi (1995) show that when users do not belong at a same class (i.e. the same working context), the system must tailor its behavior to the different habits of the classes of users. Testers and engineers working on a same device (digital processor boards in a telecommunications production plant) solve the same problem differently (e.g., oscilloscope versus logic state analyzer), each way being effective for routine problems in their workplace but inadequate for the other's.

Studies of KBS use in real-world applications show four main failures (Brézillon and Pomerol, 1996a and 1996b):

(1) Exclusion of the user from the problem solving. KBSs were assimilated to oracles and users as novices. However, unexpected

problems to solve are the norm rather than the exception. KBSs cannot solve such unexpected problems when users, with their practical experience, are not given the opportunity to interfere. What is required is a cooperation between the user and the system and the consideration of the context in which a problem has to be solved.

(2) KBSs do not use correctly their knowledge. Knowledge, which is acquired from human experts, has a *high contextual component* that is generally not acquired with the knowledge because knowledge engineers asked what experts' solution is, not how they reach it.

(3) KBSs cannot initially have all the needed knowledge. Any KBS has limited resources for problem solving and limited influence: One can never anticipate or "design away" all the misunderstandings and problems that might arise during the use of such systems. This implies that knowledge must be acquired incrementally when needed, i.e., in a *given context of use*.

(4) KBSs cannot generate relevant explanations for users because they do not know *the context in which the user ask a question*. The unique way to generate a relevant explanation is that the KBS and the user jointly construct the explanation (Karsenty and Brézillon, 1995).

These computing approaches are well-known paradigms intended to capture human-like reasoning in automatic systems or intelligent assistant systems (Boy, 1991; Brézillon and Cases, 1995; Brézillon et al., 2000). Psychology and ergonomics are also interested in human activity representation.

2.2 *Various viewpoints on context*

There are at least two different viewpoints about context, namely the engineering and the cognitive science viewpoints (Brézillon and Abu-Hakima, 1995). The *cognitive science view* is that context is used to model interactions and situations in a world of infinite breadth, and that the human dimension is the key for extracting a model. The *engineering view* assumes that context is useful for representing the reasoning about a restricted state space within which a problem can be solved. The identification of these two viewpoints permit to understand the contrasted views found in the literature (Brézillon, 1994, 1997).

Most works in knowledge engineering consider sets of discrete contexts, and try to link different contexts together. Creating a context from existing contexts, as proposed by McCarthy, makes it possible to establish a hierarchy of contexts where a formula relating two contexts involving contextual assumptions is itself in a context. The interest of a context hierarchy is that, working on an object in one context, one can derive some knowledge about that object in another context. These two related contexts may use different words, and the treatment of the object may be easier in one context than another. Conversely, in Cognitive Science, researchers, while not excluding the possibility of discrete contexts, consider that the context of interest is the context of the interaction because it is the unique context that may be perceived. The interaction context evolves continuously according to knowledge chunks introduced by the interactiong agents.

According to the engineering viewpoint, the context is considered at the level of the knowledge representation. As a consequence, there is a static view on contexts and the interest is on context management. The static part of the context is what can be coded during design. As regards its dynamical aspect, part of the problem is linked to the changing nature of context in time, by elaboration and shift (Clancey, 1995). Thus, the dynamical aspects of context must be considered during its use, say, a problem solving. One must account for both the *static aspect* (knowledge that remains constant throughout the interaction) and the *dynamic aspect* (knowledge that changes throughout interaction) of context.

Context is considered as a shared knowledge space that is explored and exploited by participants in the interaction. Contextual knowledge acts as a filter that defines, at a given time, what knowledge pieces must be taken into account (explicit knowledge) from those that are not necessary or already shared (implicit knowledge). A context is a structure, a frame of reference, that allows to not express every thing in a story. For example, "At his birthday's party, Paul blew up the candles." It is not said that there was a birthday cake because it is clear for everybody. With a computer system, however there is a compromise to find between the need to store a large number of information pieces and a tailored presentation of the answer to the user's question, i.e. to distinguish between contextual knowledge and the knowledge that stays external to the answer. We will see that often such a distinction can be made only *a posteriori*.

Beyond these two contrasted views, context possesses a time dimension that poses some problems in modeling. The temporal dimension of context arises from the interactions among agents, as opposed to the context as a fixed concept relative to a particular problem or application domain (Maskery and Meads, 1992). In other words, without interacting agents, there would be no context. In communication, the context is considered as the history of all that occurred during a period of time, the overall state of knowledge of the participating agents at a given moment, and the small set of things they are expecting at that particular moment. Mittal and Paris (1993) shows that communication, including explanations, and context interact each other: the context of the situation triggers some actions, and this in turn modifies the context of the situation. Beyond the temporal dimension, context depends on the task at hand too as we will see in the next section.

2.3 *Context and knowledge*

The first distinction between context and knowledge or mere information which is generally acknowledged is that context is *task-oriented*. All the authors we reviewed who use the notion of context, relate the notion to some specific framework of decision and/or action. For example, the fact that we know that the nearest star after the sun is *Proxima Centauri* at 4.22 light-years will never be *contextual knowledge* except for an astronomer or people engaged in star trek! Whether it is backstage contextual knowledge or immediately usable knowledge, depends on what a subject intends to do. So, the context is also *subjective*, even if it can be shared in a community.

These two characteristics (1) task oriented and (2) subjective are also two components of *know how*. This does not mean that the two concepts are similar, since the *contextual knowledge* can contain some theoretical knowledge.

It is often understate (see Tiberghien) that context is devoted to the description of a situation or some circumstances. In other words, the context carries on and is reduced to the "secondary" description of a given nature state. We think that this is a misconception. For example, in many tasks a contextual element, which is taken into account, is the gravity. This is not secondary, moreover the intensity and the role of gravity is obviously a matter of theory. We therefore think that the context is not reduced to *surface and declarative knowledge* and can involve *deep and surface knowledge*.

Context is task oriented, or more precisely the proceduralization process is task-oriented, even *task-focused* and/or *recognition-primed*. The *contextual knowledge* has many variables and that the proceduralization process is mainly an instantiation process. One can say that *knowing how* is instantiated in doing. The *proceduralized context* is an instantiation of a part of the *contextual knowledge*. This instantiation gives the keys for decision making or action. *Proceduralized context* is sufficient for action but only people with the adequate *knowing how* can bridge the gap between *proceduralized context* and action. The *proceduralized context* triggers some entailment links for people *knowing how*. As such it can be regarded either as a part of *know how* or as a signal triggering adapted answers to a situation.

The relationships between the *proceduralized context* and decision making or action are not necessarily explicit. A kind of compilation can occur that establishes some routine links between a *proceduralized context* and the subsequent action.

To sum up, the differences and analogies between context and knowledge, are:

- context and knowledge can be *explicit* or *implicit*, but both can be explicited except for some parts of *know how* context can contain *deep* and/or *surface knowledge*,
- the *contextual knowledge* is loosely task-oriented not reduced to *know how*, because it may contain *deep knowledge*
- the *contextual knowledge* is mainly concerned with this part of knowledge which is useful for describing the nature state preceding decision making or action; as such a given *contextual knowledge* may have several realizations.
- the proceduralization of a *contextual knowledge* piece is a process which may take place in a community of practice and is anyway a mandatory step on the road to action. As such, it has a role in priming action or practice. In some sense, it is the preliminary step for the activation of *knowing how*.
- the *proceduralized context* is task-oriented or/and recognition-primed and subjective like *know how* or *situated knowledge*,
- the link between *proceduralized context* and action is either explicit or implicit (compilation of the *proceduralized context*). As such, the *proceduralized context* is relevant to the so-called externalization process (Nonaka, 1994). This externalization process is a more or less a social process.
- whereas the knowledge is fixed, the *proceduralized context* changes during action.

Pomerol and Brézillon (2001) point out the relationships between context and knowledge. *Know how* is a practical knowledge which is task-oriented. *Know that* or *deep knowledge* is not related to a particular task. The *contextual knowledge* is a subset of the whole knowledge which can contains *deep* and *surface knowledge* but which is loosely related to the task while the *proceduralized context* is exactly what is necessary to perform the task.

2.4 *Some answers concerning context*

We can not speak of context out of its context. The context is something surroundings an item (e.g., the task at hand or the interaction) and giving meaning to this item. It cannot be considered out of its use. Giving meaning to an item, context acts then more on the relationships between items than on items themselves, modifies their extension and surface. Contextual knowledge concerns the future and consequences of actions, and intervenes in decision maker's look ahead reasoning. The context is considered as a shared knowledge space that is explored and exploited by participants in the interaction. Such shared knowledge includes elements from the domain (e.g., instantiated objects and constraints), the users (e.g., their goals), their environment (e.g., organizational knowledge), their interaction with a system (e.g., transaction history). The dynamic dimension of context presents also a difficulty for modeling context.

3. Context in action

3.1 *Introduction*

The SART project (French acronym for support system for traffic control) aims 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 (Brézillon *et al.*, 1997; Pasquier, 2000; http://www.lip6.fr/SART). This project is based on the interaction between the operator and the system and will ease their mutual comprehension. We analyze operational knowledge used by operators and store it in an adapted structure, which is easily understood by operators and efficiently used by the computer. In order to be an Intelligent Assistant System (IAS), SART has to accomplish several functions such as acquiring knowledge from operators; simulating train traffic on the line, possibly with incidents; changing the model of the line on operator's request for helping the operator to test alternative issues; proposing alternatives for an incident solving; training a new operator not familiar with a given line; etc. We use a multi-agent approach in SART development. To date, our group has developed three agents in the last two years, namely a line-configuration agent, a traffic-simulation agent and an incident-management agent.

3.2 *The operational knowledge in the Parisian subway control*

During the modeling phase, we tried to understand the troubleshooting strategies applied in some incidents, such as lack of motor power on a train. Our first representation used rules. This representation was not well accepted by operators and was difficult to maintain. The main reason was that the level of description of the operational knowledge was too low (too many details) and no global strategy was perceptible.

Then, we adopted a tree representation, made of two types of elements: the actions, which are directives to do an action, and the contextual nodes. Contextual nodes lead to an incident solving according to a given context by selecting a path depending on the value of a contextual piece of knowledge. Figure 4 shows the decision tree representation of the official procedure for "train lack of power."

Our tree representation is inspired by decision trees. However, our trees have no decision node, only “chance” nodes where a contextual element is analyzed to select the corresponding path. The operators’ job shows several important specificities that have consequences on the size and structure of our tree:

- Operators have a single main goal: “to reestablish a normal traffic as fast as possible, respecting elementary security rules.” This point put the emphasis on the fact that in the same situation, several strategies permit to solve the incident.

- Operators use many contextual elements to perform their choice. This leads to a large number of practical strategies, even for the same incident. This point multiplies the number of branches and the tree grows rapidly.

- Operators prefer to gather a maximum of information before making their decision. This attitude postpones most of the actions to the end of the branches of the tree. This observation is very close to the observation of Watson and Perrera (1998) that consider a hierarchical case representation that holds more general contextual features at its top and specific building elements at its leaves.

- Operators choose actions that allow acceding to common intermediary situations. They can thus reuse common strategies to clear the incident. As a consequence, the terminal action sequences are often repeated from one branch to another.

- Several action sequences are done in the same order in different situations (paths). Other actions could be done in different order, but must precede a given other. For example, before to link two trains, both have to be emptied, but the order in which they are emptied does not matter (partial order on the actions).

However, the tree structure is heavy and does not permit to represent highly-contextual decision-making in complex applications. Previously, we said that the operators look first for information on the incident context. The reason is that they want to have a clear idea of what the future may be composed of (the look-ahead in Pomerol, 1997), and try to reduce, as far as possible, the uncertainty in the scenario. The problem for operators is that many scenarios diverge progressively (at least on a part) according to context after.

For instance:

Focus of attention:	Damaged-train elimination of the line.
Context:	Activity on the line.
Action:	Lead the damaged train - at the terminal if the line activity is low - on a secondary line if line activity is high.

Landauer and Bellman (1993) find a similar conclusion by noting that the process of a problem study presents three major steps: find_context, pose_problem, and study_problem. The context includes conditions on the existence of certain information and partial mappings.

As most of contextual elements intervene on several paths (e.g., traffic activity, position of the next train), operators prefer to take them into account as soon as possible to have a general picture of the best path to choose. At this step, contextual knowledge will be proceduralized and, as a consequence, operators postpone the action during this proceduralization step. The main objective is to eliminate chance nodes as said before. By grouping together a sequence of actions in a macro-action (a usable procedure), operators hope to plan beforehand the whole look-ahead. Briefly, such macro-actions are a way to make context explicit and introduce modularity in the diagnosis process by managing different modules accomplishing the same function (scenario) but in different ways according to the context.

3.3 *Procedures and practices*

Since 1900, the company faces incidents and its agents solve them. These practices reflect the construction of operational knowledge, step by step, by the operators. Security sake and the willing of incident solving uniformization pushed the head of the company to compare the practices and to establish secure procedures for each encountered incident. In this sense, procedures are collections of safety action sequences permitting to solve a given incident in any case. These procedures are based on practices, but eliminate most of contextual information and particularities of each incident. Trying to promote sufficiently general procedures results often in sub-optimal solutions for incident solving. In this sense, procedures are useful guidelines for operators, but they have to be adapted for each new incident situation.

At the RATP, most of the incidents have been well-known for a long time (object on the track, lack of power supply, suicide, etc.). Thus, the company has established procedures for incident solving on the basis of their experience. However, each operator develops his own practice to solve an incident, and one observes almost as many practices as operators for a given procedure because each operator tailors the procedure in order to take into account the current context, which is particular and specific. In many working processes human beings can be observed to develop genuine procedures to reach the efficiency that decision makers intended when designing the task. Some parts of these practices are not coded (Hatchuel and Weil, 1992). Such know-how is generally built up case by case and is complemented by "makeshift repairs" (or non-written rules) that allow the operational agents to reach the required efficiency. This is a way of getting the result whatever the path followed. The validation of those unwritten rules is linked more to the result than to the procedure to reach it. De Terssac (1992) spoke of a logic of efficiency.

This discussion points out that if it is relatively easy to model procedures, the modeling of the corresponding practices is not an easy task because they are as many practices as contexts of occurrence. For example, when a train cannot move in a tunnel, there are procedures for evacuate travelers at the nearest station, for evacuate the damaged train by another train, etc. Some procedures are sequential, but others may be accomplish in any order compared to some ones. When a train must push a damaged train, both trains must be empty but the order in which travelers of the two trains are evacuated is not important and depends mainly on the context in which are the trains. What is important is that the two actions must be completed before to intervene on the damaged train. For complex incident solving, it is not possible to establish a global procedure, but only a set of sub-procedures for solving parts of the complex incidents. Moreover, procedures cannot catch the high interaction between the solving of the incident itself and the number of related tasks that are generated by the complex incident. As a consequence, there are as many strategies for solving an incident as operators: Cases that are similar in one context may be totally dissimilar in others as already quoted by Tversky (1977).

4. Understanding and modeling context

4.1 *Introduction*

In this section, we present our view on context. First, we describe three aspects of context. Second, we discuss the way in which contextual knowledge can be represented. Third, we give an example of context-based representation of operational knowledge. Fourth, we illustrate the movement from contextual knowledge to proceduralized context.

4.2 *Three parts of context*

We consider three types of context in a given decision making step. First, context that is shared by those involved in the problem and is directly but tacitly used for the problem solving. Second, context that is not explicitly used but influences the problem solving. Third, context that has nothing to do with the current decision making step but is known by many of those involved. We call these three types of context respectively: proceduralized context, contextual knowledge and external knowledge. We start by giving some insights about these three types of context in the framework of SART before stating more precise definitions in the next section.

We define context as the sum of all the knowledge possessed by the operators on the whole task. Thus, context is from the very beginning task-oriented. At a given step of a decision making process, we separate the part of the context that is relevant at this step of the decision making, and the part which is not relevant. The latter is called **external knowledge**. The former part is called **contextual knowledge**, and obviously depends on the agent and on the decision at hand. A part of the contextual knowledge will, as explained below, be proceduralized. We call it the **proceduralized context**. Figure 1 illustrates the three types of context.

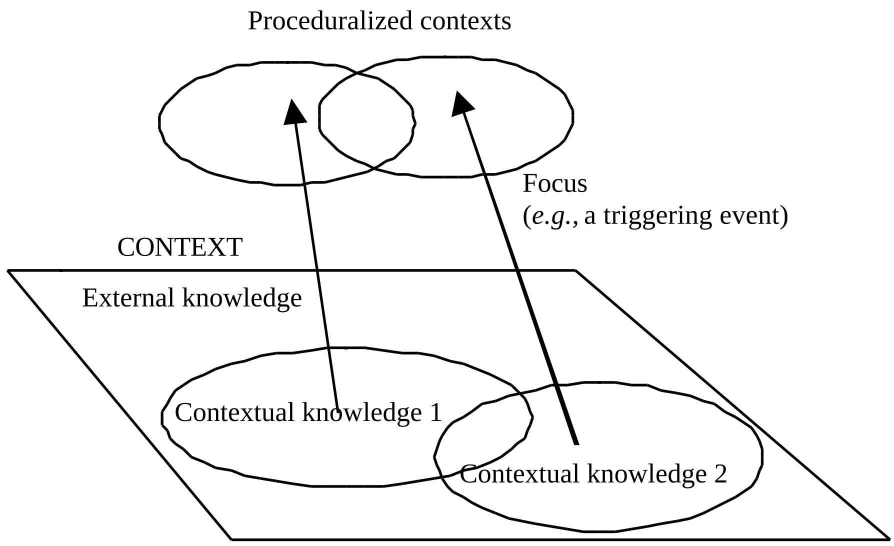

Figure 1. Different types of context

Indeed, it is difficult to define the concept of context without considering the people involved in a situation because, at first glance, context involves knowledge that is not explicit. This explicitness depends on the actors. Some common knowledge is implicit but well-known, for example the fact that it is easier to organize emergency operations in a station than in a tunnel. When the reasoning yields this type of knowledge, it is easily proceduralized and becomes an implicit part of the reasoning that can be elicited by knowledge engineers and finally included in the operation model.

The second fact is that each person involved uses a large amount of knowledge, different from one person to another, to picture the situation. We can define the contextual knowledge as all the knowledge that is relevant and can be mobilized to understand a given situated decision problem. By "situated" we mean in given, dated, well specified circumstances. The word "situated" was introduced into artificial intelligence by Clancey (1995). In its artificial intelligence sense, "situated cognition" emphasizes the role of interaction and context in human behavior. This weak situated cognition hypothesis (Menzies, 1996), which links knowledge, interaction and context, provides a good background for our views.

Contextual knowledge is *personal to an agent* and it has *no clear limit* (MacCarthy, 1993). Contextual knowledge is evoked by situations and events, and loosely tied to a task or a goal. However, when the task becomes more precise, a large part of this contextual knowledge can be proceduralized according to the current focus of the decision making. Although the contextual knowledge exists in theory, it is actually implicit and latent, and is not usable unless a goal (or an intention) emerges. When an event occurs, the attention of the actor is focused and a part of the contextual knowledge will be proceduralized. In our definition, the contextual knowledge is dependent on the situation (date, location, participants); it is a sub-part of the overall context (see Figure 1).

Contextual knowledge intervenes implicitly by constraining problem solving. For example, operators that ensure the monitoring of the distribution of water in Paris had noted that there was a peak in the water consumption late each evening. The peak was reproducible every day but not predictable because not exactly at the same time. After an inquiry, they discovered that persons use water for domestic needs (drink a glass of water, wash dishes, pour water on flowers, go to the toilets, etc.) during the advertisements introduced in the TV movie. The introduction of advertisements in the movie depends on the organization of the movie scenario. Such a knowledge (the link between the peak of consumption and the advertisements at the TV) has a contextual nature for the water distribution. Contextual knowledge constrains a given step of the problem solving (water distribution at advertisements time in the example) without intervening in it explicitly by nature.

Proceduralized context is the knowledge that is explicitly considered at a given step of a problem solving. The proceduralized context is a part of the contextual knowledge that is invoked, structured and situated according to a given focus and which is common to the various people involved in decision making. The proceduralized context may be compiled but can generally be elicited with the usual techniques of knowledge acquisition, it is limited and should be a part of the operation model. For the model to be usable, people have to define a finite number of situations, and diagnosis consists mainly of trying to determine in which situation those involved are.

4.3 *Representation of contextual knowledge*

Contextual knowledge is more or less similar to what people generally have in mind about the term 'context'. It contains some general information about the situation and the environment of the problem. Contextual knowledge implicitly delimits the resolution space (this idea is also evoked in Bainbridge, 1997). It is always evoked by a task or, in our case, an event, but does not focus on a task or on the achievement of a goal but is mobilized according to a set of tasks, even though it has not yet been proceduralized for use.

For instance, in a normal situation, the control operator faces the following concern:

> F0: the *normal* focus of attention is to see that schedules and intervals between trains are respected.

This task can be regarded as routine and does not require special attention. Nevertheless, contextual knowledge about control is involved. In this task, the word *normal* has different meanings according to the context; let us look at some possible contexts.

> C0: the normal context associated with F0 involves:
>
> k1: type of day (e.g., working day, Saturday, Sunday, Holidays),
>
> k2: period of the day (morning, afternoon, evening),
>
> k3: traffic state (rush hours, off-peak hours),
>
> k4: the section load (very busy, few people),

All these pieces of knowledge are some of the elements defining the contextual knowledge describing the environment of the problem with which the following pieces of knowledge are associated:

k5: the interval between trains according to the situation,
k6: the stopping time in stations, etc.

Contextual knowledge is therefore quite large and not focused. Many "normal" contexts are contained in this contextual knowledge. Assume now that an incident occurs on the subway line; the pieces of knowledge k1 to k4 are (or should be) immediately invoked. This results in k5 and k6 being invoked. They become a part of the proceduralized context in which the incident is resolved

Contextual knowledge is on the back-stage, whereas the proceduralized context is on the front-stage in the spotlights. It is noteworthy that, as far as engineering is concerned, only the proceduralized context matters, but contextual knowledge is necessary because this is the raw material from which proceduralized context is made. One can say that contextual knowledge is proceduralized, not necessarily explicitly, to become the proceduralized context. In a sense, the proceduralized context is the contextual knowledge activated and structured to make diagnoses, decisions and actions.

4.4 *Movement between contextual knowledge and proceduralized context*

Decision making is a dynamic process. From one step to the next one, a piece of contextual knowledge either enters the proceduralized context or become external knowledge. Conversely, a piece of proceduralized context may become either contextual knowledge or external knowledge. Thus, the content of the context evolves continuously all through the decision making. Once the first pieces of contextual knowledge are mobilized, some other pieces of contextual knowledge, such as the position of the incident on the line, also enter the focus of attention and are proceduralized. The proceduralized context may also evolve to integrate some knowledge that, up to now, has neither been proceduralized nor is contextual (*i.e.* external knowledge). Examples are maintenance activity on the line, the number of trains on the line, the available help in the control room or the experience of the train driver. Thus, the diagnosis context evolves jointly and continuously with the reasoning process.

From an analysis of action sequences, operators may face three alternatives:

(1) the action sequence leads to a successful incident solving,

(2) the action sequence must be refined, and

(3) the action sequence must be changed.

In the first alternative, the set of proceduralized context pieces leads to the incident solving. In the second alternative, the set of proceduralized context pieces must be extended, new contextual knowledge pieces are introduced in the building of the proceduralized context pieces, the decision making process is reiterated and leads to a modified action sequence. In the third alternative, a deep change of the proceduralized context pieces (i.e. the new set of contextual knowledge pieces that is chosen is different of the previous one) leads to a different decision making process and a different sequence of actions. In the two last alternatives, the process of decision making is reiterated until a successful incident solving is reached. The last alternative--change of the action sequence--corresponds to a new process of decision making by adjustment of scenarios. A change in this processus is due to a lack of contextual information that is needed to choose among alternatives in the decision making. This is because a scenario is a sequence of actions intertwined with events that are external to the decision making but act as constraints on it.

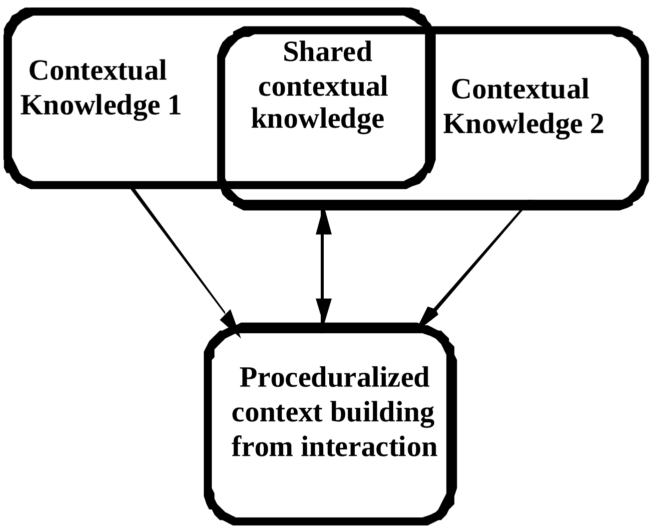

Figure 3. A representation of the interaction context

Figure 3 represents the building process of the proceduralized context from contextual knowledge during an interaction between two agents. The interaction context contains proceduralized context pieces in the focus of attention of the two agents. These pieces of knowledge are extracted from the contextual knowledge of each agent; they are structured jointly by both agents and result in a shared knowledge. Generally, the first utterance of an agent gives a rule such as “Stop at the next station” if the alarm signal is triggered. Then, on the request of the second agent, the first agent may add some pieces of knowledge related to his first utterance. If this knowledge chunk belongs to the common part of the contextual knowledge of the agents, the pieces are integrated into a mutually acceptable knowledge structure, and the knowledge structure may then be moved to the shared contextual knowledge. Thus, the proceduralized context contains all the pieces of contextual knowledge that have been discussed and accepted (at least made compatible) by all the agents. These pieces of proceduralized context then become part of the shared contextual knowledge of each agent, even if they do not remain within the focus of the proceduralized context.

Such a continuous building of the proceduralized context has the consequence that we face a context-based approach: context-based reasoning, context-based planning, etc.

4.5 *Related works on context-based approaches*

Gonzalez and Ahlers (1997) describe the development of a knowledge representation paradigm that could be effectively and efficiently used to model the intelligent behavior of simulated agents in a simulator-based tactical trainer. Their hypothesis is that whereas tactical knowledge is highly dependent on the context (i.e., the situation being faced), a combination of script-like structures and pattern-matching rules in an object-oriented environment could serve as a concise means of representing the knowledge involved, as well as an efficient means of reasoning with that knowledge. This hypothesis was tested through the development of a prototype system that implemented the knowledge of a submarine tactical officer on a patrol mission. The results of the prototype show that the combination of scripts and rules in an object-oriented environment meets the requirements described above.

Tactical knowledge is required in order to endow autonomous intelligent agents with the ability to act, not only intelligently, but also realistically, in light of a trainee's action. In general, tactical knowledge can be said to address rime-stressed tasks which require (1) assessment of the situation at hand, (2) selection of a plan to most properly address the present situation, and (3) execution of that plan. The work described by Gonzalez and Ahlers is based on the idea that by associating the possible situations and corresponding actions to specific contexts, the identification of a situation is simplified because only a subset of all possible sitions are applicable under the active context.

Their concise means of representing the knowledge involved, as well as an efficient means of reasoning with that knowledge, is called context-based reasoning (CxBR). CxBR encapsulates knowledge about appropriate actions and/or procedures, as well as possible new situations, into contexts. Applying context-based reasoning presents a highly effective and efficient methodology for imparting sufficient intelligence to agents so as to acheive their objective in a training simulator. The context would also contain a set of rules together with its own “mini” inference engine consisting of a pattern matcher, a Rete net as an agenda, as well as the capability to assert and retract facts from the local fact base, to call procedures, and to change contexts.

In CxBR, contexts are the most important representational item. There are three levels of contexts that can be represented, and they are ordered hierarchically: (1) the mission context, the major contexts, and the sub-contexts. Active contexts change in response to external events but also as a result of actions taken by the decision maker. A context can be likened to a situation that has bee recognized, and which has a prescribed set of procedures that must be carried out, either sequentially, methodically, or arbitrarily.

Turner (1993, 1997) has developed a system--an adaptive reasoner--to make context explicit for autonomous underwater vehicles to tackle unanticipated events in complex environments. Contextual information helps the agent to focus its attention on appropriate goals to achieve in the current situation. Thus, 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. An agent should be able to recognize its current context as an instance of a class of contexts it knows about. It should be able to reason about its context, bringing to bear knowledge that is explicitly known to be contextual in nature.

Contextual knowledge is represented as a set of contextual schemas (c-schemas), then retrieving the most appropriate of those and using them to help the reasoner behave appropriately for its current context. Thus, 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. Schemas provide a natural way to represent contexts that facilitates knowledge acquisition and potentially provide a tie to established machine learning approaches such as case-based reasoning.

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.

In this framework, Turner describes context-mediated behavior (CMB), an approach to context-sensitive behavior developed over the past few years for intelligent autonomous agents. Context-mediated behavior (CMB) is based on the idea that an agent have explicit knowledge about contexts in which it may find itself, then use that knowledge when it is those contexts. CMB is automatic once a context is recognized. In CMB, contexts are represented explicitly as contextual schemas (c-schemas). An agent recognizes its context by finding the c-schemas that match it, then it merges these to form a coherent representation of the current context. This includes not only a description of the context, but also information about how to behave in it. From that point until the next context change, knowledge for context-sensitive behavior is available with no additional effort. This is used to influence perception, make predictions about the world, handle unanticipated events, determine the context-dependent meaning of concepts, focus attention, and select actions. CMB is being implemented in the Orca program, an intelligent controller for autonomous underwater vehicles.

In case-based reasoning, it is assumed that similar problems have similar solutions. Retrieving relevant cases is a crucial component of case-based reasoning systems. A main problem is that what is considered similar in one situation may not be similar in another one. Jurisica (1994) proposes a context-based similarity as a basis for flexible retrieval. Case similarity is assessed with respect to a given context that defines constraints for matching. Context allows to specify what parts of information representation to compare and what kind of matching criteria to use. Context can thus be used as a basis of relevance measure, i.e. items are considered relevant if they are similar with respect to the current context. This allows, for instance, for excluding similar but irrelevant items.

Jurisica (1994) defines context-based similarity, where context is a set of attributes with associated constraints on the attribute values. The tasks that can be addressed by a context-based similarity are: comparing items, retrieving items, finding a context, and knowledge mining. Similarity is thus considered as a relation with three parameters: a set of relevant items, a context and an information base. Context-based similarity has two levels. The first level is an equivalence of items and is called a surface similarity. The second level deals with similarity between contexts and is called deep similarity. Context specifies how close retrieved items are, i.e. it can be perceived as a measure of usefulness. Thus, context in context-based similarity is useful during the process of judging how relevant the returned answer is to the current goal.

Jurisica and Glascow (1998) proposed an algorithm that is based on a notion of relevance assessment and on a modified nearest-neighbor matching. Its modifications include: (1) Grouping attribute into categories of different priorities so that different preferences and constraints can be used for individual categories during query relaxation; (2) Using an explicit context, a form a bias represented as a set of constraints, during similarity assessment; and (3) Using an efficient query relaxation algorithm based on incremental context modifications. The goal is to retrieve not only exact matches, but partial matches (similar cases) as well. In short, context is a parameter of a relevance relation which maps a case base onto a set of relevant (in terms of context) cases. There are several factors affecting performance of the algorithm (Jurisica and Glascow, 1998): the size of a case base (measured in terms of the number of cases in the case base); the size of a case (measured in terms of an average number of attributes used to describe cases); and context (measured in terms of the number of attributes defined and constraints specified); the query complexity (measured in terms of the size of a query and the complexity of the operations required); and the relaxation/restriction strategy used.

5. Contextual graphs as a context-based representation

5.1 *Introduction*

Initially, we used a rule-based representation to describe incident solving (Pasquier, 2000). This representation was not adapted to operators, and we have then chosen a tree representation as presented in Figure 4. This part of the work has been made with L. Pasquier (Pasquier, 2000; Pasquier *et al.*, 2000; Brézillon *et al.*, 2000).

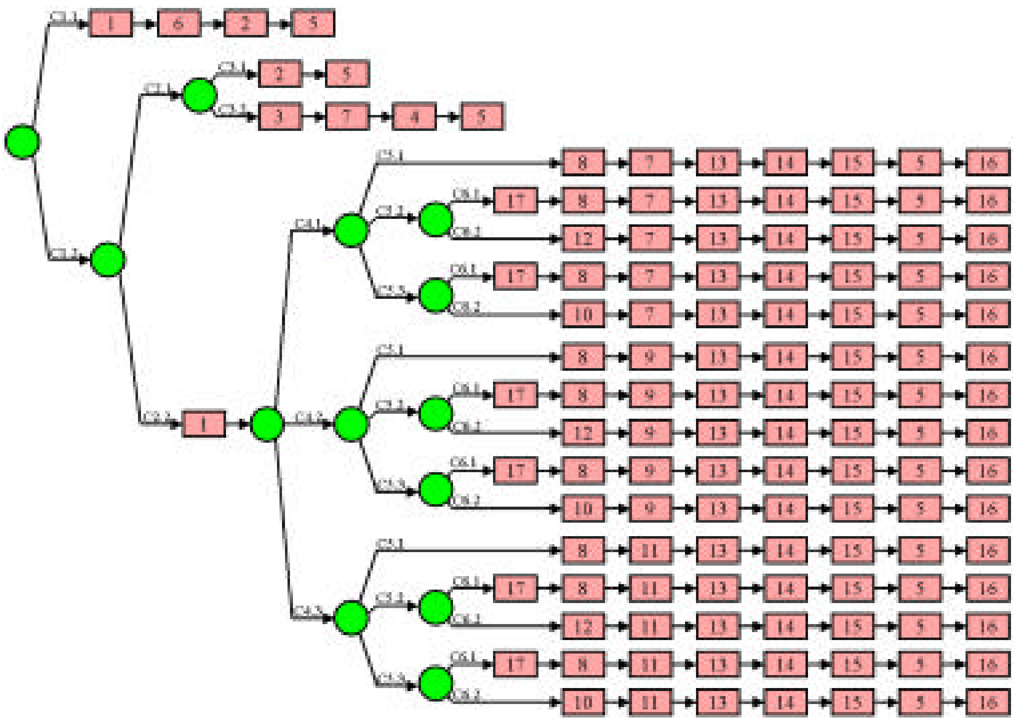

Figure 4. A tree in which many actions are postponed to the end of the branches

Some Actions (A) and Events (E) in case of incident on a metro line are given in the following table.

Actions	
1	Residual traffic regulation
2	Damaged train continues with travelers
3	Damaged train continues with travelers until a steep incline
4	Damaged train restarts without travelers
5	Stable damaged train at end station
6	Repair damage
7	Exit of the travelers out of the damaged train
8	Exit of the travelers out of next train
9	Exit of the travelers out of damaged train via available cars

Events	
C11	Immediate repair possible
C12	Immediate repair impossible
C21	Enough motor coaches available
C22	Not enough motor coaches available
C31	No steep incline between damaged train and end station
C32	Presence of steep incline until end station

C41	Damaged train at station
C42	Damaged train under tunnel

10	Exit of the travelers out of next train via available cars
11	Exit of the travelers out of damaged train via track
12	Exit of the travelers out of next train via track
13	Next train joins damaged train
14	Link both trains
15	Convoy return to end station
16	Disassemble convoy
17	Next train goes to next station

C43	Damaged train partially at station
C51	Next train at station
C52	Next train under tunnel
C53	Next train partially at station
C61	Presence of a station between damaged train and next train
C62	No station between both trains

Our tree representation is inspired by decision trees, but they mainly differ on two points. First, our trees have no decision node, only “chance” nodes where a contextual element is analyzed to select the corresponding path. Second, There are no probabilities. Our main purpose is to describe, with the maximum of parsimony, all the possible contexts in which the decision has to be made. For example, the branch with {C12, C21, C32} describes a context in which there is no immediate repair possible, but enough motor coach power is still available and a steep incline. For this reason we will talk hereafter of context nodes instead of event nodes and of context instead of nature's state. The action postponement observed in Figure 4 amounts to relating each decision to a state of nature, here the context described in the branch. Thus, our representation tends to stick to operators' behavior. In some cases, especially at the beginning of the tree, making decision under uncertainty is probably interesting, but by trying to gather relevant information the operators endeavor to make decision under certainty. This means that when undertaking an action in the tree they consider that, due to the contextual information they got, the state of nature between the root and the action undertaken is the true state of nature.

Moreover, each event at an event node carries on a part of the uncertainty of the situation. For example, C1, in Figure 4, means that either an immediate repair would be possible or not. There are two ways to manage this uncertainty either assess some probabilities for each events, which is the usual view in decision theory or consider that, anyway, the two events are possible depending on the circumstances. Our graph must provide an answer for any possible circumstances whatever the probability is. The problem of an operator is to get an adapted answer as soon as he knows what the circumstances are. Thus, the main problem is to diagnose the exact situations according to the contextual information reaching the operators. For this reason we considered that each branch, actually describes a contextual knowledge, which becomes more and more accurate as long as the branch is followed. We are no longer interested by the probability of a branch but by the possibility to determine, as soon as possible, on which path the operator is, to determine what is the next action to undertake. In other words, we come back to the original idea of Savage (1954), namely that each nature state (a sequence of events is a nature state) describes a state of the world, or using our words, a context for action.

5.2 *Contextual graph representation of the reasoning*

A contextual graph is a directed acyclic graph that represents the actions to undertake according to the context. The action nodes represent actions to undertake to achieve a goal while the event nodes become as explained above, contextual nodes describing the possible contextual issues of a given. The contextual graph is intended to represent the part of the context that we have denoted proceduralized context, i.e. context chunks ready to be used for action.

The proceduralization of the contextual knowledge is a process that makes explicit the links, especially the causal and consequential links, between contextual knowledge chunks and as such the links become a part of the proceduralized context. Thus, the proceduralized context appears as a kind of compiled knowledge that the system will have to decompile to explain its reasoning. Consequently, one can regard the contextual graph (Figure 5) as representing the proceduralized context, because for each context represented by a sequence of contextual nodes the implicit reasoning about causes and consequences implies that the action to undertake is defined without ambiguity. In other words, each sequence of contextual nodes along a branch triggers some actions due to rationales which are not represented on the graph but are generally known and compiled in the operators' mind (proceduralized knowledge). Thus, the contextual graph explicitly represents the reasoning involved in the proceduralized context.

The evolution from a decision tree to a contextual graph is not only a graphical simplification. The purpose of contextual graphs is not to make a decision under uncertainty but to represent along each branch a state of nature that the operators try to identify. The operators endeavor to work with the certainty of having diagnosed the right branch. The chaining of the different contexts along a path figures the evolving proceduralized context. Thus, in a contextual graph, the proceduralized knowledge evolves continuously. In such a dynamical domain (subway line control), it is fundamental to represent accurately the dynamics of operators' reasoning.

Another problem of modeling by trees concerns parallel sequences of actions. In our application, the order in which some actions are executed may be indifferent. For example, when a train must push a damaged train, both trains must be empty but the order in which the trains are emptied is not important and mainly depends on the context in which each train is. As the contextual chunks of knowledge are too numerous to be exhaustively considered, we have introduced a new type of branching in contextual graphs, called temporal branching. A temporal branching figures several branches that may be followed independently and in any order. This idea is inspired by industrial project management. Without such a representation we would be obliged to multiply the number of paths, one for each possible ordering.

The temporal-branching structure is shown in the slashed square in Figure 5. One can see the opening and closing square bracket in dashed lines between action 1 and macro-action MA3. Both branches are composed of a contextual sub-graph. The upper one tells how to empty the damaged train, the lower one shows how to empty the helping train. The actions of both sequences are locally and independently carried out. To detect such temporal branching in real applications, one has to detect sequences of actions that may actually be indifferently carried out in an order or another. For example, if you see that sometimes the operators decide to do a sequence "A-B" and sometimes the sequence "B-A," one can suppose that actions "A" and "B" are independent, but must be accomplished before a given step. Such situations are easy to detect automatically from records.

In (Pasquier, 2000, Pasquier et al., 2000), we show how a series of changes leads from a decision tree as represented in Figure 4 to a contextual graph (Figure 5). These changes are:

1. We identify sequences of actions that appear on several branches and replace each of them by a macro-action (e.g. "assembling trains" is a macro-action);
2. Then, we merge the branches of the tree as soon as the sequence leading to the end of the incident are identical;
3. We introduce the notion of temporal branching for representing action sequences that can be executed in any order; and finally
4. We identify sub-graphs that appear in the contextual-graphs representation of different incident solvings.

The representation by contextual graphs/sub-graphs is very similar (in the spirit) to the generic task approach proposed by Chandrasekaran some years ago (Chandrasekaran *et al.*, 1992).

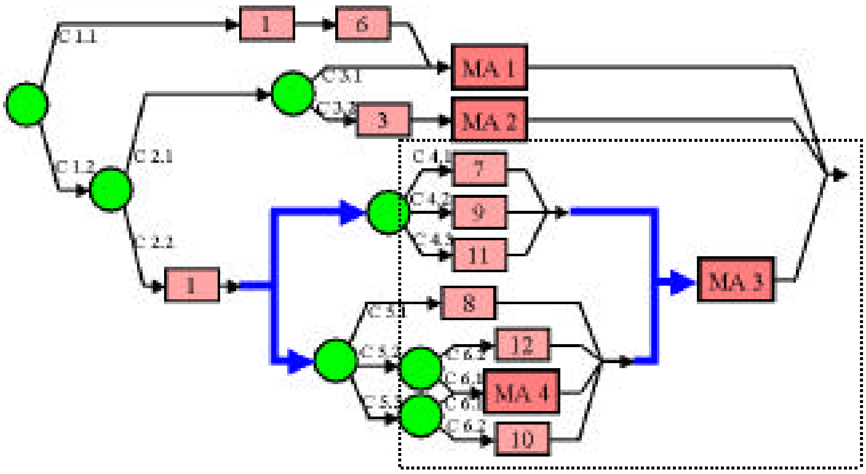

Figure 5. Contextual graph representing the official procedure for "lack of trains power" incident on Figure 4

5.3 *Contextual graph and its sub-graphs*

In Figure 5, the structure in the slashed square has its own signification. It can be thought as an independent plan named "assistance to a damaged train." This plan has a goal (to push a damaged train with another train up to the end-station) and an explanation about the way to carry it out. This explanation results in the proceduralization of the context contained in the contextual sub-graph shown in Figure 6. This sub-graph could be found again in other contextual graphs. Each sub-graph can be considered for its own, by operators as well as by designers. This is a chunk of knowledge that can be considered as an elementary piece of knowledge. We have elsewhere argued that, on the one hand, such an elementary sub-graph can be considered as an atomic task or a scheme of action (Pasquier *et al.*, 2000). On the other hand, our representation is reminiscent to Sowa's conceptual graphs (Sowa, 1984) with their mechanisms of aggregation and expansion. Macro-actions are then considered as a particular case of action structure in which the contextual graphs are reduced to sequences of actions not intertwined with context nodes. However, one must notice that our representation is simpler than Sowa's one because our graph is oriented and the path gives the reasoning followed by the designer according to the different contexts.

The block "train aid" in Figure 6 is found in the solving of different incidents. Such a block corresponds to the right level of operator's interpretation of events and the type of utterance between the operator and drivers of the concerned trains because all of them use the same language (i.e. they are all able to decompile the action “train aid” in elementary actions).

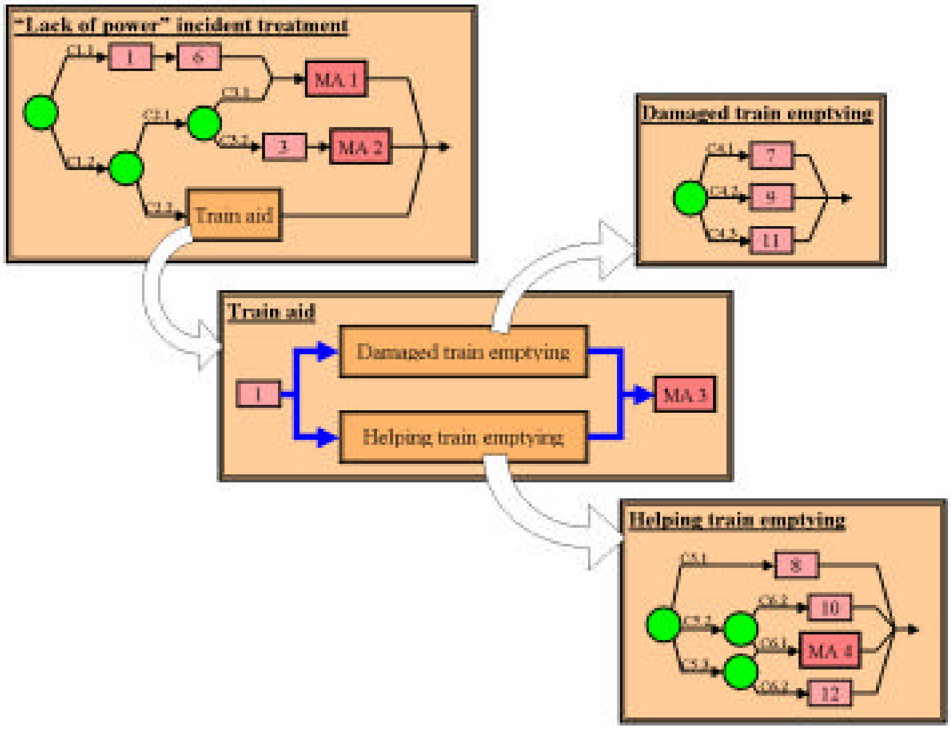

Figure 6. Set of contextual graphs used while "lack of train power" incident resolution

This contextual graph recalls that the structure makes explicit the context and its dynamics for decision-making. This representation is more compact than trees and seems to be well accepted by the operators. As our initial trees were not decision trees, these directed acyclic graphs are not influence diagrams. They simply represent the succession of actions to do for solving an incident; the different possible paths express the possible strategies according to the situation. We must also mention that our representation is much simpler than colored Petri-net (Humphreys and Berkeley, 1992) but it has not the same expressiveness as regards the dynamics because the process of proceduralization de-proceduralization must «linearly» follow the left to right reading of the graph.

5.4 *Learning in contextual graphs*

Consider the following sequence of actions carried out by an operator that must be compared to the contextual graph in Figure 5:

The system begins by identifying the part of the contextual graph which is concerned by the action sequence (see Figures 7). Thus, the system reorganizes the actions 1, 17 and 18. However, the matching stops here because the system knows that the operator is on the verge to make empty the trains (i.e. the damaged train and the aiding train). In the given action sequence, it appears however that the operator has skipped this step. Asking the operator for his rationale, the system learns that the train concerned by the emptying action is already empty because it just leaves the terminal. For integrating the new alternative (branch) in the contextual graph, the system introduces a new contextual node to check the state of the train (starting from a terminal or not) and skip the actions to make the train empty. The operator explaining why, the system learns by doing. There is thus an incremental knowledge acquisition at the level of the contextual graph by adding new contextual nodes .

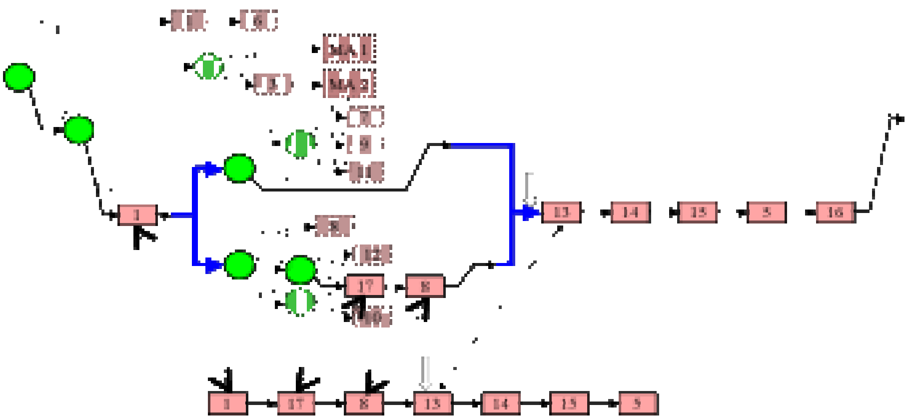

Figure 7. Evolution of a contextual graph by learning

6. Conclusion

The two questions of contextual knowledge and knowledge sharing have received wide consideration in recent years. In this paper we have addressed these two topics through Intelligent Assistant Systems and, more specifically, by using our experience in the development of IASs for process control. As a result of our analyses, we stress the dynamic aspect of the contextual knowledge. We propose to define contextual knowledge as a possibly unlimited, personal and situated set of relevant knowledge involved in an activity. Part of this contextual knowledge is proceduralized to enable cooperation and/or action and this results in a shared proceduralized context that can be elicited, and therefore made explicit by the usual methods of knowledge engineering. There is a finite number of pieces of knowledge in the proceduralized context, each of them being related to a situation, date, locations, participants, problem, etc. One difficult question which can be the main question in diagnosis is to identify the relevant situation. In the second part, we show how we tried to deal with context in a real case by introducing the notion of contextual graph.

Our opinion is that contextual issues cannot be addressed in a static framework only and that eliciting and sharing contextual knowledge is a key process in addressing and understanding and solving complex problems.

7. References

Agabra, J., Alvarez, I., and Brézillon P. (1997). Contextual Knowledge Based System: A study and design in enology. Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97). Federal University of Rio de Janeiro Ed., 351-362.

Bainbridge, L. (1997). The change in concepts needed to account for human behavior in complex dynamic tasks_*IEEE transactions on Systems, Man and Cybernetics, 27*, 351-359.

Boy, G. (1991). Intelligent Assistant Systems. Academic Press, London, Knowledge-Based Systems, 6.

Brézillon, P. (1990). Interpretation and rule packet in expert systems. Application to the SEPT expert system. In S. Ramani, R. Chandrasekar, & K.S.R. Anjaneyulu (Eds.), *Knowledge Based Computer Systems* (pp. 78-87). Heidelberg: Springer-Verlag.

Brézillon, P. (1994). Context needs in cooperative building of explanations. Paper presented at the *First European Conference on Cognitive Science in Industry*. Luxembourg, September.

Brézillon P (1997) A first step towards the modeling of contexts. Proceedings of the 2nd European Conference on Cognitive Science (ECCS'97), pp. 195-198. (Invited talk)

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

Brézillon, P. and Abu-Hakima, S. (1995) Using knowledge in its context: Report on the IJCAI-93 Workshop. The AI Magazine, 16(1): 87-91.

Brézillon, P. and Cases, E. (1995). Cooperating for assisting intelligently operators. Proc. of the International Workshop on the Design of Cooperative Systems, INRIA Ed., pp. 370-384.

Brézillon, P. and Pomerol, J.-Ch. (1996a). Misuse and Nonuse of Knowledge-Bases Systems : the past experiences revisited. In Implementing Systems for Supporting Management Decisions, P. Humphreys, L. Bannon, A. McCosh, P. Migliarese, J.-Ch. Pomerol, Eds., Chapman and Hall, pp.44-60.

Brézillon P. and Pomerol J.-Ch. (1996b), "User Acceptance of Interactive Systems: Lessons from Knowledge-Based and Decision Support Systems", International Journal on Failures and Lessons Learned in Information Technology Management, to appear.

Brézillon P., Pasquier L. and Pomerol J.-Ch. (2000) Reasoning with contextual graphs. European Journal of Operational Research, (to appear).

Brézillon, P., Gentile, C., Saker, I., and Secron, M. (1997). SART: A system for supporting operators with contextual knowledge. Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97). Federal University of Rio de Janeiro Ed., 209-222.

Brézillon P, Cavalcanti M., Naveiro R. and Pomerol J-Ch (2000) SART: An intelligent assistant for subway control. Pesquisa Operacional, Brazilian Operations Research Society, 20(2): 247-268.

Chandrasekaran, B., Johnson, T.R. and Smith, J.W. (1992). Task-structure analysis for knowledge modeling. *Communications of the ACM*, 35(9): 124-137.

Clancey, W.J. (1983). The epistemology of a rule-based expert system: A framework for explanation. Artificial Intelligence Journal 20, 3, 197-204.

Clancey W. J. (1995) The conceptual, non-descriptive nature of knowledge, situations, and activity. Proc. of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Montreal, Canada, August 1995.

Gonzalez A.J. and Ahlers R. (1997) Context-based representation of intelligent behavior in training simulations. Transactions, 15(4): 153-166.

Hatchuel, A., and Weil, B. (1992). *L'expert et le système*. Paris: Economica.

Hoc, J.M., and Amalberti, R. (1995). Diagnosis : Some theoretical questions raised by applied research_*Current Psychology of Cognition*, *14*, 73-101.

Hollnagel, E. (Ed.). (1993). *Human Reliability Analysis, Context and Control*. London: Academic Press.

Humphreys, P. and Berkeley, D. (1992). Support for the synthesis and analysis of organisational systems in deciding on change. *Decision Support Systems: Experiences and Expectations*, T.Jelassi, M.R. Klein and W.M. Mayon-White (Eds.). Elsevier Science publishers, Amsterdam, North-Holland, pp. 29-50.

Jansen, B. (1995). Context in context (Working Draft V4) http://mac145.syd.dit.csiro.au/Context/context.html.

Jurisica I. (1994) Context-based similarity applied to retrieval of relevant cases. Proceedings of AAAI Fall Symposium Series on Relevance. New Orleans, Louisiana, November 4-6.

Jurisica I. & Glasgow J. (1998) An efficient approach to iterative browsing and retrieval for case-based reasoning. Proc. of the 11th IEA-98-AIE, Tasks and Methods in Applied Artificial Intelligence. Springer Verlag, Vol. II, pp; 535-546. A.P. del Pobil, J. Mira & M. Ali (Eds.).

Karsenty, L., and Brézillon, P. (1995). Cooperative problem solving and explanation. International Journal of Expert Systems With Applications, 4, pp. 445-462.

Landauer, C. and Bellman, K. (1993) The role of self-referential logics in a software architecture using wrappings. Proceedings of ISS'93: 3rd Irvine Software Symposium, Irvine, 1-12.

Leake D.B. (1996) Case-based reasoning: Experiences, lessons, and future directions. Chapter I: CBR in context: The present and future. Menlo Park: AAAI Press/MIT Press.

Leplat J. (1985) Erreur humaine, fiabilité humaine dans le travail. In: A. Colin (Ed.), Collection Uiversitaire, pp. 100-120.

McCarthy, J. (1993). Notes on formalizing context. Proceedings of the 13th IJCAI, Vol.1, pp 555-560.

Maskery, H., and Meads, J. (1992). Context: In the eyes of users and in computer systems. *SIGCHI Bulletin, 24*, 12-21.

Mittal, V.O., and Paris, C.L. (1993). *Context : identifying its elements from the communication point of view*. Paper presented at the *International Joint Conference on Artificial Intelligence Workshop on Using Knowledge in its Context*. Paris: France, August.

Nonaka I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1): 14-37.

Pasquier, L. (2000). Raisonnements basés sur le contexte: Contextes procéduralisés, graphes contextuels et schèmes d'action. Research Report LIP6 N.2000-010.

Pasquier, L., Brézillon, B. and Pomerol, J.-Ch. (2000). From representation of operational knowledge to practical decision making in operations. *Decision Support throught Knowledge Management*, S. Carlson, P. Brézillon, P. Humphreys, B. Lundberg, A. McCosh and V. Rajkovic (Eds.), ISBN 3-901882-11-1, pp. 301-320.

Pomerol, J.-Ch. (1997) Artificial Intelligence and Human Decision Making, *European Journal of Operational Research*, 99, 3-25.

Pomerol, J.-Ch. and Brézillon, P. (2001) Is "context" another name for "knowledge"? Modeling and Using Context (CONTEXT-01). Lecture Notes in Computer Science. Springer Verlag (to appear).

Raïffa, (1968). Decision Analysis, Mac Graw Hill.

Savage, L.J. (1954) The foundations of statistics. John Wiley and Sons.

Sowa, J.F. (1984). "Conceptual Structures: Information Processing in Mind and Machine", Addison Wesley Publishing Company.

de Terssac, G. (1992). *Autonomie dans le travail*. Série Sociologie d'Aujourd'hui. Paris: P.U.F.

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. Rapport de Recherche 93/13, LAFORIA, Université Paris VI, Paris, France, 1993.

Turner R.M. (1997) Determining the context-dependent meaning of fuzzy subsets. Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97), Rio de Janeiro, Brazil, February, 1997.

Tversky, A. (1977). Features of similarity, *Psychological Review* 84(4): 327--352.

Vanwelkenhuysen, J., and Mizoguchi, R. (1995). Adaptation of reusable knowledge for workplace integration. Proceedings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning. Technical Repport 95/11, LAFORIA, University Paris 6, France, March, pp. 167-177.

Watson I. and Perera S. (1998) A hierarchical case representation using context guided retrieval. Knowledge Based Systems Journal.