

# **Operational Knowledge Representation for Practical Decision Making**

Jean-Charles POMEROL, Patrick BREZILLON  
and Laurent PASQUIER

*LIP6, Case 169, University Paris 6,  
4 place Jussieu,  
75252 PARIS Cedex 5  
FRANCE*

[Jean-Charles.Pomerol@lip6.fr](mailto:Jean-Charles.Pomerol@lip6.fr)

*Tel 331 44 47 30 65*

# Operational Knowledge Representation for Practical Decision Making<sup>1</sup>

**Abstract** *For the design of an “intelligent” assistant system aimed at supporting operators’ decision in subway control, we modeled operators’ activity and know-how. As a result, we introduce the notion of **contextual graph** which appears as a simple solution to describe and manage operational decision making.*

**Key Words:** *Decision Tree, Context Representation, Contextual Graphs, Operational Knowledge Representation*

## 1. Introduction

In high technical and heavily dynamical regulation processes, operators who are responsible for the process control have to rapidly react. If an incident occurs, they have only few minutes to forge a representation of the issue, gather information on the situation, analyze the incident and undertake the correcting actions. To ease their job, many companies have established fixed procedures. Initially, general procedures have been designed to provide operators with a secure reference for process control. However, these general procedures forget the contextual dimension of the situation at hand. Nowadays, companies are diversifying these procedures by increasingly introducing contextual considerations. This rises the question of representing and managing these very numerous, contextual-dependent procedures.

In this paper we present a new framework inspired from decision trees to represent and use contextual information in operational decision making. This representation named “*contextual graph*” heavily relies on our previous works on the representation of context (*e.g.* [3]).

This paper is organized as follows: in Section 2 we remind some useful insights about context; in Section 3 we introduce contextual graphs and we contrast our views with decision trees and other decision networks; then we finish in section 4 with an application.

## 2. Operational knowledge representation and context

Operational practices are difficult to model: firstly they are numerous, secondly they are often implicit within a community of practice [6] and strongly linked one to each other and, thirdly the main distinction among them is the context in which these practices apply. Moreover these practices are often dynamically constrained in sequences of actions. To gather and study operational practices, we worked on the control of a subway line [4, 5, 25, 26]. In this framework, we recorded the sequences of actions undertaken by the operators as a set of characteristics (pre-condition), including context description and the subsequent result. Using production rules vocabulary we could say that we collected the pre-condition of the actions then the action carried out and its conclusion. With this feedstock, we constructed an adapted representation to record and model this type of knowledge for reuse purpose. Before going forward to the description of the representation by contextual graphs let us just remind some facts about context.

### 2.1. Importance of the contextual dimension and of the dynamics of context

From an engineering point of view we can start from the definition of context as a collection of relevant conditions and surrounding influences that make a situation unique and comprehensible [1, 14]. The difficulty with this definition is that there are “numerous interacting factors that people do

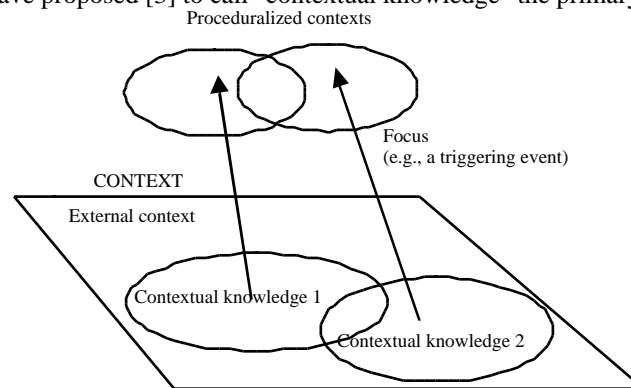
---

<sup>1</sup> RATP and COFECUB in France and CAPES in Brazil provide grants. We also thank J.-M. Sieur at RATP and all the other members working on the SART project.

not even pay attention to on a conscious level, and many of which are outside the ability of machine input devices to capture" [12].

Let us take an example, in the control of a subway line [4, 5], here a large amount of knowledge about trains, electricity, people reaction, etc., contributes to make the situation unique, while some more particular conditions about the time, the day, the weather and so on, specifically influence many decisions. In other words, there is a common background context which is then specified by some conjecture and contingent influences. For example, the general context is subway control which differs from train or bicycle control although they share some mechanical laws and the particular context is specific to a line, a day, an hour, etc. These considerations explain why Tiberghien [37] defines context as the whole set of secondary characteristics of a situation or secondary properties of a cognitive or motivational state of an individual which may modify the effect of an effective stimulation (stimulus) or an oriented activity.

Thus, it would probably wise to talk of primary and secondary context to distinguish between the general, relatively fixed primary characteristics of a situation, and the secondary characteristics which are more mobile. If we think about primary context, we must confess that it is difficult to avoid the word knowledge about this general background used by the operators to carry out their task. This is the reason why we have proposed [3] to call "contextual knowledge" the primary or back-stage context.



**Figure 1.** Three types of context

Therefore, at a given step of a decision process or of a task performing, we distinguish between the part of the context which is relevant at this step of the decision making or task performing, and the part which is not relevant. The latter part is called *external knowledge*. The former part is called *contextual knowledge*, and obviously depends on the agent and on the decision at hand. At a given step of the decision making, a part of the *contextual knowledge* is proceduralized. We call it the *proceduralized context*. The *proceduralized context* is a part of the *contextual knowledge* which is invoked, structured and situated according to a given focus.

The *contextual knowledge* is a backstage knowledge whereas *proceduralized context* is immediately useful for the task at hand. In our representation of context, the *contextual knowledge* is largely tacit, mainly because it is the context that everybody knows without expressing it. In a distinction reminiscent to cognitive ergonomics [21], we could say that the contextual knowledge is useful to identify an *activity* whereas the proceduralized context is relevant to characterize a task.

An important issue is the passage from *contextual knowledge* to *proceduralized context*. This *proceduralization* results from the focus on a task. Thus, the proceduralization process is task-oriented just as *knowing how*; it is often triggered by an event or primed by the recognition of a pattern. Another aspect of *proceduralization* is that the operators transform contextual knowledge into some functional knowledge or causal and consequential reasoning in order to anticipate the result of their own action (see also [11], for a similar observation). Thus, the functionalization is a part of the *proceduralization process*, and this is the reason why we have chosen the term of *proceduralization*. This *functionalization, or proceduralization*, obeys to the necessity of having a consistent explicative framework to anticipate the results of a decision or an action. This consistency is obtained by reasoning about causes and consequences in a given situation. Thus, we can separate the reasoning between diagnosing the real context and, anticipating the follow up [28]. The second step needs a

conscious reasoning about causes and consequences. This explicit reasoning in the mind of the subject has also been recognized by Levesque [22] about beliefs, this is very close to our view.

A second *proceduralization* aspect is a kind of instantiation (see also [13]). This means that the contextual knowledge or background context needs some further specifications to perfectly fits to the task at hand. These precisions and speciation brought to the contextual knowledge is also a part of the *proceduralization* process. We gave elsewhere [26] some insights about the construction of the proceduralized context by the interaction between agents.

## 2.2. Representing context

Artificial intelligence developed for several years, formalisms for operational knowledge representation. Here we present some approaches. Let us start with production rules since, as we said, at a first glance, operational decision making can be interpreted as rule application. These rules are pieces of knowledge of the form “if *preconditions* then *conclusions*.” They are recorded in large rule bases difficult to update. The rules are structured chunks of knowledge which are easily understood by the 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 on the splitting of the rule bases into several rule packets, each containing a subset of rules applied to solve a specific sub-problem [2]. Clancey [9] proposed to add screening clauses to the precondition part of the rules so that they are activated only in some kind of context, this amount to add in the preconditions of the rules 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. Moreover each rules describes a part or a whole context without any reference to the dynamics of the situation which is fundamental to understand the situation.

The decision tree approach [32] tries to represent the decision 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 all the possible realizations of an event on which the decision maker has no influence. At a decision node, the person makes a choice. Thus, we obtain a tree of scenarios each depending on the events. The main problem with this structure is the combinatorial explosion (see [30]). The number of leaves is an exponential function of the depth of the tree. The addition of a contextual element may easily doubles the size of the tree.

Belief networks [27] and influence diagrams (*e.g.* see [23]) have been also used for decision. In these nets, the nodes represent propositions while the arcs make explicit direct dependency between the linked propositions. Thus, the combinatorial explosion is avoided, but the propositions are random variables and the complexity is re-introduced via the conditional probabilities that must be propagated along the links. The belief networks and influence diagrams are particularly used for diagnosis. Indeed, they face the same difficulty as decision trees when the number of events or random variables increases. They are very information-consuming in term of probabilities. Another difficulty is that influence diagrams bring a simplification of the decision tree only when many variables are independent [15, 27]; this depends on the case at hand and is generally not true in troubleshooting.

Another interesting approach is the Case-Based Reasoning (CBR), which is a kind of analogy reasoning [18]. 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. Note that instead of adapting prior solutions, Leake [19] proposes as an interesting alternative to store and reuse traces of how those solutions were derived. The main advantage of this reasoning is its great power of generalization and its maintenance facility. However, it fails to provide explanations on the obtained solution.

## 3. Contextual Graphs

### 3.1. Representing contextual information

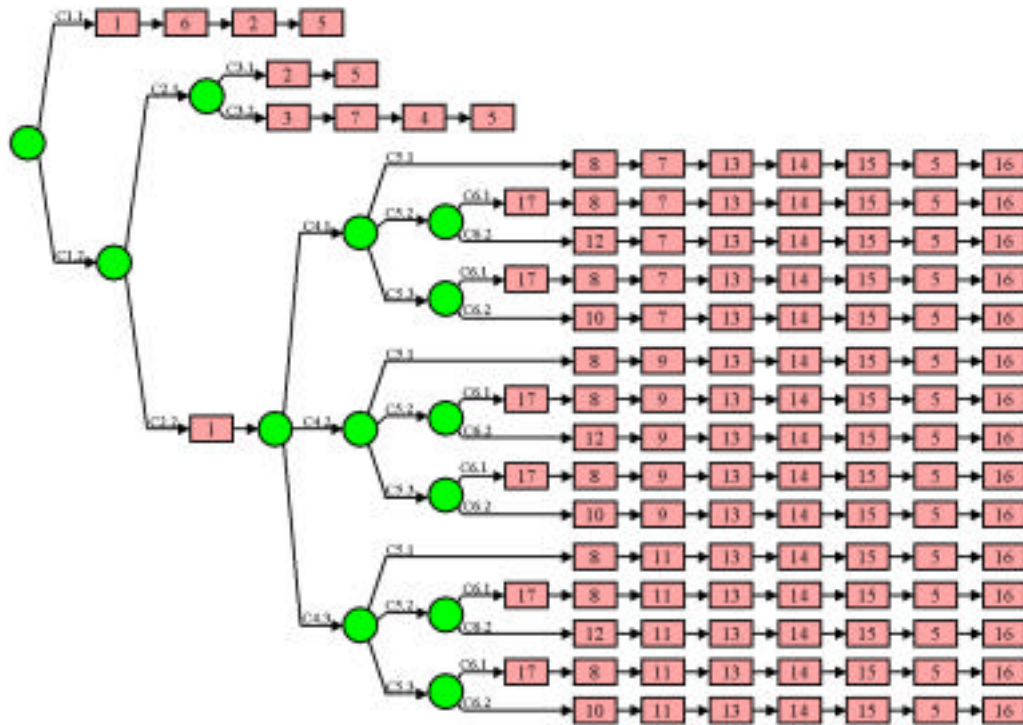
None of the above representations gave us satisfaction and stuck to the operators’ reasoning which is more or less a mixture of rules, decision trees and case-based reasoning. This is the reason why we started to thoroughly study the operators’ practice in the case of incident troubleshooting in the Paris subway. This study proved that the operators are very sensitive to the context of their action. They try to diagnose, as soon as possible, the real state of the system by recognizing and anticipating sequences

of events. As such it is reminiscent of scenario thinking. For this reason we started from the notion of decision tree.

However there is a big difference with scenario thinking. In a decision tree, each event at an event node carries on a part of the uncertainty of the situation. 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 representation 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 situation is. 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. This is not a question of nature bet, all the situations that may happen have to be represented whatever their probabilities are. Thus, we are no longer interested by the probability of a branch but by the possibility to determine, as soon as possible, on what 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 [33], namely that each state of nature (a sequence of events is a state of nature) describes a state of the world, or using our words, a context for action. We must also stress that the operator has no choice of the path and therefore no optimal choice, because the path is dictated by the context of the incident.

At this step there is no probabilities on the events, the main purpose is to describe, with the maximum of parsimony, all the possible contexts in which the decision has to be made. For example, in Figure 2 below the branch with  $\{C_{12}, C_{21}, C_{32}\}$  describes a context in which there is no immediate repair possible but still enough power and a steep incline in view (see Annex 1 for the significance of the nodes). For this reason *we will talk hereafter of context nodes instead of event nodes and of contexts instead of states of nature.* Let us repeat that we are not interested in the probability of say,  $C_{12}$  versus  $C_{11}$ , because our representation intends to be used for any incident, whatever the probabilities are. In this sense, the  $C_i$  are not random variables unlikely in influence diagrams [23]. At a first glance, influence diagrams are acyclic graphs bringing a simplification to decision trees when the random variables are only moderately each other dependent. In operation, the difference is that we are less interested in the possible causes of the problems than in its clearing. When an incident occurs, the probability of occurrence does not longer matter, what matters is the chaining of troubleshooting actions.

The action postponement to the leaves observed in Figures 2 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' behaviour. In some cases, especially at the beginning of the tree, decision making under uncertainty would probably, if possible, be interesting, but by trying to gather as much as possible relevant information before action, the operators endeavour to make their 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.



**Figure 2.** Decision tree representing the official procedure for “lack of train power” incident

Thus, we adopted a tree representation, made of two types of elements: the **actions** and the **contextual nodes** which select a branch depending on the knowledge about the current context. Figure 2 shows the decision tree representation of the procedure for “train lack of power” solving (the meaning of the boxes is not important here but can be found in Annex 1 and in [24]), it suffices to say that the rectangular boxes are actions and the circles, contextual nodes.

In summary, our representation is inspired by decision trees, but mainly differs on two points. Firstly, our trees have no chance nodes but contextual nodes. Secondly, there are no probabilities. This representation shows several important characteristics that have some consequences on the size and structure of our tree:

1. As said in section 2.2, operators use many contextual elements to perform their choice. This leads to a large number of practical strategies, even for the same incident. This multiplies the number of branches and the tree grows rapidly.
2. The operators prefer to gather a maximum of information before making their decision. This behavior postpones most of the actions to the end of the branches of the tree [2, 30]. This observation is very close to the observation of Watson and Perera [39] who consider a hierarchical case representation that holds more general contextual features at its top and specific building elements at its leaves.
3. The operators privilege actions allowing to reach common intermediary situations. Thus, they can reuse common strategies to clear the incident. Graphically, the tip sequence of actions is often repeated from one branch to another.
4. Several action sequences are executed in the same order in different situations (paths).
5. Some actions could be done in different order, but must precede a given one. For example, before linking two trains, both have to be emptied, but the order in which they are emptied does not matter (partial order on the actions).

The large expansion of the tree structure does not easily allow to represent highly-contextual decision making in complex applications. In the next section, we explain the modification we have done, based on the characteristics discussed above, to obtain a manageable structure for representing operational knowledge on incident solving on subway lines.

### 3.3. From contextual trees to contextual graphs

First, we can reduce the number of objects in the structure by replacing repeated sub-sequences of actions (characteristics #3 and #4 above) by a single object called **macro-action**. The choice made for defining the different macro-actions is based on common sub-procedures known by the operators, such as “linking trains,” “return to end-station without travelers”... The principle of the replacing is explained on Figure 3.



Figure 3. From a sequence of actions to a macro-action

This replacement simplifies the lecture of the tree, but do not reduce the structural expansion of the tree. Secondly, relying on the previous observation #3, we can merge the branches of the tree as soon as the sequence leading to the end of the incident are similar, as shown in Figure 4.

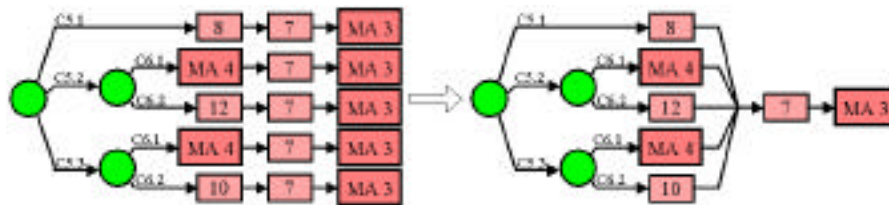
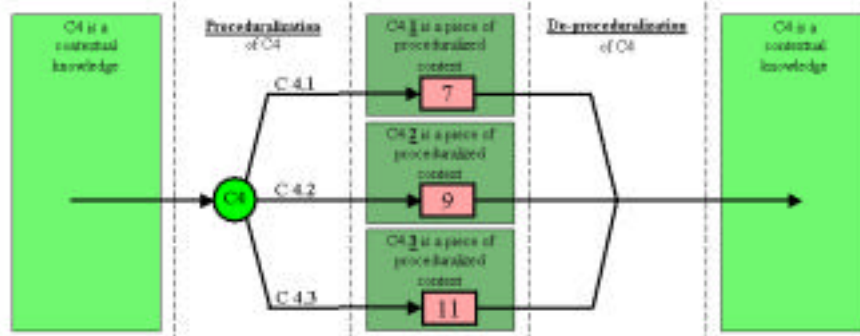


Figure 4. From tree to graph

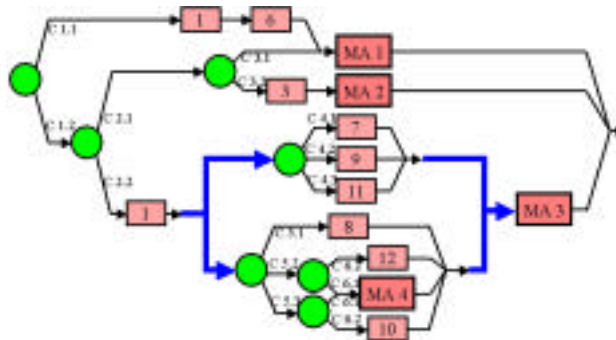
Cognitively speaking, this amount to use a scarcity principle that leads the operators to try to reuse well-known procedures as soon as possible. This operation has several consequences on the structure of the representation and on the meaning of the model.

1. We no longer face a tree but a graph. This graph is oriented without circuits, with exactly one root and one goal because the operators have only one goal (clear the incident and go back to normal exploitation) and the branches express only different strategies, depending on the context, to achieve this goal. The graph structure moreover allows extending of the representation.
2. The size of the structure is now under control and the consideration of a new contextual element will add some elements in the graph, but not drastically increase its size.
3. The change of the structure introduces a dynamics comparable to the dynamics of the change between proceduralized and contextual knowledge. Indeed, when two branches are merged, it means that the undertaken actions led to a common situation from different contexts. The contextual elements attached to the different branches are proceduralized at the diverging node. They stay in this state for the different action sequences, because they intervene in the branch decisions. Finally, they are de-proceduralized when the branches are merged. By this way we explicitly express the life duration of the contextual elements (Figure 5).



**Figure 5.** Proceduralization and de-proceduralization

Even in a part of a subgraph it may happen that the actions are only partially ordered (see the above example of two trains on the same line which have to be cleared of the travelers whatever the order of the operations is.)



**Figure 6.** Contextual graph representing the official procedure for “lack of train power” incident represented in Figure 2.

We need to represent this issue; this is why we introduced the **temporal branching** symbols to represent action sequences that can be done in different order (characteristics # 5). This symbol is made of two parts: a divergent branching and a convergent branching; the parallel branches wear the temporally independent decision blocs. Finally we obtain the structure represented in Figure 6, that we called **contextual graph**.

In the contextual graphs that we have built for several incident types, we note that some parts of the different graphs are identical. Analyzing these sub-graphs we exhibit that they complete a common sub-goal (for example, when a train lacks of power and cannot restart alone, or when a train has no more brake, both trains need to be helped by another train). Such a representation by contextual graphs/sub-graphs (Figure 7) is very similar to the generic tasks proposed by Chandrasekaran [8]. The difference is that our tasks are not generic in the sense that they cannot be combined to lead to more integrated tasks as Chandrasekaran’s ones; they are rather elementary or atomic tasks that can only be linked or followed each other in any order.

### 3. Contextual Graphs in action

Most of examples in this study are drawn from a system (SART) that we designed by using contextual graphs for the Paris subway organization (RATP). The SART project [4, 24] aims at the design and development of such a decision support system for the operators in charge of the control of a subway line. This project is based on the interaction between the operator and the system and will ease their mutual comprehension. For this purpose, we tried to mimic the system reasoning on the



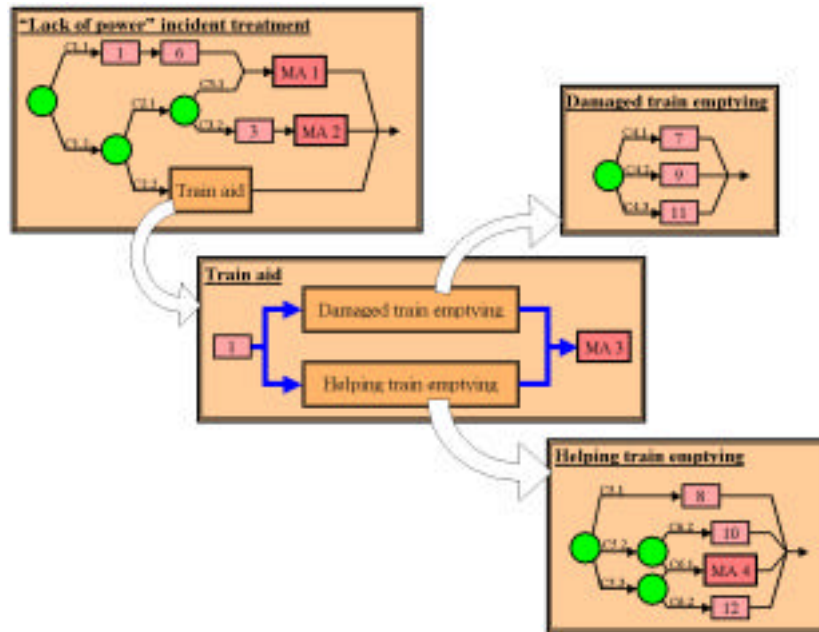
operators' one. Thus, we needed to analyze the operational knowledge used by operators and to record it in an adapted structure which can be easily understood by the operators and efficiently used by the computer. For an extensive description of the context of the case see [4] and [24].

Our representation by contextual graphs comes from this experience. This representation is more compact than trees and is well accepted by the operators [40], it simply represents the succession of actions to carry out in order to solve an incident; the different possible paths express the possible strategies according to the context.

The idea to use the dynamics of scenarios for modeling practices is somehow reminiscent of various scenario-based analyses in particular in strategic planning [20, 38], design [7], software engineering, especially in requirements engineering (e.g. [16, 29, 36]). For instance, the manual V1.3 of the Unified Modeling Language (UML) describes how activity graphs are used to model operations (action state). When an activity is modeled it is equipped with some internal procedures to execute and some actions which trigger other activities. UML let the possibility to execute actions either linearly or in parallel. This last possibility is quite similar to temporal branching in our contextual graphs.

Design also uses a notion of scenario but in a slightly different sense than in software engineering. The defining property of a scenario is that it projects a concrete description of the activity that the user engages in when performing a specific task, a description sufficiently detailed so that design implications can be inferred and specified [7]. Central to most scenario based design is a textual description or a narrative episode of use. A scenario is a narrative that describes someone trying to do something in some environment [17]. As such, it is a description of a context, which contains information about users, tasks, and environment. Scenarios are viewed by the user and may include social background, resource (e.g. disk space, time) constraints and background information. Scenarios seek to be concrete; they focus on describing particular instances of use, and according a user's view, they describe what happens, how it happens, and why [7]. By using a narrative it is possible to capture information about the user's goals, and the context the user is operating in. Scenarios help to express the complexity of the working context to design, and reveal some of the organizational trade-offs implicit in developing new automation [10]. In some sense contextual graphs offers an adequate representation for the capture of all these kind of scenarios. What is striking is that, in all these frameworks, as in our case, what is important is not the probability but the exhaustibility: all the possible paths must have been envisioned.

Each isolated sub-graphs is dubbed (operators know the corresponding procedures and agree on a name), as such, isolated and identified, it is more or less reminiscent of a script [34] [35] or a scheme of action but it moreover include the dynamics of proceduralization. These structures can be reused and adapted for another actions. For example, in Figure 7 the sub-procedure "*Helping train clearing*" is derived from the sub-procedure "*Damaged train clearing*" and adapted by the introduction of the fact that an available train may run to the next station, if this station is free, to evacuate its travelers in better conditions. In our system the sub-graphs are the elementary chunks of reasoning stored and reminded to the operators in case of incident. The adaptation to the context is made by the choice of the path. Some flexibility is left to the operators by the adaptation of the elementary actions contained in the graph, because the actions are generally defined by their result rather than by a detailed *modus operandi*.



**Figure 7.** Set of contextual graphs used while "lack of train power" incident resolution

#### 4. Conclusion

Our representation by contextual graphs is inspired from practice. We observed that contextual information matters more than probabilities to make a decision in an operational setting. Thus, we adapted the notion of decision tree to take into account contextual representation and its dynamics. We simplified the tree representation, thanks to two notions, namely macro-action and temporal branching. Going one step beyond the computer representation of the reasoning relying upon context and contextual graphs, we have pointed out that some sub-graphs represent usual procedures or scheme of action. These sub-graphs, beyond the fact that they give a simple computer representation of reasoning, have a deep meaning for operators and are significant, even drawn out of the context of a given incident. We thus are able to propose a set of interrelated contextual graphs that incorporate the notion of context in any problem solving (e.g. an incident solving) and represent the dynamics of the proceduralization de-proceduralization process.

#### 5. References

1. Anderson J.R. Rules of the Mind. Hillsdale N.J : Lawrence Erlbaum, 1995.
2. Brézillon, P. and Pomerol, J.-C. Using contextual information in decision making. *Context Sensitive Decision Support Systems*, Berkeley, D., Widmeyer G., Brézillon P. and Rajkovic V. (Eds.), London: Chapman and Hall, (1998) 158-173.
3. Brézillon, P., and Pomerol, J.-C. Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62, 3 (1999), 223-246.
4. Brézillon, P., Gentile, C., Saker, I., and Secron, M. SART : A system for supporting operators with contextual knowledge. *First International and Interdisciplinary Conference On Modelling and Using Context (CONTEXT'97)*. Rio de Janeiro : Federal University of Rio de Janeiro, 1997,

209-222 (also available at <http://www-poleia.lip6.fr/~brezil/Pages2/CONTEXT-97/index.html> ).

5. Brézillon, P., Pomerol, J.-C., and Saker, I. Contextual and contextualized knowledge: An application in subway control. *Special issue on Using Context in Application, International Journal on Human-Computer Studies*, 48, 3 (1998), 357-373.
6. Brown, J.S., and Duguid, P. Organizational learning and communities of practice : towards a unified view of working, learning and and organization, *Organization Science*, 2, 1 (1991), 40-57.
7. Carroll, J.M. (Ed.) *Scenario-Based Design*, New-York : Wiley , 1995.
8. Chandrasekaran, B., Johnson, T.R., and Smith, J.W. Task-structure analysis for knowledge modeling. *Communications of the ACM*, 35, 9 (1992), 124-137.
9. Clancey, W.J. Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies*, 11, 1 (1983), 25-49.
10. Dearden, A., Harrisson, M., and Wright, P. Allocation of function: scenarios, context, and the economics of efforts. *International Journal of Human-Computer Studies*. 52, 2 (2000), 289-318.
11. Decortis, F., Noirfalise, S., and Saudelli, B. Activity theory, cognitive ergonomics and distributed cognition : three views of a transport company. *International Journal of Human-Computer Studies*. 53, 1 (2000), 5-33.
12. Degler, D., and Battle, L. Knowledge management in pursuit of performance : the challenge of context. *Performance Improvement Journal*, 39, 6 (2000), 25-31.
13. Grimshaw, D.J., Mott, P.L., and Roberts, S.A. The role of context in decision making: some implications for database design. *European Journal of Information Systems*, 6, 2 (1997), 122-128.
14. Hasher, L., and Zacks, R.T. Automatic processing of fundamental information : the case of frequency of occurrence. *American Psychologist*, 39 (1984), 1372-1388.
15. Howard, R.A. From influence to relevance to knowledge. In Oliver and Smith (Eds.), *Influence Diagrams, Belief Nets, and Decision Analysis*. New York : John Wiley & Sons, 1990, 3-23.
16. Jarke, M., Rolland, C., Sutcliffe, A., and Dömges, R., (Eds.). *The Nature of Requirements Engineering*, Aachen : Shaker Verlag, 1999.
17. Karat, J. Scenario use in the design of a speech recognition system. In *Scenario-Based Design: Envisioning Work and Technology in System Development*. J.M. Carroll Ed. New-York : J. Wiley & Sons (1995), 109-133.
18. Kolodner, J. *Case-Based Reasoning*. San Francisco : Morgan Kaufman, 1993.
19. Leake, D.B. Case-based reasoning: Experiences, lessons, and future directions. In *CBR in Context: The Present and Future*. Menlo Park: AAAI Press/MIT Press, 1996.

20. Leemhuis, J.P. Using scenarios to develop strategies. *Long Range Planning*, 18 (1985), 30-37.
21. Leplat, J., and Hoc, J.M. Tâche et activité dans l'analyse psychologique des situations. *Cahiers de Psychologie Cognitive*, 3 (1983), 49-63.
22. Levesque, H.J. A logic of implicit and explicit belief. *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI-84)* (1984), 198-202.
23. Oliver, R., and Smith, J. (Eds.) *Influence Diagrams, Belief Nets and Decision Analysis*. New York : John Wiley & Sons, 1990.
24. Pasquier, L. Modélisation de raisonnements tenus en contexte et application aux agents d'aide à la gestion d'incidents de SART. LIP6 Research Report N.2000-010. Paris: LIP6, 2000.
25. Pasquier, L., Brézillon, P., and Pomerol, J.-C. Context and decision graphs in incident management on a subway line. *Modeling and Using Context (CONTEXT-99)*. In: Lecture Notes in Artificial Intelligence, N° 1688, Springer Verlag (1999), 499-502.
26. Pasquier, L., Brézillon, P., and Pomerol, J.-C. From representation of operational knowledge to practical decision making in operations. *Decision Support Through Knowledge Management*. S. Carlsson, P. Brézillon, P. Humphreys, B.G. Lundberg, A. McCosh & V. Rajkovic (Eds.). Edsbruk (Sweden) : Akademitryck AB, 2000, 301-320.
27. Pearl, J. *Probabilistic Reasoning in Intelligent Systems*. San Mateo : Morgan Kaufmann publishers, 1988.
28. Pomerol, J.C. Artificial Intelligence and Human Decision Making. *European Journal of Operational Research*, 99 (1997), 3-25.
29. Pomerol, J.-C. Scenario development and practical decision making under uncertainty, Application to requirements engineering. *Requirements Engineering*, 3 (1998), 174-181.
30. Pomerol, J.-C. Scenario development and practical decision making under uncertainty. *Decision Support Systems*, 31 (2001), 197-204.
31. Pomerol, J.-C., and Brézillon, P. Dynamics between contextual knowledge and proceduralized context. *Modeling and Using Context (CONTEXT-99)*. In: Lecture Notes in Artificial Intelligence, n° 1688, Springer Verlag (1999), 284-295.
32. Raïffa, H., *Decision Analysis*. New-York: Mac Graw Hill, 1968.
33. Savage, L.J. *The Foundations of Statistics*. New-York: John Wiley and Sons, 1954.
34. Schank, R.C. *Dynamic Memory, a Theory of Learning in Computers and People*. Cambridge: Cambridge University Press, 1982.
35. Schank, R.C., and Albersohn, R.P. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*, New-York: L. Erlbaum, 1977.
36. Sutcliffe, A. Scenario-based requirement analysis. *Requirements Engineering*, 3 (1998), 48-65.

37. Tiberghien, G. Context and cognition: introduction. *Cahier de Psychologie Cognitive*, 6, 2 (1986), 105-119.
38. Wack, P. Scenarios: uncharted waters ahead. *Harvard Business Review*, 63, 5 (1985), 72-89.
39. Watson, I., and Perera, S. A hierarchical case representation using context guided retrieval. *Knowledge Based Systems Journal*. 11, 5-6 (2000), 285-292.
40. Zanarelli, C., Saker, I., and Pasquier, L. Un projet de coopération ergonomes/concepteurs autour de la conception d'un outil d'aide à la régulation du trafic du métro. *Ingénierie des Connaissances (IC'99)*. Palaiseau: Association Française pour l'Intelligence Artificielle (1999), 161-170.

**Table 1** : Some Actions (A) and Contexts (C) in case of incident on a metro line.

Actions		Contexts	
1	Residual traffic regulation	C11	Immediate repair possible
2	Damaged train continues with travelers	C12	Immediate repair impossible
3	Damaged train continues with travelers until a steep incline	C21	Enough motor coaches available
4	Damaged train restarts without travelers	C22	Not enough motor coaches available
5	Stable damaged train at end station	C31	No steep incline between damaged train and end station
6	Repair damage	C32	Presence of steep incline until end station
7	Exit of the travelers out of the damaged train		
8	Exit of the travelers out of next train	C41	Damaged train at station
9	Exit of the travelers out of damaged train via available cars	C42	Damaged train under tunnel
10	Exit of the travelers out of next train via available cars	C43	Damaged train partially at station
11	Exit of the travelers out of damaged train via track		
12	Exit of the travelers out of next train via track	C51	Next train at station
13	Next train joins damaged train	C52	Next train under tunnel
14	Link both trains	C53	Next train partially at station
15	Convoy return to end station		
16	Disassemble convoy	C61	Presence of a station between damaged train and next train
17	Next train goes to next station	C62	No station between both trains

Macro-actions	Actions lists
MA 1	Damaged train continues service
MA 2	Damaged train stops service
MA 3	Make a convoy with damaged train and next train
	Actions 2 and 5
	Actions 7, 4 and 5
	Actions 13, 14, 15, 5 and 16

## 7 Biographies

**Jean-Charles Pomerol** is Professor of Computer Science and one of the Vice-Chairpersons for Research of the University Pierre and Marie Curie in Paris (UPMC). He was formerly the head of the common UPMC-CNRS Artificial Intelligence laboratory of Paris, now a part of the Computer Science laboratory of the UPMC (LIP6). Jean-Charles Pomerol defended his “Thèse d'Etat” in convex analysis in 1980. Then, he turns to decision theory and decision support systems. His current interests are about the design and development of 'intelligent' decision support systems. He is the author or co-author of many papers and four books, concerning expert systems, decision support systems and recently about practical multicriterion decision making (Kluwer Pub, 2000). He is also the chief editor of the Journal of Decision Systems and of the “Revue Française d'Intelligence Artificielle” (Hermès-Science Pub).

**Patrick Brézillon** is a fellow researcher of the National Center for Scientific Research in France (CNRS). In 1983, he received his “Thèse d'Etat” in natural sciences at the Pierre and Marie Curie University. His thesis is devoted to the mathematical modeling of the calcium metabolism as a self-oscillating nonlinear model. Now, he aims at merging mathematical modeling with Artificial Intelligence and is interested in topics such as cooperation, context, explanation and incremental knowledge acquisition in the framework of intelligent assistant systems. He strongly contributed to found several years ago and, he is now one of the leaders of the worldwide research community working on of the notion of context representation and use for in real-world applications. Further details about Brézillon's activities can be found at <http://www-poleia.lip6.fr/~brezil/>.

**Laurent Pasquier** is engineer in Automatics and Operational Research. He is completing a PhD thesis under the direction of P. Brézillon, at the RATP (Parisian public transportation company) and LIP6, entitled "Modeling of context-based reasoning, and application to incident management on subway lines." He is interested in Knowledge Representation and Decision Support System design.