Modeling and Using Context: 25 Years of Lessons Learned

Patrick Brézillon

2

Modeling and Using Context: 25 Years of Lessons Learned[1]

This chapter presents the results of 25 years of research on modeling and using context in real-world applications. The first milestone was an operational definition of context relating it to a focus (problem-solving, decision-making, task realization). The focus provides context with links between context and knowledge, a division between contextual knowledge and external knowledge with respect to a focus, and a dynamic linked to reasoning evolution. Conversely, context makes focus more explicit. The second milestone was a context-based representation of decision trees as contextual graphs with contextual elements (expressing the focus) instead of chance or event nodes. The operational definition of context opens up a conceptual framework that can be associated with an implementation framework. The third milestone was the development of a context-based representation formalism that led to the software Contextual Graphs (CxG) that has been used in about 20 real-world applications. The fourth milestone concerns an extension of the CxG representation formalism for representing group activity with a cyclic use of a CxG for simulation purposes. The formalism now allows a modeling of "nonlinear" behaviors of a group (e.g. interaction, negotiation, etc.) as well as of a user (e.g. checking alternatives). The fifth milestone opens the way to Context-based Intelligent Assistant Systems.

2.1. Introduction

We started research on context in two projects (SEPT and SART) in the technological domain in which exist formal models of equipment to monitor. In the first project, SEPT with *Electricity de France* (Bau and Brézillon, 1992), we rapidly discovered that having a formal model was not sufficient because a number of parameters were not controllable or not considered in the formal model, although they intervene indirectly in it. For example, equipment in an Extra High Voltage (EHV) substation in a narrow valley in mountains cannot function as if it were on a plain. More globally (the observation was reinforced in the SART project (Brézillon *et al.*, 2000) for the subway in Paris and Rio de Janeiro), if it is possible to model formally the functioning of equipment in an EHV substation, some external elements have to be considered: the actor, the task the actor must realize (e.g. monitoring of equipment

[1] Patrick Brézillon

functioning in the substation), the situation in which the task is realized, and the local environment where there are resources available for the task. All these external elements may constrain the normal functioning of what is in the focus. Actor, task, situation and environment are sources of contextual knowledge that matters for a focus.

As a consequence, context is strongly related to knowledge and reasoning, both in models and in the heads of users. Working on real-world applications obliged us to avoid arbitrary simplification in modeling context, and our interest in decision-making led us to establish an operational definition of context. It was then possible to really model context in terms of contextual elements that had a concrete meaning for actors with whom we worked in a very large spectrum of domains[2]: power systems, subway exploitation, enology with wine making, computer security, collaborative understanding, car driver support, web site analysis, self-training of car drivers, medical image access, contextualization of platforms of interoperable open source tools for enterprises, database administration, a strategic infrastructure project in transportation, a cognition-driven explorer for histopathology for breast cancer grading, process and sharing of medical images, and contextualized support on a tablet for the transmission of command information.

The operational definition of context and the study in depth of different aspects of context have allowed the design and implementation of a context-based representation formalism called Contextual Graphs (CxG). The CxG formalism has been used with success in all the applications cited in the previous paragraph. We only give in this paper the main features of applications needed to understand some lessons learned about context and its role in applications; most of the key references on the CxG formalism are given in (Brézillon, 2018).

In the two first applications, the unique consideration of a task model (e.g. technical manuals in the SEPT project and incident reports in the SART project) was insufficient for developing a competent computer software: it is important also to consider the way in which the task is realized by an actor. In the SART project, operators give a great importance to contextual knowledge for monitoring subway lines in the situation of an incident (e.g. at rush hour). An operator said, “when I am informed of an incident on my line, I first look to the current state of the whole line to first gather contextual information”, and then make decision accordingly. Landauer and Bellman (1993) similarly conclude that the process of problem solving presents three major steps: find_context, pose_problem, and study_problem. The context includes conditions on the existence of certain external information and partial mappings between elements.

[2] The list of the publications (and PDF for some of them) on most of all the projects and applications is given at http://www-poleia.lip6.fr/%7Ebrezil/Pages2/Publications/index.html

Operators interpret the task model (procedures of the company that are supposed to be mandatory) according to the contextual information collected to determine how to solve the incident with minimal cost at policy and strategic decisional levels (e.g. prioritizing a fast solution to avoid a degraded situation in the SEPT project, and traveler security and traffic regularity in the SART project). Again, operators use contextual knowledge from sources like the task, the situation and the local environment.

This remainder of this chapter is organized in three parts. Section 2.2 presents knowledge in action to point out the relationships between operational knowledge and (1) contextual knowledge, and (2) mental models of actors. This leads to a discussion of the modeling of operational knowledge and, if it is not possible, the contextual alternative. Section 2.3 presents context in action through its conceptual modeling under different aspects and the implementation in the Contextual Graphs (CxG) representation formalism. Section 2.4 develops the action of context in real-world applications, first with respect to a precise focus (task realization, decision making, problem solving), and second with respect to actors responsible for a focus alone or in group. The section ends on a discussion of the extension to the CxG formalism for modeling group activity. Section 2.5 sums up the lessons learned and the results obtained during these 25 years.

2.2. Knowledge in action

2.2.1. Operational knowledge and contextual knowledge

Brézillon and Pomerol (2001) said that context is knowledge about the instantiation of *know how*. It is the framework that reveals *know how*. In other words, knowledge is the information that is integrated and understood in the mind of an actor, and information results from data interpretation. The focus allows differentiating two parts of context, namely, contextual knowledge and external knowledge.

Contextual knowledge constitutes a set of elements more or less related to a focus in a flat way. *External knowledge* concerns elements that are not important for the focus at the moment we consider it. The frontier between contextual and external knowledge is porous, and elements move between contextual knowledge and external knowledge as the focus progresses. The selection of the elements of the contextual knowledge concerning the focus is a kind of first instantiation of the contextual knowledge or background context that needs some further specification to perfectly fit the task at hand. The precision and differentiation brought to the contextual knowledge is also a part of the proceduralization process. “Knowing how” is “instantiating in doing”, and contextual knowledge obeys a dynamic of instantiation and proceduralization during action. The important issue is the passage from contextual knowledge to *proceduralized context*, the part of context used in the focus.

The proceduralization aspect corresponds to the instantiation of the contextual elements entering in the focus (see also Grimshaw *et al.*, 2000).

Contextual knowledge is personal to an actor and is evoked by situations and events. It is something that is stored in long-term memory, and recalled as a whole at some step of task realization. Such a conceptual framework supposes a uniform representation of the elements of knowledge, reasoning and context. The actor's focus is guided by a mental representation that the actor has of the task realization in its current step, the situation in which the task is being realized and the local environment in which resources are available. The mental model of the actor represents the personal context in which the actor interprets the focus. It describes something important about knowledge that is not captured in traditional knowledge measures, such as its operational and dynamical organizations. Knowledge organization is different from an actor to another one.

2.2.2. Operational knowledge and mental models

In a real-world application, an expert develops a mental model based on their experience accumulated from task realizations (use of domain knowledge in different contexts). Experience relies on operational knowledge as well as domain knowledge.

In the TACTIC project (Kabil *et al.*, 2015), experts were asked to develop mental maps as a semi-structured expression of their mental models (this was also done in another project in medicine, Brézillon *et al.*, 2014). Their mental maps show depth- and breadth-first strategies for modeling how they realized their task. Usually, reasoning is between a "depth-first" strategy and a "breadth-first" strategy. The "depth-first" strategy goes to the finest possible granularity on a line of reasoning in order to anticipate as much as possible the course of events from a maximum of contextual elements and their instantiations. It assumes that we know what to do and how to get there quickly. This strategy allows studying the technical feasibility of an approach as well as the needs in terms of resources, and, in a second step, gradually expands this approach. Conversely, the "breadth-first" strategy is applied when it is necessary to consider first all possible options before making decision. Mental models in a breadth-first strategy are interesting for also detecting, for example, weak signals because experts keep an "open mind". Their focus is not on the task realization in an isolated way, but rather on how the realization is placed in its context. Such experts reason at a strategic decisional level and do not consider low-level details of the task realization. Conversely, expert maps made by experts in a depth-first strategy can be used to decide rapidly at the tactical level.

The breadth-first strategy is observed in the expert maps of experts that maintain the more relevant contextual elements, even if not directly necessary in the task realization. For example, "environment" was considered by an expert only as a part of "situation", even if environment is the main source of contextual elements on the battlefield map. The reason is that experts consider environment through what they really need to know for realizing their task. For example, it is only when there is contact with the enemy that participants zoom in on the location of the enemy (e.g. in a city or in landscape). The participants thus are more interested in an external event through its effect on task realization rather than by its origin.

The interplay between depth- and breadth-first strategies is similar to global search (FlexMIm project, Brézillon *et al.*, 2014). Both are based on a cycle beginning with a phase of fast exploration for finding zones of interest (global search) followed by a phase of fine-grained analysis of the zones judged of eventual interest (local search). Context-aware systems also distinguish two types of context: the "local" context that is close of the focus and highly detailed, and the "distant" context that is general (with less details). In the spirit of Sowa's (2000) conceptual graphs, all the concepts can be expanded in the local context and aggregated in the distant context. Local details are needed for the local interactions with a structure. The global context is needed to tell the actor what other parts of the structure exist and where they are. Global information is important even in the simple interpretation of local details.

2.2.3. Modeling operational knowledge

The two types of reasoning (breadth-first strategy and depth-first strategy) are illustrated on Figure 2.1 and 2.2 respectively (only the general form of the two maps is important here). The mental model of the task “Give a reconnoitering order” (find out what the enemy is doing) was elaborated by knowledge acquisition techniques from a panel of ten experts that routinely used a battlefield simulator for different purposes. Each expert had expressed in a mind map the knowledge they generally used for realizing the task and simplified the map for the specific simulation (i.e. in a specific context). Figure 2.1 shows the map made by a tactical expert (a simulator trainer) and Figure 2.2 shows the map of an operational expert (a developer). Grey items correspond to the contextual elements not used in the specific scenario on which they had to work.

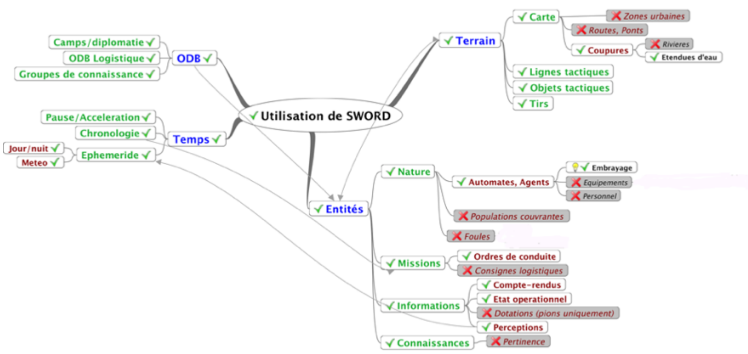

Figure 2.1. *Knowledge organization in a tactical view (Brézillon, 2018)*

Figure 2.1 shows the knowledge organization of the tactical expert (the general tree), which is very close to that of the operator interacting with the simulator (the end-user). The focus is limited to the position of the mission in the context of the battlefield, and lets all other problems remain at the periphery of the mental model. The expert also introduces links between leaves in the mind map to express his strong compilation of the contextual elements. For example, the item “perceptions” is linked to the item “Ephemeris” because the useful information on weather and day/night only addresses the question of the soldier’s vision around him.

Figure 2.2 shows how the operational expert organizes (the general tree) his knowledge in depth for analyzing more what the execution of an action on the simulator is than for analyzing what the action is on the battlefield (i.e. in the simulation).

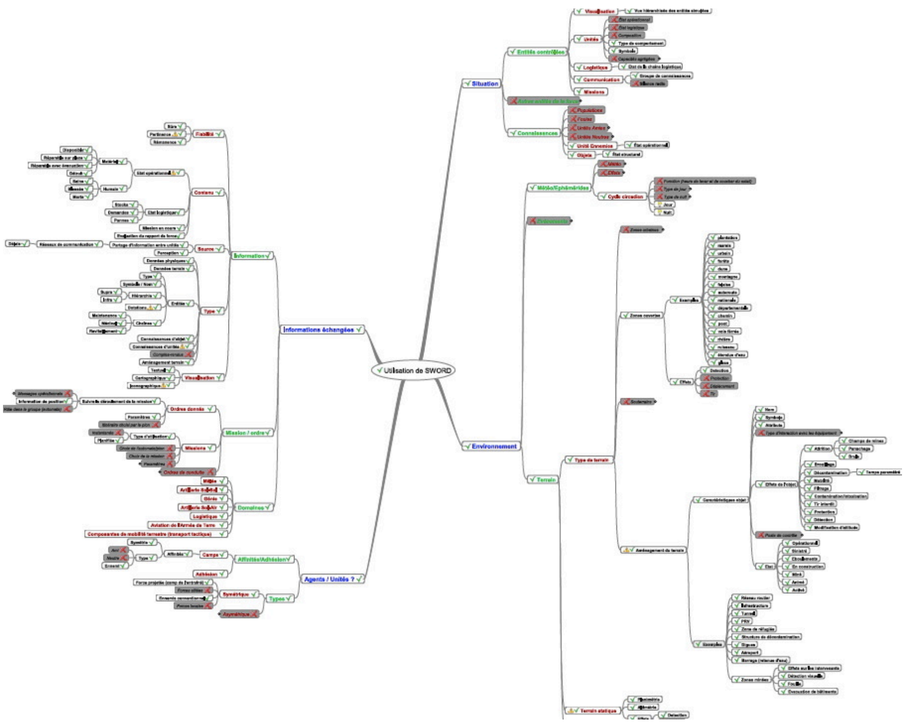

Figure 2.2. *Knowledge organization in an operational view (Brézillon, 2018)*

2.2.4. Indirect modeling from experience reuse

The direct reuse of known practices instead of restarting from scratch with a procedure is observed in domains where experts’ experience is preferred to domain knowledge that is weakly formalized, if at all. Experts rely on highly-compiled experience, and the reuse of their experience is never direct because it was obtained in numerous specific contexts. Thus, reusing a past experience implies that the actor follows the three steps:

(1) "contextualize" (i.e. find the contextualization of) a known past experience in relation with the experience at hand,

(2) "decontextualize" the past experience for finding the model (i.e. retrieving the contextual elements from their instantiations), and finally

(3) "recontextualize" the model of the past experience by re-instantiating its contextual elements in the context at hand.

This process of contextualization-decontextualization-recontextualization (CDR) is used in different works.

Brannen (2004) analyzes the two first attempts of the Walt Disney Company made at internationalization: Tokyo Disneyland (1983) and Disneyland Paris (1992). The goal was to implant a strict copy of the initial park at Anaheim (USA) in Japan and in France, and the internationalization challenge was assessing the fit of what they wish to transfer abroad with the new host environment. Generally, firms transfer their whole organizations, including business models, steeped in image-based, ideological, people-dependent management that are even more closely related to context and sociocultural environment in which they are enacted. However, if the source is significantly foreign from target, the transferred assets may not fit the target context in the host country. It was the case of Disneyland Paris, but after the positive experience with Tokyo Disneyland (no CDR needed), the context of the French target was too different (recontextualization was mandatory, as discovered several years later).

Fan *et al.* (2011) applied the CDR process in the search for a scientific workflow (SWF) in virtual screening of the H5N1 virus (see Figure 2.3): A researcher extracts a successful SWF from a repository that he considers as close to his focus (phase of contextualization), extracts the corresponding SWF model (phase of decontextualization), and, finally, instantiates the SWF model in the working context at hand (phase of recontextualization). Then, a workflow engine ensures the execution phase in which input data is consumed according to the instantiated SWF and output data are produced as results. If it is a success, the new scientific workflow is put in the repository for other scientists. Otherwise, a new iteration is needed by reconsidering either the phase of decontextualization for finding a new model, or by trying a different recontextualization of a new scientific workflow.

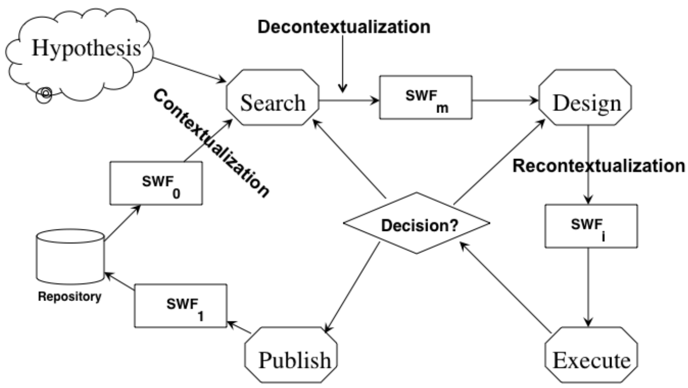

Figure 2.3. ***Flexibility inside workflow lifecycle (from Fan et al., 2011)***

The experience acquired by the researcher during this process relies on context management. This modeling allows making context, consequently operators' behavior, explicit.

2.2.5. Lessons learned

The main lessons learned were the following:

- Tactical experts tend to create contextual graphs with a predominant parallel structure (breadth-first strategy) while operational experts are more prone to model their problems by developing contextual graphs with a predominant series structure (depth-first strategy). The main point is that the former reasons on the basis of their experience (i.e. structured knowledge), when the latter assembles contextual elements more in a reactive way rather than in a pro-active one.
- The quality of the task-realization model depends directly on the expert's knowledge about the task realization, the situation and the local environment. The operational expert focuses on the narrow understanding of the problem, while a tactical expert sees the problem at hand among other problems for finding a solution.
- The tactical expert thinks in an organized and structured way, avoids assumptions leading to unrealistic scenarios, and identifies actions that can be performed in parallel. The operational expert executes actions sequentially without paying attention to the logic of the task. The operational expert may add sub-tasks, or change the order of the subtasks, while the tactical expert applies the general scenario of collecting first contextual information and second applying the better strategy for the proceduralized context found.
- The tactical expert has a holistic view of the task realization and discriminates relevant from useless contextual elements in a kind of "cognitive simulation"

to look ahead in the task realization. The operational expert has a fragmented view of the task realization and, thus, has reasons reactively, discovering the task during its realization, sometimes missing some details, considering irrelevant aspects or establishing no links between the different steps.

- An expert map, like in Figures 2.1 and 2.2, expresses what could be a working context, with its contextual elements, their possible instantiations and their organization. Studying the expert maps shows there is no unique model of the working context (see the difference between the expert maps between Figures 2.1 and 2.2). Clearly, the organization of the contextual elements depends on the experience of the actor that must realize the task. The tactical expert is not interested by the details and retains only the direction to follow. Conversely, the operational expert follows one direction and analyzes at a fine-grain what to do.

2.3. Context in action

2.3.1. Conceptual modeling

Brézillon and Pomerol (1999) defined context as "what constrains the focus without intervening in it explicitly". The focus could be decision-making, problem solving, or task realization. This definition implies two consequences: first, one cannot speak of context in an abstract way – it is always relative to a focus; and second, context and focus are interdependent – the context makes the focus explicit and the focus defines the relevant contextual knowledge needed at each step. Making context explicit implies that our modeling of a task concerns its realization as a process and not just a formal description of the task model. From an implementational viewpoint, context is an aggregation of contextual elements coming from four sources, namely the user, the task, the situation and the local environment.

The first key point of context modeling is the association of conceptual and operational views, ensuring a direct implementation of the conceptual model of context. Figure 2.4 shows our approach from the conceptual view to the operational view. This double viewpoint (conceptual and operational frameworks) constitutes the ground of all our (successful) research in real-world applications.

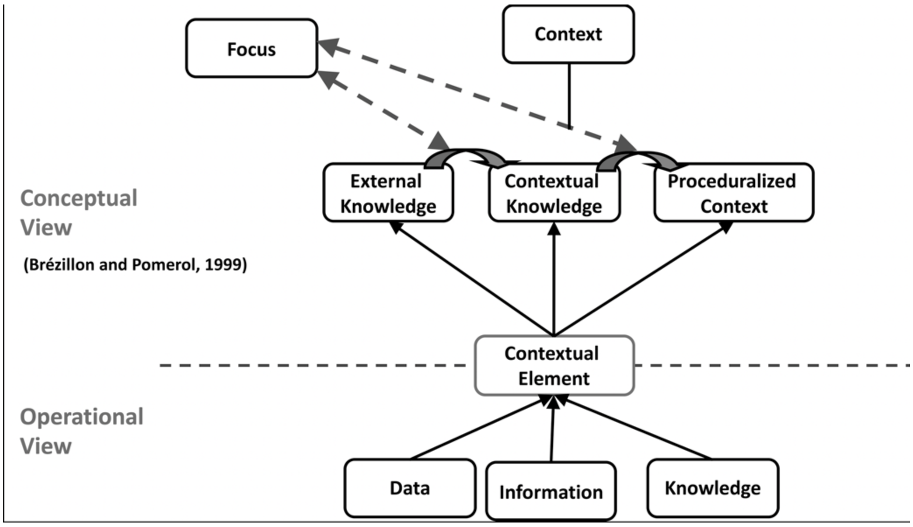

Figure 2.4. *Conceptual and operational views on contextual elements*

2.3.2. A typology of contexts

The *working context* contains the contextual elements that an actor associates with their focus (e.g. task realization), all the known values of each contextual element and a chosen instantiation. Arriving at a contextual element, the practice development needs its instantiation to select the right path to follow and the action to execute. The instantiation can be known prior to the task realization, provided by the actor when (and if) needed, or found by a system using the contextual graph in the local environment.

The *shared context* concerns a task realized by a group of actors. The shared context is elaborated from the working contexts of the group members. It evolves continuously according to the group focus and the additions of contextual elements by group members. A group task realization starts by establishing the shared context. The group members first gather a maximum of contextual elements without looking for a global picture (in a kind of brainstorming), a member proposes a contextual element supposed to be relevant and other actors accept it or ask for explanation. For accepting the new contextual element, all group members must be able to relate it to their individual working contexts. Thus, the joint development of the shared context allows group members to keep their mental models compatible (Karsenty and Brézillon, 1995). An example of collaborative building of a proceduralized context is shown in an application Section 2.3.3.3.

The phase of shared-context building does not concern only contextual knowledge but also external knowledge that the proposal originator has (e.g. “I had the opportunity to visit a place concerned with this topic a couple of years ago”). When group members reach a consensus on their shared context, they enter the second phase of the process by organizing, assembling and structuring most of the contextual elements of their shared context in a proceduralized context allowing to reach a group focus. In the SART project, operators work 8 hours daily in the control room of a subway line and are replaced by other operators that receive a debriefing on what happened before their arrival and the difficulties encountered. This sharing of contextual information among actors is essential for assuring a continuity of the normal operation of the subway line (and in other places indeed). It is also the case in several projects we worked on after the SART project. Group activity is mandatory when either the task model is too complex for a single actor (e.g. the monitoring of a nuclear plant) or there does not exist a real task model (like in medicine and in domains like wine making). In some domains, such as scientific workflows, actors have no task model at all and are obliged to have an approach of contextualization-decontextualization-recontextualization.

A *proceduralized context* is quite similar, in the spirit at least, to a chunk of knowledge (Miller, 1956; Schank 1982), and, in its creation, to Clancey’s (2002) view of diagnosis as the construction of a situation-specific model. A proceduralized context is built progressively with the practice development. This “chunk of contextual knowledge” is activated and structured to make decision making or task realization (Pomerol and Brézillon, 1999). Concretely, the proceduralized context is the ordered series of contextual elements that are instantiated during practice development. If contextual knowledge is background knowledge for the focus, the proceduralized context is foreground, that is, immediately useful at the given step of the focus.

It is possible to retrieve how the proceduralized context was built, the reasons behind choices (contextual elements considered and their instantiations), the alternatives abandoned (actions corresponding to the values of the contextual elements not retained), etc. A proceduralized context is either static, when related to an item, or dynamic, when related to the development of the practice (see section 2.4).

2.3.3. About contextual elements

2.3.3.1. Organization around the focus

The onion metaphor has been proposed in the SART project (Brézillon *et al.*, 1997, 2000) to illustrate context organization. The focus (the successive steps in incident solving) is the onion heart, and we observed that contextual elements were organized in layers around the heart. Figure 2.4 illustrates the solving of the incident “Ill traveler in a train.” The step “Stop at the next station” is the onion heart, and dotted lines around it represent the successive layers of contextual elements. For instance, the contextual element “Procedures” is in the first layer, “Past experience” in the second one, which is an explanation for procedure, etc.

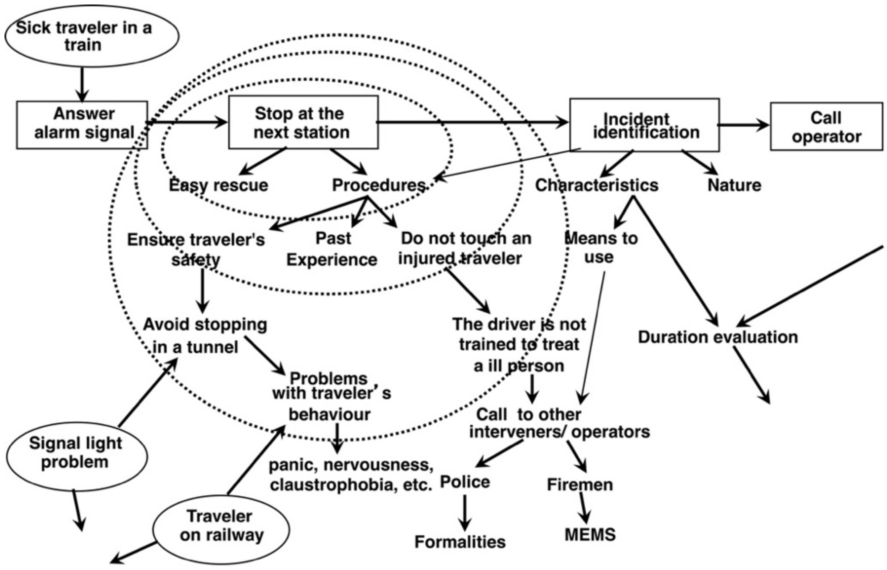

Figure 2.4. *Contextual organization in the incident “Ill traveler in a train”*

The lessons learned with the onion metaphor reveal several interesting points for our modeling of context for real-world applications:

- Only contextual knowledge is considered, which avoids the problem of the infinite dimension noted by McCarthy (1993), linked to the external knowledge.
- A step has a meaning within a context. Contextual knowledge does not intervene explicitly at this step but constrains it.
- Each contextual element itself has a meaning within a context. For example, the contextual element "Procedures" on Figure 2.4 has its own context with contextual elements like "Past experience" and "Do not touch an injured traveler." It is an illustration of McCarthy’s (1993) claim about the definition of a context in an outer context.
- Layers imply a partial ordering of contextual elements with respect to the focus. For example, “Procedures” (layer 1) < “Don’t touch an injured traveler” (layer 2) < “Driver not trained to treat a ill person” layer 3) < etc.
- Incidents may be linked each other through a sequence of ordered contextual elements. For example, “Ill traveler in a train” is linked to the incident

> "Traveler on the railway" through the contextual elements "procedure", "ensure traveler's safety", "avoid stopping in a tunnel", "problem with traveler's behaviors", this later contextual element being shared with the incident "Traveler on the railway."

The onion metaphor allows the development of explanations of different natures and at different levels of detail corresponding to the layers. It is useful in many different applications. In the wine-making application (Agabra *et al*., 1997), "far" contextual elements may be more crucial for the task realization than some close ones. For example, it is well known (in France) that the contextual element "weather" a few days before vintage—which is qualitatively far from "wine fermentation"—plays a central role in wine making. It is particularly important because, unlike the technical domain, often the task model may be too difficult to model formally (see section 2.2.4). For example, enologists consider the step of wine fermentation as a black box due to numerous and complex chemical reactions occurring during this step.

The onion metaphor shows a representation of contextual elements that allows remembering that even a distant contextual element can be more important than a close one. Conversely, fisheye views (Furnas, 1986; Pook *et al.,* 2000) also are one way of integrating context and focus into a single view on local details and global context simultaneously. However, the global context cannot be exploited as local details.

2.3.3.2. Instantiation of the contextual elements

The second key point of the context representation is the distinction between a contextual element and its instantiation, a distinction often not considered in other representation formalisms. The explicit introduction of contextual elements for representing contextual knowledge and the distinction with their values offer new perspectives: the acquisition of a new practice corresponds to the addition of either a new contextual element (learning by accommodation) or a new value for an existing contextual element (learning by assimilation), the latter case leading to only a refinement of the practice. Several types of value of a contextual element are possible, from qualitative to quantitative. For example, the contextual element "Temperature?" may have values like "warm" or "cold" but, according to the focus, quantitative values like "24.5°C" are possible instead of "warm".

During practice development, the instantiation of contextual elements may be altered by either an external or an internal event. The external event corresponds to an unpredicted event (e.g. the power supply being cut off), and thus is not represented in the task realization. An internal event occurs as the result of an action execution. For example, a fuse becomes too hot and blows. The change of instantiation of a contextual element modifies the working context, and may lead to a change of task to realize if the initial hypothesis is abandoned, and to consider the development of another practice in the task realization, or to modify the conclusion of the task realization.

2.3.3.3. Collaborative building of the proceduralized context

Figure 2.5 represents the building of a proceduralized context by two actors for a collective focus. Actor_1 introduces the contextual elements CE_1 from his working context into the shared context. Because CE_1 belongs to Actor_2's working context too, the contextual element is accepted. When Actor_1 proposes the introduction of CE_2, Actor_2, which has no reference to it in his working context, needs explanations for accepting CE_2 and both agree on CE_2' (compatible interpretation of CE_2 in the shared context). For accepting CE_2', the receiver builds an interpretation CE_2" in his working context. Once Actor_2 agrees to the contextual element, the contextual elements of the shared context are combined into a proceduralized context for addressing the focus.

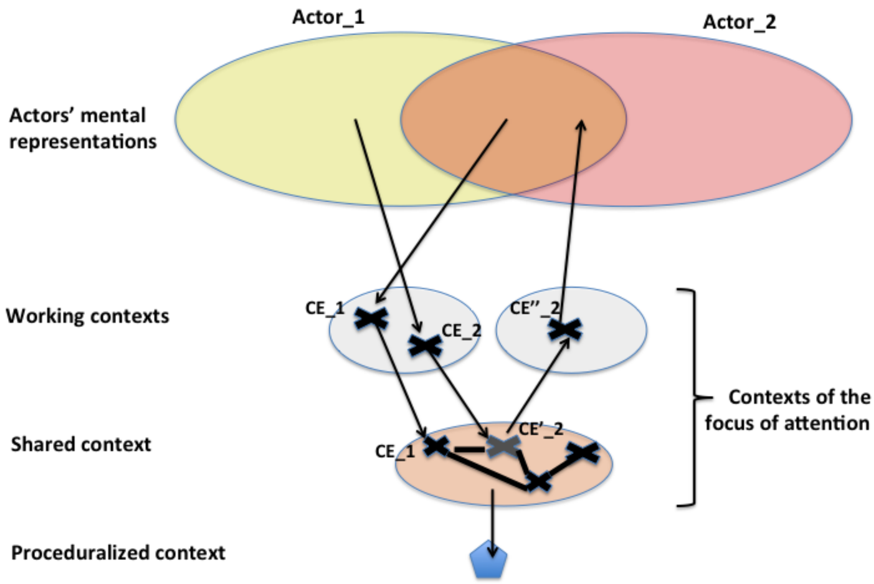

Figure 2.5. *Collaborative building of a proceduralized context*

Brézillon *et al.* (2006) describes an experiment for modeling verbal exchanges between two actors for collaboratively generating answers to questions like "How does the oyster make pearls?" and "How does water become mineral?". The collaborative process of answer building includes a first phase of building of the shared context of the collaboration. Each actor proposes elements of their working context. The other actor may agree, or ask for an explanation, and eventually negotiates with the first actor. Finally, actors agree on a shared context satisfying both of them. In a second phase, the actors organize, assemble and structure the contextual elements of the shared context to generate their proceduralized context that will be integrated in the answer building.

Some results of the project are the following. Eleven pairs of participants had to address 16 questions (176 MP3 files of 1m30 correspond to an answer). The experiment setup was in two phases: a phase of collaboration (1mn30) and then a phase of reading/comprehension phase, including analysis of eye movements and answers to questions. Here, we only discuss the first phase. Participants naturally followed a process in four steps, namely, reformulate the question for understanding, find an example to confirm their understanding, gather domain knowledge and elements of their working context, and build the answer either by looking for characteristics or by assembling explanation elements (for an integration).

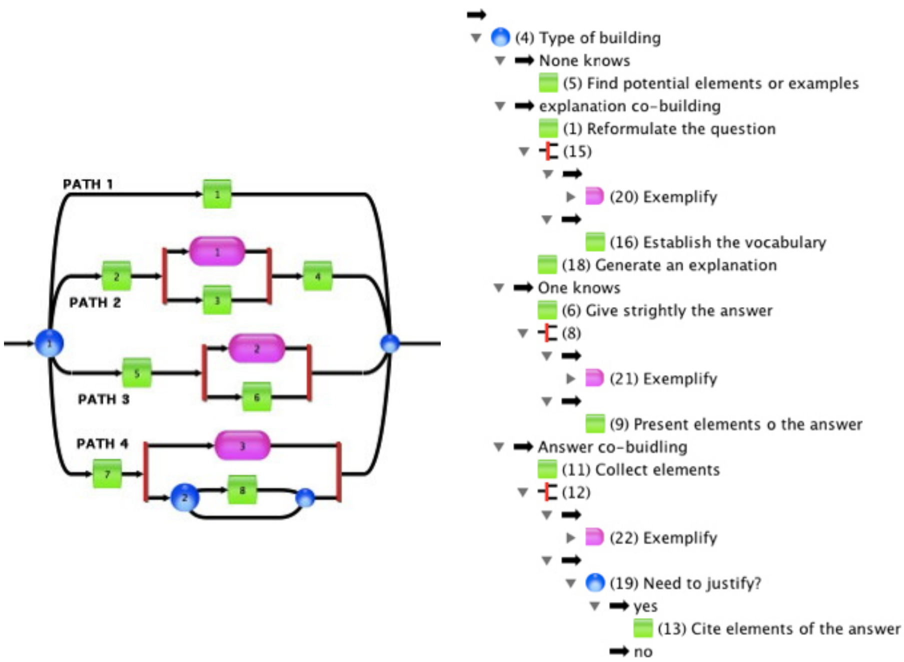

Figure 2.6. *Contextual graph of the identified collaborative answer building processes*

Figure 2.6 represents in the CxG formalism (see explanation section 2.3.4 for the components) the four types of the collaboration model found:

Path 1: No knowledge about the answer

Path 2: No answer but elements of explanation, Path 3: Two-way knowledge (Co-building of the answer),

Path 4: One-way knowledge (answer provided by only one participant).

Several observations could be made from the elicitation phase:

- Subjects had a problem for finding the right granularity of their answer. The answer can be given at different levels of granularity, and can be at the right level as well as at a too fine granularity (too many details) or too coarse granularity (rough description of the answer), such "gas" instead of "CO_2".
- Subjects often immediately repeat the question to be sure they understand correctly it, i.e. to be able to find some relationships between elements of the questions and elements of answer from their personal working contexts.
- Subjects first gather contextual information, not to immediately build the answer but to determine the right granularity of their answer, and second assemble the instantiated contextual elements to use in the proceduralized context. When subjects cannot identify the right level of granularity, they enter into a process of explanation to justify their proceduralized context.
- Subjects can know the answer without knowing elements of the answer. As a consequence, subjects express an external and superficial viewpoint.
- Collaboration is a minimal expression of cooperation: one subject leads the interaction and the other subject only feeds in information (or only agrees), reinforcing the statements of the other.
- Subjects provide explanations spontaneously to: (1) strengthen a known answer, (2) make progress in the co-construction of the answer; and (3) be sure of the granularity of the answer. The explanation is generally less precise than the answer (i.e. generally at a too general level), but useful between them to reinforce their ties.
- Several groups of subjects were confused and explained instead of giving the answer (thus with additional details not necessary).

The shared context contains all contextual knowledge discussed and accepted by all the actors during this process of shared context building. Then, actors build the proceduralized context. Once the focus is addressed, the proceduralized context enters actors' mental models. Later, this proceduralized context may be recalled as a whole for another task realization. Thus, the more an actor experiments, the more the actor acquires structured knowledge, and the more actors work together, the more they reinforce their ties too.

2.3.4. Implementation of the Contextual Graphs formalism

Based on the notion of contextual element, we have developed a formalism of representation called Contextual Graphs (CxG, Brézillon *et al.*, 2000). This formalism relies on four components, namely actions, contextual elements, activities, and ESIA (Executive Structure of Independent Activities), and has been implemented as a CxG software. The Figure 2.6 shows an example of use of the CxG software. Actions are green square boxes, contextual elements are represented by a contextual node (blue circle with a number) and a recombination node (blue circle without number) where end the paths coming from the contextual nodes, activities (contextual subgraph identified by actors as independent of the main graph) are represented by pink ovals. An ESIA is represented by two vertical red bars with independent activities between them. The independent activities can be treated in any order, but all will be executed before to resume the traversal of the main graph. This allows to reduce the size of contextual graph, otherwise it would be necessary to have twice the same subtree after a branch activity1-activity2 and another branch activity2-activity1.

A contextual element in the CxG formalism allows the management of alternatives on specific parts of practices according to their specific instantiations (i.e. the values at the focus at hand). A contextual element is represented by a pair {contextual node, recombination node}, and the legend is a question about its instantiation (e.g. “Temperature?”). Between the contextual and recombination nodes, there are as many exclusive branches as known practices (and instantiations, e.g., “warm” and “cold”). A value (quantitative or qualitative) of the contextual element corresponds to a branch between the two nodes with an action or an activity or an ESIA specific of the decision making on this instantiated contextual element. When the inference engine arrives at a contextual node, it needs to instantiate the contextual element at one of its values (known initially or asked to the user) to select the corresponding branch and progresses in the development of the practice. At the recombination node, the instantiation does not matter anymore, since the corresponding action has been completed. The sequence of instantiations during the practice development corresponds to a context-specific modeling of the practice (the proceduralized context), while contextual elements belong to the model of the task realization (i.e. the contextual graph).

Series-parallel graphs are used in a wide range of applications, the most common being electrical circuits (Eppstein, 1992) and scheduling problems (Finta *et al.,* 1996). In these domains, flows follow all the branches simultaneously while in a contextual graph only one path at a time is followed because only one instantiation can be activated for a contextual element. Moreover, each total path in the graph corresponds to a unique practice developed for the task realization. As presented in section 2.2.3., a contextual graph has a parallel structure for the tactical expert who wants to “keep an open mind”, and has a series structure for the operational expert who prefers to focus on the details of the task realization without a holistic view.

2.4. Using context in real-world applications

2.4.1 Context and focus processing

Enterprises develop *procedures* to address a focus in any situation, but generally procedures result in sub-optimal solutions for any specific focus. As a consequence, actors develop *practices* that contextualize the procedure for addressing the specific contexts where is the focus (Brézillon, 2005b, 2006). In some way, they elaborate a *contextualized task model* in the spirit of what is shown Figure 2.7. At the time expert systems (see Turner and Brézillon, 2022, this book), the knowledge engineer elicited knowledge from experts by asking a question who answered by specifying "In the specific context C1, I will apply solution S1", but the knowledge engineer only retained (problem, solution S1) whatever the context was.

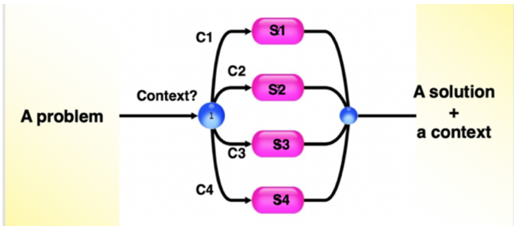

Figure 2.7. *Contextualization of a procedure by operators*

Formally, consider a practical unit of reasoning "IF P1, P2, … THEN C1". In a logical and theoretical reasoning (a procedure), the action leads to the conclusion C1 that is added alone to the knowledge base. In a practical reasoning (a practice), the premises stay attached to the conclusion, that is, C1 has a meaning coming from premises. A contextual element like C1 is defined in a context of use (P1, P2, … are true) that is lost in the procedure that aims to be applicable to a large class of similar problems, whereas a practice is a contextualized version of the procedure for the problem in a specific context. Such a practice generally is the "best practice" in that specific context, and there are as many best practices as possible contexts.

If a procedure is a representation of a task model, the modeling of practices would lead to a large number of contextualized task models because each context of the task realization generally is unique (Brézillon, 2005b). The Contextual Graphs representation formalism allows a global modeling of all the practices developed by actors in one contextual graph. The contextual graph represents the *contextualized task model* where all the branches correspond to intertwined practices developed by actors. The practice are paths from one input to one output in the contextual graph and often a practice differs from other practices (i.e. other paths) generally on small parts, for example by different instantiations of a contextual element (and thus different actions between the contextual node and the recombination node). Several practices use same parts of the paths too. This contextualization of the procedure is at the tactical decisional level, while the development of a specific practice is at the operational decisional level (see Brézillon *et al.*, 2007; Brézillon and Brézillon, 2008, for an application concerning the self-training of car drivers).

2.4.2. Context and actors

A task model often comes from a decontextualization process of a large set of practices to extract what is common to a maximum of practices. For example, in the SART project (Brézillon *et al.,* 1997), procedures are established in a kind of CDR process by analyzing incident-solving reports (identification of the contextualization in the practices) that are formalized (i.e. decontextualized) by merging all reports on a given incident in a procedure applicable to the largest class possible of incident-solving, procedures that will be recontextualized by train drivers when needed. Procedures are exceptionally applicable directly to “real-world” incidents. In their daily work, actors have to apply the procedure in specific contexts that require adaptation of the task model (i.e. recontextualization). As a consequence, the actor uses knowledge and experience to interpret and decide how to (re)contextualize the procedure during their task realization in the specific context. Actors have mental models of the type of incident-solving that is generally used for this type of incident in the context at hand as operational knowledge. We observed this in the SART and TACTIC projects (Brézillon *et al.*, 1997; Kabil *et al.*, 2015).

An activity is described in a script that runs in a specific context. The activity is the application of an operator’s strategy at the tactical level, and the practice is the description of this activity at the operational level. The notion of “activity” extends the notion of “task realization” by including the concrete implementation of a task that is realized by an actor in a specific situation with the resources at hand. The task is what an actor must do, and the activity is the (physical and mental) behavior that the actor exhibits during task realization (Sarrazin *et al.*, 1996). Associated with each activity, there are one or more scripts that describe the ways an activity must be executed. The context determines the script to be run for an activity, and concerns the result as well as the process leading to the result.

If the task realization corresponds to a personal viewpoint, the notion of activity is more adapted to a collective viewpoint. The two notions (task realization and activity) coincide if only one actor is involved. In collaborative work, several actors work together as group members. This requires an external viewpoint for “sharing out” the task realization between group members. The group activity is related to the way in which the group as a whole realizes the task. Benitez-Guerrero *et al.* (2012) present the development of an activity (called effective activity) as an instance of an activity model that describes the group members that can participate in the activity, how the activity can be carried out, the family of objects that can be manipulated or produced, and which roles group members and objects will play in the activity (what we call context of the activity). A group activity is more than the simple sum of actor activities because these activities are interdependent. Each group member generally intervenes at different moments during the development of the group practice. Each intervention of a group member can be represented as an independent elementary task, and the interaction between members during the development of the group activity seems to be better understood as a cycle of realization of independent tasks taken in the activities of the group members. This has led us to extend the CxG formalism.

2.4.3. Extension of the CxG formalism

The modeling of group activity requires the extension of the CxG formalism, first for managing several actors, and, second, for cyclic use of a contextual graph in order to describe the development of the group activity as a sequence of independent subtasks. As said previously, this supposes too the management of the shared context with the addition of particular contextual elements for modeling the management of group members' interaction (turn, acceptance, etc.) during group-activity development (Brézillon., 2006). The shared context becomes a crucial place for interaction management as well as task management in each actor's activity. When a subtask is realized, the shared context is analyzed at the end of the turn for determining which actor must intervene next and which independent subtask the actor must realize. We obtain a sequence of turns generated during the development of the group practice.

Because a contextual graph is directed and acyclic, its traversal corresponds to a unique subtask realization, and a simulation corresponds to a sequence of turns (i.e. successive traversals of the contextual graph). The management of the turn sequence is obtained, first, by changes made in the shared context during the previous turn, second, by the execution of the actions in the subtask, and, three, by the instantiation of some "particular" contextual elements defining (1) the actor (particular contextual element SENDER) who was the previous manager, (2) the actor responsible for the subtask in progress (particular contextual elements MANAGER of the turn), and (3) the next manager (particular contextual element TASK_GOAL and RECIPIENT respectively) and its subtask concerned. The simulation of a sequence of turns requires a cyclic use of the contextual graph as long as the shared context is modified during the turn. The group leadership during interaction is assured by the actor responsible of the turn, that is, for the duration of a turn (Brézillon, 2017).

The modeling of the interaction between group members specify turns and the association of each turn with a subtask of one group member. Each turn is associated with a use of the contextual graph for realizing the subtask of an actor, and the sequence of turns corresponds to a cyclic use of the contextual graph for realizing the task by the group (i.e. after several turns until the shared context stops being modified). The main point here is a modeling of the task realization in terms of *independent elementary subtasks* for each group member. Concretely, the group management is represented by a general structure, called a contextual meta-graph, and a turn consists in the selection of (1) the group member that will be the manager of the turn, (2) the independent elementary subtask realized by the manager during the turn and (3) the selection of the next manager and of the concerned subtask. The cyclic use of the contextual meta-graph opens the possibility of backtracking into the group reasoning as well as the possibility of cycles for negotiation between two (or more) group members.

Figure 2.9 shows the organization of the contextual meta-graph in two windows, which is obtained from the CxG software. The window on the right is the graphical representation of the meta-graph with four contextual elements (blue circles, the contextual nodes with a number and the combination nodes without number) and six activities (pink ovals containing a similar contextual subgraph with a similar structure as shown in Figure 2.9 where each branch corresponds to an independent subtask). The window on the left presents the legends of all the items on the graphical part.

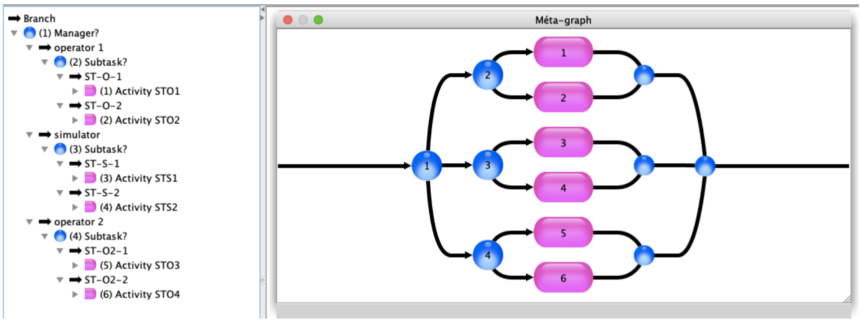

Figure 2.9. Example of conceptual meta-graph

The explicit consideration of the shared context with the contextual meta-graph opens the door to more options than before like:

- The simulation can now be stopped at the end of any turn (with an action “RECIPIENT = <nil>”).
- The use of the simulation parameters introduces in a natural way the notion of loops for managing negotiation among actors and realizing a given subtask (or sequence of subtasks) in different contexts, thanks to the association of the contextual graph and the shared context.
- The same goal can be assigned to several actors (e.g. several reviewers for a paper submission).
- An elementary subtask may have several possible outputs depending on a contextual element instantiated in the subtask. Outputs may concern the actor or a subset of actors.
- An actor may change the instantiation of a contextual element used in the task of another actor (or several other actors).

The modeling of a group-activity development, first, is just a natural extension of the practice development by one actor, and, second, the results obtained by the group-activity modeling give a new insight into modeling the task realization by an actor as an activity, with the possibility to also have a “nonlinear” representation of the actor activity, for example, checking first different alternatives by simulation (cyclic use of the contextual meta-graph for one actor) before making decision. The shared context plays a more important role than the working context because the shared context is built collectively to realize the group activity.

In this paper, we just want to discuss the interests of the extension of the CxG formalism:

- A contextual meta-graph captures the accumulated experience of one or several actors that collaboratively realize a task, as said initially for one actor.

- A contextual meta-graph stays a directed acyclic graph with one input and one output like in the initial version of the CxG formalism, but with automation and control of the cyclic use of the contextual meta-graph, thanks to a full use of the shared context.

- Traversing the meta-graph corresponds to entering in an actor’s activity and realizing the specific independent subtask specified for the turn.

- The development of the group activity corresponds to a sequence of turns taken in different members’ activities, a turn corresponding to the realization of an independent subtask and the management of particular contextual elements (e.g. MANAGER) controlling, on the one hand, transitions between turns and, in the other hand, the task status of the group practice development (identification of the next manager or the exit of the cycle).

- The particular contextual element MANAGER is the actor responsible for the current turn. MANAGER is instantiated for the next turn by an action during the subtask realization of SENDER.

- The particular contextual element TASK_GOAL monitors the independent subtasks to realize during the turn in the manager activity. TASK_GOAL was instantiated by an action during the subtask realization of SENDER during the previous turn.

- The shared context is composed of the contextual elements of the group activity realization and the contextual elements specific to each member’s activity, information transfer between members’ activities, and interaction management of members during the group-activity development.

The introduction of the turn mechanism leads to a new type of simulation, namely the *CxG-based simulation* (Garcia and Brézillon, 2015). The CxG-based simulation allows development of complex reasoning based on the dynamical combination of a limited number of independent elementary tasks, which does not require complex modeling, because the model of a group activity is built dynamically depending on the instantiation of the contextual elements at the beginning of a cycle. Moreover, the subtasks can be reused in a number of different combinations and several times with a unique representation at the implementation level. A contextual graph is similar to an *experience base* that contains a lattice of practices developed for realizing a given task. The complexity of the interactions is represented by the interplay of the particular contextual elements.

A CxG-based simulator presents a number of advantages that a model-based simulator does not present. If a model-based simulation is a top-down model, a CxG-based simulation corresponds to a bottom-up model. In a model-based simulation, the whole working context is defined at the start of the simulation and stays constant during the simulation, while in a CxG-based simulation, the shared context evolves during practice development. A formal model is given initially and is compared to observations, while a practice is built progressively by the CxG simulator. In that sense, a CxG-based simulation is a particular type of simulation. The behavior of a CxG simulator is similar with the usual model-based simulator's behavior, supposing that (1) contextual elements in the contextual graph can be compared to the parameters in the formal model (a change of parameter values impacts the model behavior as a change of instantiation modifies the practice developed), and (2) variables in a model-based simulation are related to the result of the progressive building of the practice corresponding to the working context.

2.5. Conclusion

This paper sums up 25 years of research on how to model and use context in real-world applications, based on about 20 applications on a spectrum from technology-centered (SEPT, SART) to human-centered (medicine) applications, i.e. from well-defined domains to more fuzzy ones. Real-world applications impose strong requirements and imply the rejection of models based on simplifying assumptions: the operator in a crisis room and the surgeon in an emergency room have to rapidly make a decision that will be definitive. Thus, the answer "Well, in 80% of the cases, the solution is…" is not an option.

On the theoretical side, McCarthy (1993) pointed out the infinite dimension of context that would appear to be a severe limit for any modeling of context. However, we have shown that if context has an infinite dimension, all the context elements are not of interest for a given focus. This was made explicit by distinguishing contextual knowledge and external knowledge with respect to the focus (lesson learned 1).

Context is strongly associated with knowledge and reasoning about a real-world focus. In some sense, context has multiple aspects like knowledge but also like a process (reasoning in decision making). Context intervenes more on the nature of the knowledge than on the knowledge itself: knowledge is contextual or not depending on the focus. However, it is not sufficient to just say that for modeling and using context.

Context cannot be defined in an abstract way (lesson learned 2): it is dependent on a focus (task realization, problem solving, decision-making, etc.). The contextual knowledge (the part of the context linked to the focus) is represented as contextual elements (lesson learned 3). A context-based representation formalism called Contextual Graphs was developed and has been implemented as a software (lesson learned 4). A contextual graph gives an inclusive representation of all the practices developed by actors for a given task realization. The nodes of a contextual graph (blue circles) correspond to contextual elements as pairs of a contextual node and a recombination node, with exclusive branches between the two nodes representing parts of the practices that actors have developed by considering different instantiations of the contextual element (i.e. in different contexts). The lesson learned 5 is that the contextual element must be differentiated from its instantiation.

The different paths from the input to the output of the contextual graph represent all the practices developed by actors for the task realization. A contextual graph is a contextualized task model (lesson learned 6). The set of contextual elements in a contextual graph corresponds to the working context of the task realization, and the ordered set of contextual elements that are instantiated during the development of a practice is the proceduralized context that is used in the focus (lesson learned 7).

The extension of the CxG representation formalism combines what was done initially (i.e. modeling an actor activity) with four new "particular" contextual elements for (mainly) defining which group member is the manager, which independent elementary subtask must be realized by the manager during the turn, and who will be the next manager. A group-activity development is simulated by the cyclic transversal of the contextual graph in different shared contexts (lesson learned 9). Beyond the simple simulation of the group-activity development, it is now possible to backtrack in the group reasoning, to simulate alternatives, etc.

A contextual graph contains an organization of all the known practices for a task realization, which makes it an experience base associated with the focus (lesson learned 8). This practice structure is organized by contextual elements that provides a coherent global picture of the group activity in various contexts (lesson learned 9).

Although the CxG formalism has been applied successfully in several fields (e.g. medicine, transportation, business management and military operations), there are some limitations. However, users have difficulty following the development of a specific practice in the contextual graph and to make the decision that is associated with it. Often, experts need to quickly identify the right practice that matches the specific context at hand (especially in emergency situations), first, to evaluate a situation in order to make immediate decision (e.g. handle an incident in the subway), and second, to infer the type of object of the reasoning corresponding to the content of a given practice. An object involved with reasoning can be either a physical object (e.g. a digital image in medicine) or a cognitive object (e.g. a diagnosis). A version of the final results as a pseudo-tree representation has been developed for this purpose for allowing a fast understandability of the focus by actors with an explicit proceduralized context and of the sequence of actions to do (Garcia and Brézillon, 2018) (lesson learned 10).

The revised version of the CxG formalism developed for group activity stays applicable to a unique actor with the interests of:

- A nonlinear exploitation of a task realization or activity, while the task model stays "linear";
- Integration of the environment state. The actor may change his/her mind during practice development when an unexpected event occurs (e.g. the actor selects an object and later discovers that the object is not adapted to the objective or not available),
- Simulation. The actor can replay several times a particular subtask for checking different hypothesis (i.e. different contexts) before making decision,
- Explore a decision tree. The actor can backtrack in his/her reasoning.

There are other potentialities to explore. First, a system using experience bases is able to explain, anticipate by simulation, complete incrementally its knowledge when failing and alert about abandoned solutions for the practice. New practices are elicited from actors when new contexts are encountered, and thus the number of practices may become very large. The capabilities of incremental knowledge acquisition, practice learning and explanation allow a system exploiting contextual graphs to become increasingly "intelligent" because it can benefit from the situation to learn incrementally new practices when it fails. In that sense, a new generation of AI systems to be developed would be context-based intelligent assistant systems working on local experience bases corresponding to the realization of different tasks.

Second, it is possible to add in an experience base practices with failures as well as successes in the realization of the task. This could be interesting for training future actors by showing the potential danger if a contextual element is not instantiated correctly. In the SART project, the operator starting his work may replay (alone or with the previous operator) the solving of an incident to check if there is an alternative more efficient.

Third, the actors' behavior realizing the same task (in parallel or sequentially) can be studied and compared by simulation of their task realizations to determine, say, those that develop safe or risky practices (and also for statistics). This could also be used as a tool for hiring an incoming actor based on simulation of real-world conditions.

Fourth, a contextual graph, being an experience base corresponding to the realization of one task, makes it possible to sum up the experience bases obtained of different task realizations and to compare them in order to reuse the lessons learned in one experience base in another one.

2.6. References

Agabra, J., Alvarez, I., Brézillon, P. (1997). Contextual knowledge-based system: A study and design in enology. *First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*. Rio de Janeiro, Brazil, Federal University of Rio de Janeiro (Ed.), 351-362.

Benitez-Guerrero, E., Mezura-Godoy, C., Montané-Jimenez, L.G. (2012). Context-aware mobile collaborative systems: Conceptual modeling and case study. *Sensors*, *12*, 13491-13507.

Bau, D.Y., Brézillon, P. (1992). Model-based diagnosis of power station control systems: the SEPT experiment. IEEE Expert, pp. 36-44.

Brannen, M.Y. (2004). When Mickey loses face: recontextualization, semantic fit, and the semiotics of foreignness. *Academy of Management Review*, 29, 4, 593–616.

Brézillon, P. (2005b). Task-realization models in Contextual Graphs. *Modeling and Using Context (CONTEXT-05)*. A. Dey, B.Kokinov, D.Leake, R.Turner (Eds.), Springer Verlag, LNCS 3554, 55-68.

Brézillon, P. (2006). Some characteristics of context. In: M. Ali & R. Dapoigny: *Proc. Of IEA/AIE 2006*, Springer Verlag, LNAI 4031, 146-154.

Brézillon, P. (2017). Contextual modeling of group activity. Springer International Publishing AG 2017, P. Brézillon *et al*. (Eds.): *CONTEXT 2017*, LNAI 10257, 113-126**.**

Brézillon, P. (2018). CxG-based simulation of Group Activity. *ISTE OpenScience* – Published by ISTE Ltd. London, UK – openscience.fr, Vol. 2-1, DOI: 10.21494/ISTE.OP.2018.0231

Brézillon, P., Attieh, E., Capron, F. (2014). Modeling glocal search in a decision-making process. In: Gloria PhillipsWren, Sven Carlsson, Ana Respício, Patrick Brézillon (Eds.): *DSS 2.0 – Supporting decision making process with new technologies*. Frontiers in Artificial Intelligence and Applications, 261, IOS Press, pp. 80-91. DOI: 10.3233/978161499399580

Brézillon, J., Brézillon, P., Tijus, Ch. (2007). Improving driver's situation awareness. In: *Modeling and Using Context (CONTEXT-07)*. LNAI 4635, Springer Verlag, 136-149.

Brézillon, P., Brézillon, J. (2008). Contextualized task modeling. *Revue d'Intelligence Artificielle*, 22(5), 531-548.

Brézillon, P., Drai-Zerbib, V., Baccino, T., Therouanne, T. (2006). Modeling collaborative construction of an answer by contextual graphs. *Proceedings of IPMU*, Paris, France, May 11-13.

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

Brézillon, P., Pasquier, L., Pomerol, J.-Ch. (2000). Representing operational knowledge by contextual graphs. *Advances in Artificial Intelligence*, M.C. Monard and J.S. Sichman (Eds.). Berlin: Springer. Lecture Notes in Artificial Intelligence, N° 1952, 245-258.

Brézillon, P., Pomerol, J.-Ch. (1999). Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3), 223-246.

Brézillon, P., Pomerol, J.-Ch. (2001). Is context a kind of collective tacit knowledge? *European CSCW 2001 Workshop on Managing Tacit Knowledge*. Bonn, Germany, M. Jacovi and A. Ribak (Eds.), 23-29.

Clancey, W.J. (2002). Simulating activities: relating motives, deliberation, and attentive coordination. *Cognitive System Research, 3(3)*, 471–499.

Eppstein, D. (1992). Parallel recognition of series-parallel graphs. *Information and Computation*, 98(1), 41-55.

Fan, X., Zhang, R., Li, L., Brézillon, P. (2011). A context-based framework for improving decision making in scientific workflow. *Proceedings of the 3rd International Conference on Computer Research and Development (ICCRD-2011)*, Shanghai, China.

Finta, L., Liu, Z., Mills, I., Bampis, E. (1996). Scheduling UET- UCT series-parallel graphs on two processors. *Theoretical Computer Science*, 162(2), 323-346.

Furnas, G.W. (1986). Generalized fisheye views. *Proceedings of Human Factors in Computing Systems CHI'86 Conference,* 16-23.

Garcia, K. and Brézillon, P. (2015). A contextual model of turns for group work. In: Christiansen H., Stojanovic, I. & Papadopoulos G.A. (Eds.): *Modeling and Using Context (CONTEXT-15)*, Springer, LNAI 9405, 243-256.

Garcia, K., Brézillon, P. (2018). Model visualization: Combining context-based graph and tree representations. *Expert Systems With Applications*, 99(1): 103-114.

Grimshaw, D.J., Mott, P.L., Roberts, S.A. (2000). The role of context in decision making: some implications for database design, *EJIS Conference*.

Kabil, A., Brézillon, P., Kubickki, S. (2015). Contextual interface for operator-simulator interaction. In: Christiansen H., Stojanovic, I. & Papadopoulos G.A. (Eds.): *Modeling and Using Context (CONTEXT-15)*, Springer, LNAI 9405, 483-488.

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

Landauer, C., 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.

McCarthy, J. (1993). Notes on formalizing context. *IJCAI*, 555–560.

Miller, G. A. (1956). The magic number seven plus or minus two: Some limits on our capacity for processing information. *Psychological review*, *63*, 91-97.

Pomerol, J.Ch., Brézillon, P. (1999). Dynamics between contextual knowledge and proceduralized context. *Modeling and Using Context (CONTEXT-99)*. In: Lecture Notes in Artificial Intelligence, N° 1688, Springer Verlag, 284-295.

Pook, S., Lecolinet, E., Vaysseix, G., Barillot, E., (2000). Context and interaction in zoomable user interfaces. *AVI 2000 Conference Proceedings (ACM Press),* 227-231.

Sarrazin, P., Biddle, S., Famose, J.-P., Cury, F., Fox, K., Durand, M. (1996). Goal orientations and conceptions of the nature of sport ability in children: A social cognitive approach. *British Journal of Social Psychology, 35*, 399-414.

Schank, R.C. (1982). *Dynamic memory, a theory of learning in computers and people.* Cambridge University Press.

Sowa, J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations* Pacific Grove, CA: Brooks Cole Publishing Co.