

Context in problem solving: a survey

PATRICK BRÉZILLON

LIP6, Case 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France. Email: Patrick.Brezillon@lip6.fr

Abstract

Dealing with context seems to be a major challenge which the field of Artificial Intelligence (AI) will have to face in the next few years. The importance of the challenge is borne out by the various scientific events focusing on context which have been held since 1995. This does not mean that context has not been studied before—it has in such domains as Natural Language Processing—but it has yet to be studied in depth since previous research has only covered few aspects of the problem. We present in this paper a survey of the literature dealing directly and explicitly with context whatever the domain. This makes it possible for us to have a clear view of context in AI. One of the conclusions of this survey is to point out the existence of different types of context in areas such as the representation of knowledge in a computer system, the reasoning that the system carries out using the knowledge, and the interaction the system has with people.

1 Introduction

Context has played an important role in a number of domains for a long time. This is especially true for activities such as predicting context changes, explaining unanticipated events and helping to handle them, and focusing attention. (See Jansen (1995) for philosophical considerations on context.) Context also is widely used. When looking for the term “context” using AltaVista on the Web, we found more than 440,000 documents containing the term, while a similar search for the term “concept” turned up only 400,000 documents. The main problem is the lack of a consensus about the meaning of “context”. It has been used to refer to a set of preferences and/or beliefs (Cahour & Karsenty, 1993), a window on a screen (Abu-Hakima, 1993), an infinite and only partially known collection of assumptions (McCarthy, 1993), a list of attributes, the product of an interpretation, and a collection of context schemata (Turner, 1993), paths in information retrieval (Boy, 1991b), slots in object-oriented languages, buttons which are functional, customizable and shareable (Maskery & Meads, 1992), possible worlds (Mylopoulos et al., 1990), assumptions under which a statement is true or false (Cavalcanti, 1993), a special, buffer-like data structure, or an interpreter which controls the system’s activity (Barr & Feigenbaum, 1981), the characteristics of the situation and the goals of the knowledge use (Bastien, 1992), entities (things or events) related in a certain way (Ogden & Richards, 1946), and the possibility that makes it possible to listen what is said and what is not said (Winograd & Flores, 1989).

Currently, one notes an increased interest in context. Since 1993, several scientific events in artificial intelligence have dealt specifically with context, and a number of works consider context explicitly in areas which stretch from modelling domain knowledge to programming languages. The modeling, representation and use of context appear to be a major challenge for the next few years, especially since we now must deal with very complex problems, large knowledge bases and multimedia applications.

In the following, we present a survey of the literature to determine a definition of context. However, our interest is in the modeling and the use of context in real-world applications. The context of our survey on context is cooperative problem solving by a human and a machine. In this

broad framework, we study more specifically the link between context, explanation and incremental knowledge acquisition. Thus, our survey of the literature on context has been made according to this point of view and so may be a biased view of a literature which is very large and covers a number of different domains. The survey has been made with the following definition of context in mind:

“Context is what constrains problem solving without intervening in it explicitly.”

Hereafter, the paper is organized in the following way. Section 2 presents the way in which context is perceived in different domains related to Artificial Intelligence (AI) such as natural language processing, databases, communication, electronic documentation and vision. The section ends with a presentation of the Pepys project which uses physical context as a cue for recall. Section 3 discusses context in AI. We present first the lack of consideration of context as one reason for the failures of the first generation of knowledge-based systems. We then present the use of context in AI, with an emphasis on explanation, knowledge acquisition and machine learning. The section ends with the presentation of two projects, namely the CYC project and the ORCA project. Section 4 describes how context is represented in logics, rule-based formalisms and other formalisms. We then propose a synthetic view of these approaches. The section ends with an example of context representation in database interrogation. Section 5 gives some clues about the way context relates to other elements. This section is the result of discussions at different scientific events in 1993, 1995 and 1997 all of which focussed on context. The general discussion in section 6 closes the paper.

2 Context in domains related to AI

In this section, we discuss the way in which context is studied in five domains related to AI. We end the section with the presentation of a real-world application.

2.1 Context in natural language processing

The first work which explicitly considers context is in the area of Natural Language. For instance:

1. Frege identifies three different contexts, ordinary, direct and indirect, in which names can be used. In an ordinary context, names have their customary denotation and sense. The direct context is what is now known as the use-mention distinction: words name (denote) other words. (Quotation marks or italics are used for direct contexts in writing.) In an indirect context, names denote their customary denotation and have an indirect sense that is different from their customary sense (Frege, 1985).
2. Russell introduces contextual descriptions that have no meaning of their own. However every sentence in which they occur has a meaning (Arbab, 1992).
3. The idea that an entity representation/knowledge about a subject is influenced by something internal to that entity has been around in philosophy for some time. Husserl used the term *Noemata* instead of context (Guha, 1991).
4. Cheng and Holyoak present pragmatic schemas of reasoning that are induced from experience and evoked by the context (Politzer, 1991).

For his Discourse Representation Theory, Kamp developed a version of logic with explicit contexts to map sentences from natural language in a way that would preserve their context structure (Kamp, 1981). For reasoning about context, Pierce defined a complete set of inference rules based on primitive operations of copying and erasing graphs in various contexts (Sowa, 1992). Pierce's rules of inference for reasoning with graphs are explicitly formulated in terms of contexts and the conditions for importing and exporting information from contexts.

In conceptual structures (Sowa, 1984), a context was defined as a collection of propositions that were not distinguished from the statements and situation they describe in the real world. More recent formulations of conceptual graph theory (Sowa, 1991) make a distinction between a

proposition as an abstract entity, its statement in some language or system of logic, and its use in describing a situation (Sowa, 1992).

Moore (1995) states the explanation problem as follows: given a communicative goal, find information from the expert system's knowledge sources that is relevant for achieving this goal, and organize this knowledge into a coherent multisentential text. To allow the explanation to be responsive to the user's feedback and sensitive to previous explanations, the system must be able to reason about its own previous responses, for example, to interpret the user's follow-up questions in the context of the ongoing conversation and to determine how to elaborate or clarify a response when necessary. New information must be related to what the information seeker already knows to provide missing information in a way that facilitates understanding and learning.

Moore's approach employs rhetorical strategies as compiled knowledge about what actions may be used to satisfy certain intentions. Explanation strategies thus enable systems to generate a range of different explanations from the same knowledge representation (Carenini & Moore, 1993). Information about focus is inferred from the dialogue history, using two pointers into its structure: *global-context* and *local-context*. *Global-context* indicates the current topic under discussion. It points to the place in the dialogue history where this topic was begun. *Local-context* points to the most recent thing that was said. It always points to the node in the text plan representing the last clause uttered by the system.

Still in the area of Natural Language Generation, Pereira and Pollack (1991) present a system, called *Candide*, to incrementally interpret natural-language utterances in context. Context-independent and context-dependent aspects of an interpretation are separated. The context-independent part of the information that is invariant with respect to further incremental interpretation is separated from the context-dependent part that may vary. The authors' discussion is organized around three types of context, namely, the immediate context, the local context, and the global context.

More recently, one of the main conclusions of the IJCAI-95 Workshop on Context in Natural Language Processing (NLP) is that context in NLP is still largely unexplored (Iwanska, 1995). This has serious consequences. First, there is a theory-versus-practice gap. Secondly, context appears to be a huge bag of issues, virtually always closely related to the specific task, domain, application, and research problems of immediate interest to authors. Thirdly, work addressing the problem of context is often too vague and does not address specific questions about context. Finally, one should not even attempt to unify the various notions of context as long as a consensus is not reached. However, there is agreement about the fact that context provides constraints on reasoning, increases information content of natural language utterances, and facilitates learning.

2.2 Context in databases and ontologies

The main role of context is to provide humans with a much greater control over knowledge. Context makes it possible to define what knowledge should be considered, what are its conditions of activation and limits of validity and when to use it at a given time (Bastien, 1992). This is especially important for the building and the use of large and reliable knowledge systems. Contexts act like adjustable filters for giving the right meaning in the current context and to present the minimal number of information pieces and essential functions that are necessary to the task at hand (Barthe, 1991). For instance, the concept of water is viewed differently by a thirsty person, a plumber, a chemist, and a painter. However, one individual could be any one of these at different times.

As an alternative to the integration approaches in the literature, Goh et al. (1995) propose a strategy based on the notion of context interchange in databases. The Context Interchange strategy is an approach for achieving interoperability among heterogeneous and autonomous data sources and receivers. Context refers to the (implicit) assumptions underlying the way in which an interoperating agent routinely represents or interprets data. Data contexts, as event scripts, are abstraction mechanisms that allow us to cope with the complexities of life. In the context interchange framework, assumptions underlying the interpretations attributed to data are explicitly

represented in the form of data contexts with respect to a shared ontology that reduces the cost of communication among members of a group and constitutes a shared vocabulary for context definition. The approach makes it possible to distinguish the source (export) and receiver (import) contexts. A context mediator is used to compare the source and receiver contexts and detect any conflicts. The export context captures those assumptions integral to the “production” of data in the data source. The import context captures those assumptions which the data receiver will employ in interpreting the data.

Sciore, Siegel and Rosenthal (1992) have proposed an extension to SQL, called Context-SQL (C-SQL) which allows the receivers’ import context to be dynamically instantiated in an SQL-like query. “Context-SQL provides an excellent vehicle for users who are interested in modifying their import contexts dynamically as queries are formulated.”

Walther et al. (1992) use the PROTEGE-II system as a metatool for constructing task-specific expert-system shells. The system associates each method with an ontology that defines the context of that method. All external interactions between the method and the world during the method assembly are a mapping of knowledge between the method’s context ontology and the ontologies of the methods with which it is interacting. Each ontology contains all necessary information for defining the role of the module in the module-assembly process, and thus places a number of further requirements on the representation language. This association is described in a context-definition language called MODEL. As a result, it is shown that shareable ontologies are a fundamental precondition for reusing knowledge, serving as means for integrating problem-solving, domain-representation, and knowledge-acquisition modules. A shared context is referred to as an ontology because the domain ontology provides a common understanding of the involved design concepts and of the topological relations between them. The context in PROTEGE-II (1) captures the role of the component during assembly, (2) describes the knowledge required by the component, (3) specifies the input and output requirements of the component, and (4) encapsulates the component’s behavior so that the component can be reused and shared.

Along with research on implementation of context in databases, some products have appeared on the market. For example, the Oracle7 ConText Option is aimed at a challenge facing business—how to effectively manage the increasing volume of electronic texts, documents, manuals, e-mail and news (Oracle, 1996). The software builds on previous generations of Oracle text and database products to provide intelligent management of all corporate information. The product is supposed to ‘understand’ complex documents, summarizes them into Gists context-based document summaries and conveys their key themes without human intervention. It supports retrieval through standard SQL, treating text data as a peer to relational data. A variety of standard tools, ranging from Forms interfaces with sophisticated Web environments, can combine structured data searches with intelligent text searches and text reductions in a single query. The result is a familiar, unified environment for rapidly developing new text-enabled applications, or for easily performing ‘hot upgrades’ to text capabilities of existing systems. Another example in indexing is the CONVECTIS system developed by HNC Software Inc. (USA) for automatically indexing free text documents. The system uses a context vector representation of text, which encodes similarity of meaning at the word level. The context vector representation offers all the benefits of traditional vector space representations plus a number of additional advantages. Most important, unlike traditional vector representations, context vectors encode meaning relationships at the word level. As a result, documents can have similar vectors even though they have few words in common.

2.3 Context in communication

In a report on a workshop on context held at the 1989 Computer Human Interaction Conference (CHI’89), Maskery and Meads (1992) present the conclusion obtained by attendees in discussions limited to the context of a relationship between a human and a computer-based system. Context was considered as a property of the interactions among agents, as opposed to context as a fixed property of a particular problem or application domain. That is, without interacting agents, there would be

no context. In communication, the context is considered as the history of all that occurred over a period of time, the overall state of knowledge of the participating agents at a given moment, and the small set of things they are attending to at that particular moment. Context appears as a shared space of knowledge. Each entity involved in an interaction has its own context, which may or may not be consistent with parts of the contexts of other entities. In order to succeed as a collaborative partner, a system should provide ways for a user to express, explore, recognize and negotiate their shared context.

In the following workshop on context at CHI'90, Maskery et al. (1992) report a more practical view on context. Context can be thought of as a kind of expert system that would be expert in "predicting" what the user would likely want/need to do next because of its knowledge of what had happened to either that user or other users with the same goals/needs. Then, context can be provided through a well-elaborated user interface using currently known graphical techniques. The key concept behind this kind of context is individualizing the user interface according to the current task. For providing context-sensitive help, a system must be able to answer the user's question for help, to figure out what they want help on, relieve the user of having to formulate precise queries, to provide concise, pertinent information immediately, to help the user to define questions, and to anticipate users' needs for information. For this, such a system would need to know the history of interaction with system, the transaction history, the user's characteristic, the user's intention, the possible sources of ambiguity, the state of the system, and the user's profile and system access.

The main point underlined by Mittal and Paris (1995) is that communication (and mainly explanation as a part of it) and context constrain each other: context of the situation activates behavior potential, which in turn modifies the context of the situation. They bring together different notions of context as elements of a global picture that might be taken into account by an explanation module, depending on the needs of the application. They describe an implemented intention-based planning framework for explanation that can take into account two different aspects of context, namely the participants and the discourse, although they find that there are other aspects of the problem solving situation. For instance, the participants involved, the mode of interaction in which communication is occurring, the discourse taking place, and the external world all have a bearing. Indeed, many systems have aspects of context already represented in appropriate ways (e.g., a domain model, a description of tasks and methods, an execution trace, etc.) to be used by their components.

Producing/interpreting a message is done in one context, which most of the time is assumed to be shared by all the participants. For Cahour and Karsenty (1993), five types of components are essential in defining the context of the dialogue: the dialogue memory, the task memory, the environmental situation, the psycho-social situation, and the general knowledge about the world. These components are like "knowledge bases". Because we do not share exactly the same knowledge bases, and consequently the same activated context, every participant has his/her own vision of the shared context. A failure in communication happens as a result of the illusion of sharing a context that creates misunderstandings.

Karsenty (1994) argues that communication implies the sharing of linguistic code and a context. The context is a set of information pieces that are accessed or built to give a meaning at a message. Thus, explanation is a means to share the context that is needed for a full understanding. Its aim is to differentiate between an initial context, in which an information is not understood or misunderstood, and a target-context, in which the information becomes comprehensible. It is a way to make explicit the implicit knowledge in a procedure and interpret a new information. Explanation generation acts as a contextualization process (Edmondson & Meech, 1993; Karsenty & Brezillon, 1995), and manages the interaction context. Explanations are a type of validation of the context. Conversely, the explicit use of context permits explanations to be tailored to a specific request. It is the context that supplies any explanation needed to validate suggestions (Karsenty & Falzon, 1992).

Making context explicit would also make it possible to revise some well-known paradigms in AI. For instance, Reichman (1985) proposes restating two of Grice's conversational maxims, quantity and relevance.

- *Quantity*: In the development of a context space, only specify those aspects of the referent being discussed that are needed for the accomplishment of the one specific conversational move served by this context space.
- *Relevance*: To be relevant means either to embellish the active context space by stating a succeeding utterance that continues the current conversational move or to shift to a different context space (new or old) through the development of a distinct conversational move, whose relationship to some previous context space is clear. If all previous topics have been fully developed (all criteria for the development of particular conversational move are satisfied), then the new move may begin an entirely topic. If, on the other hand, there are uncompleted conversational moves in the discourse, the new move will either constitute a temporary interruption or will have to clearly signal which portion of the previous discourse it is related to.

Effective design of interfaces for complex tasks requires some kind of model of their cognition. Grant (1992) uses the term “context” for the conceptual entity that have some features in common with scripts, frames or schemata as developed in human cognition. The view that human knowledge structures are divided into small units is the basic assumption underlying a modular contextual view (e.g., the contextual modularity in SOAR is expressed in terms of separate space problems). The context here is a candidate for something that is stored in long-term memory, and recalled as a whole, as a viable unit of task strategy appropriate to some stage of some task. The essence of the contextual modular view is that regularities appropriate to certain contexts are stored together, and are accessible together. However, the author goes beyond these previous models including the knowledge necessary for context changing in the context itself rather than being controlled by some separate process. Grant (1994) considers two different kinds of transition between contextual modules: learned (context-specific transitions) and general (associative transitions).

2.4 Context in electronic documentation

The notion of context is crucial in navigation problems. In electronic documentation, Boy (1991a, 1991b, 1995) proposes a mechanism for incrementally constructing contextual conditions from scratch in a formalism of block knowledge base. The basic entity is a “block” that includes five characteristics: a goal, actions, initial conditions, abnormal conditions and contextual conditions. Blocks are organized by context. A situation pattern is a problem statement that will activate a context of blocks. Contextual conditions, or contexts, are distinguished from preconditions to express a different granularity of representation of knowledge in the hierarchy of triggering preconditions. Contextual conditions represent a very compiled expression of knowledge and are represented as a disjunctive set of predicates and organized in hierarchies.

Compton and Jansen (1988) propose a representation of the contextual property that is based on a formalism called “ripple-down rules”. The ripple-down rule knowledge representation attempts to capture and apply the context to reduce the maintenance activity. The context defines when the knowledge should be considered. Without it, the chunk of knowledge can be used inappropriately by the inference engine. As proposed by Sowa, the idea of context is closely related to concepts, in that contexts can be viewed as concepts in their own right, albeit complex ones (i.e., having complex topologies), as well as relationships between concepts. The context is defined dynamically: the properties of a view must be extracted from the properties of the object. “Context should be treated as a living entity, that grows and changes shape over time, depending on the requirements of the end users, and the increasing intelligence of the system.” The authors organized their discussion about context around Minsky’s idea of K_lines (Minsky, 1986).

2.5 Context in vision

Context is an important factor in different applications in the area of vision. It plays a role in character recognition (Toussaint, 1978), target recognition (Forman et al., 1984), image recognition

(Mohr & Masini, 1982). In a vision system, the role of the context is the following (Desvignes et al., 1989): guiding the research, solving ambiguities, filling gaps, correcting errors and learning. Desvignes et al. (1991) propose the SISI system to exploit context (representation and exploitation of information related to context) and its role in the interpretation of a sequence of images. The authors define context as the set of properties that are associated with an entity according to the environment in which the entity is. Using context effectively improves qualitative and quantitative performances (relevant interpretation and minimization of the needed tests). However, there is to find a compromise between the time for analyzing context and the improvement in the image analysis.

More recently, an IEEE Workshop on Context-Based Vision (CBVIS '95), which studied the use of context in vision, was held in 1995. According to participants, context is considered for change detection (site updates or activities, and irrelevant changes due to illumination differences or seasonal variations) (Burlina et al., 1995), for mapping from sequences of image events to actions the system is to perform (Kjeldsen & Kender, 1995), for tracking between frames (Bobick & Pinhanez, 1995; Intile & Bobick, 1995). Globally, it is acknowledged that the use of context allows for the design of less complex and more robust image understanding algorithms.

Brémond and Thonnat (1997) propose a definition of context through the description of the different information types manipulated by a process. They give an explanation of the role of the granularity level of processing and the role of the abstraction level of the application in modelling context. The authors then illustrate the interest of their definition with the example of a scene interpretation process. A main result of their work is to give different viewpoints of a particular representation.

2.6 The PEPYS project

The idea of Lamming and Flynn (1994) is that physical context can be a powerful cue for recall. Researchers in psychology have developed theories about this sort of memory, called episodic or autobiographical memory. They observe that we naturally organize our memories of past events into episodes, and the location of the episode, who was there, what was going on, and what happened before or after, are all strong cues for recall. Thus, the authors propose computer-based support for human memory (finding a lost document, remembering somebody's name, recalling how to operate a piece of machinery), called the Intimate Computing Model. The user's context—acting as a retrieval key—is supposed to provide a valuable key for indexing information automatically. A detail from a past event in which the user was involved might be difficult to recall, the name of a document, for example. However, the context of the event can be easier to remember. For example, we may be able to recall the place where the document was received, the people present when it was handed over, or the task being carried out at the time. Being involved in many of your activities, it can become intimately familiar with them, and adapt to them like a personal assistant. The more the intimate computer knows you, the greater its potential value to you. While personal computing provides you with access to its own working context—often a virtual desktop—intimate computing provides your computer with access to your real context.

Newman et al. (1994) present, along the same line of reasoning, one part of a broad research project entitled Activity-Based Information (AIR) which is being carried out at EuroPARC. The basic hypothesis of this project is that if contextual data about human activities can be automatically captured and later presented as recognizable descriptions of past episodes, then human memory of those past episodes can be improved. They describe an application called PEPYS, designed to yield descriptions of episodes based on automatically collected location data. The program—made concrete by an active badge that each employee carries—pays particular attention to meetings and other episodes involving two or more people. The episodes are presented to the user as diary generated at the end of each day and distributed by electronic mail. After reading their diaries, users remembered events that they had previously forgotten.

3 Context in artificial intelligence

3.1 Failures in first generations expert systems

In artificial intelligence, the lack of explicit representation of context is one of the reasons for the failure of many Knowledge-Based Systems (KBS) (Brézillon & Pomerol, 1996a, b). Studies of the use of KBSs in real-world applications make it possible to point out four main failures.

The first failure is the exclusion of the user from the problem solving. This is because KBSs are expected to be oracles and users are expected to be novice followers, an approach which works well when a KBS is asked to solve a problem to which it knows the solution. However, in practice, problems which the KBS explicitly knows how to solve are the exception rather than the norm. KBSs cannot solve such unexpected problems when users, with their practical experience, are not given the opportunity. Excluding the user thus means that the KBS is often useless. This failure is due to the lack of user-system cooperation and because the context in which a problem is solved is not considered in relation to context for the original knowledge acquisition.

The second failure arises when KBSs do not correctly use their knowledge. Knowledge acquired from human experts has a high contextual component that is generally not acquired with the knowledge. Context of use must also be elicited.

The third failure is because KBSs cannot initially have all the necessary knowledge. Any KBS has bounded resources for problem solving and bounded influence: One can never anticipate or “design away” all the misunderstandings and problems that might arise during the use of such systems (Fischer, 1990). This implies that knowledge must be acquired incrementally when needed, i.e., in a given context of use.

The fourth failure is that KBSs cannot generate relevant explanations for users because they do not know what the user’s problem solving context is. The unique way to generate an explanation is that the KBS and the user jointly construct the explanation (Karsenty & Brézillon, 1995). This implies that one must take into account the context in which a KBS and a user cooperate for problem solving.

The main problems arise mainly because the users that work with computer systems are not taken into account. We now give two examples.

Schmidt (1995) describes aspects that are common to cognition situated in front of automatic dispensers. His conclusion is that the interface often is a barrier to communication between an end-user and machines such as automatic dispensers. Such machines impose a forced interview, a prior determination of all the interaction, and a set of instructions that is the same for all the end-users. However, the more instructions that are displayed in an explicit manner by the machine, the less know-how is required from the user. In such an approach, there is no consideration for the user’s context and its evolution during interaction with the system.

Vanwelkenhuysen and Mizoguchi (1995) describe work practice of two troubleshooters—testers and engineers—working on the same devices (digital processor boards in a telecommunications production plant), with access to the same information but adapted to different workplaces. They find that the two troubleshooters never came to an agreement because they solved their problems differently (e.g., oscilloscope versus logic state analyzer), each way being effective for routine problems in their workplace but inadequate for the other’s. The main difference arises from a different viewpoint about the priority of performance requirements (e.g., fidelity and precision versus efficiency). Thus, context appears more as a mechanism for presenting knowledge rather than for modeling knowledge. Such problems also are encountered when several humans work together. We evoke now this situation through two examples.

Grusenmeyer (1995) underlines the fact that functional communication is an essential means for operators to cooperate, coordinate and adjust their knowledge and representations. This is particularly crucial at the changing of operators: the incoming operator must adjust his understanding and operational representation of the current situation with those of the outgoing operator. Outgoing and incoming operators need to share knowledge and build common references. Functional communication increases crucially when the change-over occurs during a malfunction-

ing phase of the process that they have to monitor successively. The verbal exchanges during this phase are not reported on working documents because it is expensive and strongly reduces the possibilities of these exchanges.

In the workplace, executives develop ways of working (strategies, relationships among themselves and so on) to reach the efficiency that decision-makers require when they design the work. Part of their practical solving of problems is not coded (Hatchuel & Weil, 1992). Such know-how is generally elaborated case-by-case in unwritten rules. An unwritten rule takes into account the real context of the problem at a given moment. Such “makeshift repairs” permit the executive actors to reach the required efficiency. This is a manner to reach the solution whatever the path that is followed. The validation of such unwritten rules is more linked to the result than to the procedure used to reach it. This is the logic of efficiency (de Terssac, 1992).

3.2 Using context

In AI, context was first discussed by McCarthy who argued that contexts should be treated as first-class objects with a formal set of axioms that govern their interactions with one another and with all the other features of logic (McCarthy, 1979). Contexts are rich objects in the sense that they cannot be completely described. Context makes it possible to make explicit all what is implicit in a proposition. As a consequence, context makes reasoning local (Giunchiglia, 1993).

Other authors try to use context explicitly:

1. Hendrix proposes a partition of semantic networks by managing context (Hendrix, 1975).
2. McDermott (1982) describes the use of substructures in R1 (called contexts) to organize subtasks. Thus, functionalities in R1 are represented by separate contexts and their associated rules.
3. A subgoal in SOAR is considered in the context of a stack that comprises earlier decisions about goals, problem spaces, states and operators (Laird et al., 1987). Stack contexts are organized in a hierarchy of increasingly specific contexts.
4. de Kleer states that a context-sensitive consistency provides a good way of achieving control in problem solving without consideration of the overall consistency of the knowledge in the system (de Kleer, 1987).
5. Guha and Lenat use context in CYC for partitioning a knowledge base into more manageable modules (Guha, 1991, 1993).
6. Packages in Common Lisp are supposed to do for programming what contexts are supposed to do for representation. Different packages may have different symbols and the same symbol may have different values in different packages (Guha, 1991).
7. The Carnegie Representation Language (CRL) of Knowledge Craft proposes to organize the knowledge representation in a context tree. If the current context does not possess the needed information, CRL looks for the information in the context parents. Thus, there is a recursive climbing until the root of contexts.
8. Aikins CENTAUR system explored the use of context-like prototypes to partition the ruleset. (Aikins, 1983).

3.3 Context and explanation

Leake (1992) considers the relationships between explanations and context in the framework of case-based reasoning. An explanation is required when there is a conflict between an event and a model that we have of the place where the event occurs. Leake argues that such a conflict is a property of the interaction between events and context: Any particular fact can be anomalous or non anomalous, depending on the situation and on the processing we are doing. To be relevant to an anomaly, explanations must resolve the belief conflict underlying the anomaly. To resolve an anomaly, the information in an explanation must account for why prior reasoning led to false expectations or beliefs. Any anomaly vocabulary would allow retrieval of explanation for identical

anomalies, provided that the same anomaly was always described the same way and that distinct anomalies always received distinct characterization.

Leake lists ten major explanation purposes triggered by anomalies that rely on several elements of context (expected/believed conditions, previously unexpected conditions, possible repair points, actor's motivations, etc.). Indeed, Leake acknowledges that Mackie (1965) already stressed the context-dependence of explanation as a process of making a distinction between some current situation and another class of situations. Thus, context—involving both beliefs and goals of the explainer—is crucial in deciding the quality of an explanation, and a theory of contextual influences can be used to determine which explanations are appropriate. Deciding the quality of explanations makes sense only in the context in which an explanation has been triggered and is dependent on how the resultant explanation will be used. This leads to the view that explanation is primarily a process of hypothesizing causes of events and expectation failures, rather than deriving them from known factors.

These ideas have been implemented in ACCEPTER. ACCEPTER characterizes anomalies to index into explanatory information stored in memory. The facts that ACCEPTER stores during routine understanding, and the expectations based on those facts, become the context in which new inputs are understood. ACCEPTER focuses its explanation effort on filling current gaps in its knowledge. It maintains conflicting families of beliefs by placing them in hierarchical understanding contexts, which are used to maintain a tree of alternative world models; by evaluating competing explanations in different contexts, ACCEPTER avoids interactions between their beliefs.

Most researchers focus on explanations considering them as a transfer of knowledge from the system to the user. Feedback is used by the system to tailor its explanation to user's needs. However, users rarely may intervene in the generation of explanations. An opposite position is taken in SEPT by letting the user build his explanation alone (Brézillon, 1990). However, this is not a better solution than the previous one: users must tackle complex commands that are not always compatible with their work and temporal constraints. The lesson learned is that the user and the system must cooperate to solve jointly the problem and to co-construct an explanation for the solution (Karsenty & Brézillon, 1995). Figure 1 illustrates this lesson.

As represented in Figure 1a, MYCIN, and all earlier expert systems, align their explanation facility directly with the reasoning paths that define movement across contexts of the diagnostic system. Thus, all traces of reasoning that represent the contexts which have been traversed are kept and their contents provided to the user for explanation. Here, the definition of context is restricted to knowledge and reasoning of the problem solving. MYCIN is not selective in its construction of explanations.

In other systems, such as that described by Wick (Paris et al., 1988; Wick & Thompson, 1992), an explanation facility is aligned only periodically with the reasoning of the system (see Figure 1b). In Wick's system, only some of the contexts that the system reasons with, are explained to the user. In

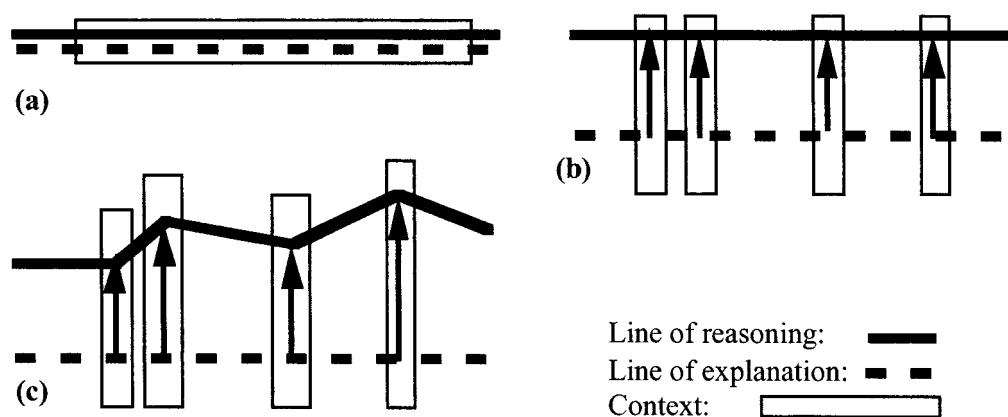


Figure 1 Line of reasoning versus line of explanation.

this approach, additional explanatory knowledge (knowledge about the domain and the expertise which are not directly necessary for the task at hand) may be used to generate enhanced explanations. This implies that the explanation path separates from the path of reasoning to produce effective explanations. Context here is an extended version of the previous one because it also contains domain and task knowledge not directly considered in the reasoning of the problem solving, and eventually leads to a model which gives some information about users. One problem with such an approach is that it may be unsuitable for critical applications whose results may affect the safety of processes and people.

Another approach for explanation is to accept that the reasoning of the system is often different from that of the user. Thus, the user and the system may have different interpretations of the current state of the problem solving. The differing interpretations will be compatible if the user and the system make proposals, explain their viewpoints and spontaneously produce information (Karsenty & Brézillon 1995). To align the system's reasoning with that of the user and *vice versa*, the user and the system must co-construct the explanation in the current context of the problem solving. People who are trying to understand something often may offer an explanation that embodies their current understanding, expecting to have it corrected (Mark, 1988). Thus, explanations become an intrinsic part of the problem solving and, as a consequence, the line of reasoning of the system may be modified by explanation (see Figure 1c). This leads to cooperative problem solving. Again, context here is an extended version of the context in the previous approach because it also integrates direct information from users, mainly on the basis of their actions on the system and on the real-world process.

Explanation and context are strongly intertwined. Making context explicit permits the tailoring of explanations to a precise need, to decrease the amount of knowledge required for the exchange between the user and the KBS, to show the coherence of an explanation and rapidly reach an agreement between the user and the system. Explanations permit the context to be explicit to understand a step of the reasoning and shared knowledge and experience. They are a means to point out the links between the problem at hand and shared knowledge on its current state.

The way in which an explanation must be chosen and produced depends essentially on the context in which the explainer and the explainee find themselves. The sensitivity of "knowledge" to context becomes evident when one attempts to implement an automated explainer that chooses between alternate explanations (Suthers, 1993). An explanation always takes place relative to a space of alternatives that require different explanations according to the current context. Comparing two explanations leads to see how their contrast spaces differ. This gives us a measure of dislocation between two explanations and a basic presupposition of the explanation context. Thus, an additional piece of structure—the explanation context—is necessary to explain how explanations function (Garfinkel, 1981; Lester & Porter, 1991).

Context appears to guide the interactions between actors and the search for suitable explanations. Interaction is seen as navigation in a context space. For instance, Huuskonen and Kortenieni (1992) propose following different steps in order to produce an explanation accounting for the context:

- Find the context (context can be inferred from previous explanations, assuming that the user likes to ask more on the same subject);
- Find the question: questions matching with the context are displayed, and the user chooses one among them;
- Refine the context if necessary;
- Find the answer and show it.

With such a method, it seems that the user must easily accept the idea of conversation through context refinement. According to this view, producing acceptable explanations means identifying mutual knowledge of specific events, objects and contexts and relating explanations to the explainee's personal characteristics and the interaction history (Johnson & Johnson, 1992). Using context permits a kind of "cognitive coupling" between the user and the system.

If it seems acceptable that explanations intervene in the evolving context of interaction, it is difficult to say more about this for two reasons. First, the co-building of explanations is an accepted idea but rather very few studies consider it. Secondly, because context is not a mature domain of research, its effect upon explanations has not been considered very widely. For example, Lester and Porter (1991) propose a model of explanation generation that includes simple methods for representing and updating context. However, their model makes assumptions about the representation of the context, not about how it is inferred.

3.4 Context and knowledge acquisition

Knowledge acquisition is a difficult and time-consuming task. The difficulty arises because experts do not report on how they reach a decision. As a consequence, the decision is acquired out of its context when experts justify why the decision is correct within a specific context (Jansen, 1993). Knowledge acquisition is generally considered either before the building of the system or during the use of the system.

The former is the main approach that is followed in the knowledge-acquisition community. For example, Walther et al. (1992) describe a context-definition language called MODEL for the PROTEGE-II system. MODEL is a metatool architecture that associates an explicit data model with every reusable module in the tool libraries. Such a data model describes the externally visible aspects of every module, and defines a context ontology for reusing the module because it contains all necessary information for defining the module's role in the module-assembly process. A context ontology does not differ significantly from any other knowledge-representation system: Every context contains sets of concepts (also called schemas, frames, or structures) that describe the basic terms used to encode knowledge in the ontology. Furthermore, each context contains a set of constraints that restrict the manner in which instances of these concepts may be created and combined. In addition to these basic functions, however, the role of context ontologies places a number of further requirements on the representation language.

Encoding knowledge as part of the task at hand leads to contextualized knowledge acquisition where knowledge is encoded into the system when it is needed. Contextualizing the knowledge acquisition process helps to ensure that relevant knowledge is put in the knowledge base, based on the premise that you won't know what is really needed until you're in the design process (Henninger, 1992). We are here in the realm of incremental knowledge acquisition.

There are different ways to acquire knowledge in context:

- Compton and Jansen (1988) address the long term maintenance of expert systems. They attempt to capture the context by entering the expert's new rule directly as provided, including an "IF LAST_FIRED (rule n°)" condition. That is, the new rule will not fire on a case unless the old rule, which produced the wrong interpretation, has fired first. Thus the new rules are tested precisely in the context in which the expert provided them, that is the portion of the expert system that comes before this rule is exactly the same as the expert system which produced the interpretation coming before this rule. (This representation is used in a 'Knowledge Dictionary' (Jansen & Grosz, 1990).)
- Gruber (1991) proposes considering justification-based knowledge acquisition to divide the load in knowledge acquisition between a cooperative user/teacher and elicitation program. The machine provides the computational medium, including the knowledge representation and the context of use, such that every thing that is acquired from the user can be assimilated into the computational model. The knowledge acquired using the justification technique is guaranteed to be operational because the user always conveys something to the machine by getting the machine to say it. (A similar view is presented in Srinivasan et al., 1991.)
- Bloom et al. (1992) present a task driven approach to acquiring, analyzing and representing knowledge for an intelligent tutoring system (ITS). This approach provides the context to ensure

that the knowledge acquired and represented is the knowledge required to support task performance.

- O'Hara et al. (1992) propose the access to a model of the problem-solving process that will help users to select an action. It is acknowledged that the governing model would contextualize the information used.

The main claim of all these approaches is that experts provide their knowledge in a specific context and the knowledge can only be relied upon in this context. This context is largely determined by the case that prompted the change to the knowledge base (Compton et al., 1991; Paton et al., 1993). Knowledge is not to be generalized when it is acquired. It is fundamental to record the context in which the knowledge is acquired (Compton & Jansen, 1990; Guha, 1991; Turner, 1993).

However, the acquisition of knowledge in context is still a challenge. First, even what we take to be a highly stable behavior, such as reciting a phone number, is highly contextual. You establish this context by sitting in front of a phone (Clancey, 1991). Such situated knowledge is not acquired with the classical tools prior its use. Considering to acquire the knowledge when needed, implies that the knowledge is compatible with the computational medium of the system (see the position of Gruber above). In Clancey's example, one generally faces automatisms that authorize to dial the phone number without formulating it. For instance, one only "sees" the sequence of physical positions that the finger must have on the phone keyboard. It is rather difficult to acquire such knowledge that may be expressed in a representation formalism not known by the machine.

Secondly, as well as capturing knowledge in context, it is also essential that the knowledge in a knowledge base should be able to be examined in any other context (Compton & Jansen, 1990). One advantage is that it allows different contexts to be compared, in particular how the corrections to knowledge in a particular context may be used to review conclusions drawn in another context. How may knowledge that is acquired in a given context be used out of this context?

Although the approaches described in this section yet present serious weaknesses, they are ascribed in the realm of the incremental knowledge acquisition in context (Brézillon & Abu-Hakima, 1995; Brézillon & Cases, 1995).

3.5 Context and machine learning

The meaning of many concepts heavily depends on some implicit context, and changes in that context can cause more or less radical changes in the concepts. Incremental concept learning in such domains requires the ability to recognize and adapt to such changes. Widmer (1996) presents a general two-level learning model, and its realization in his METAL(B) system. This system can learn to detect certain types of contextual clues, and can react accordingly with context changes. The model consists of a base level learner that performs the regular on-line learning and classification task, and a meta-learner that identifies potential contextual clues. His operational definition of contextual attributes is based on the notion of predictive features. A feature is considered as a contextual clue if it does not directly determine or influence the class of an object, but if there is a strong correlation between its temporal distribution of values and the time when certain other attributes are predictive. Intuitively, a contextual attribute is one that could be used to predict which attributes are predictive at any point in time. In a similar spirit, Turney (1996) reviews five heuristic strategies for handling context-sensitive features in supervised machine learning from examples to recover hidden (implicit, missing) contextual information. Then he presents two methods for recovering lost (implicit) contextual information.

Park and Wilkins (1990) describe a system for failure driven learning which uses a context analysis mechanism as a method to constrain explanations and thereby increase the number of learning opportunities by 17% and increases the overall amount of improvement to the expert system by around 10%. The context analysis program maps an observed action on the explanation plane. An explanation on this plane has a pointer to a set of actions that are explained by the explanation. Such an explanation becomes a sub-context that explains a subset of observed actions. The context

analysis program can find a sub-context that explains all the actions. This sub-context is considered as the context of the observed actions. Here, contexts are judged to be similar if their strategy axes are the same and their focus axes can be grouped by a known relation. For example, consider $\text{context1} = (\text{clarify-finding, surgery})$ and $\text{context2} = (\text{clarify-finding, neurosurgery})$. Since surgery and neurosurgery are defined by a relation *more_specific*, a transition from context1 to context2 is considered as a natural one.

Interface agents assist users with daily computer-based tasks, learning by ‘watching over the shoulder’ of the user and detecting patterns and regularities in the user’s behavior. The learning approach, which is proposed by Lashkri et al. (1994), achieves a level of personalization that it is impossible for a knowledge engineer to obtain, without the user intervention required by rule-based systems. However such agents have always trouble with new situations. Thus, the idea is to permit agents to exchange their knowledge and collaborate. When a user takes an action, it is paired with the corresponding situation and the situation-action pair is recorded in the agent’s memory. For example, if the user reads a message *M*, the pair $\langle M', \text{read-action} \rangle$ is memorized, where *M'* contains details about *M* and relevant context information.

3.6 The CYC project

One of the primary reason for considering contexts explicitly is to simplify the construction of a knowledge base. The context mechanism was introduced into the CYC system in May 1990. In November 1991, CYC contained over 1.5 million sentences and covered a wide range of phenomena (Guha, 1991). Contexts are considered as rich objects in a first-order framework, extending the logic as required (Guha, 1993). The basic change is that formulas are not just true or false; they are true or false in a context. This permits us to make statements “about” contexts, although they cannot be completely described (McCarthy, 1993).

The context of an utterance is set up by a very wide variety of parameters. These parameters range from very coarse-grained factors such as the cultural settings and the socio-economical backgrounds of the conversants, to medium-grained factors such as the goals of the conversants, to fine-grained factors such as the immediately preceding utterance or even preceding gestures. A context is defined as a consistent set of propositional assumptions about which something can be said. Such a set forms a theory of some topic, e.g., a theory of mechanics, a theory of the weather in winter, etc. In that sense, a context is called a “Microtheory”. The scope of a context (the theory associated with the context) is the set of objects over which its predictions hold.

There are as many contexts as sets of assumptions under consideration. Based on a statement made about an object in one context, something may be derived about that object in another context. The two contexts use different vocabularies and associate different attributes with an object, but these attributions are about the same object. So, there might be some contexts in which it might not be possible to state *P* (in the vocabulary of that context) and there might be yet other contexts in which *P* is stated in different ways. Different expressions might be used by different contexts to state the same fact or the same expression might be used by different contexts for stating the same fact or the same expression might mean different things in different contexts. The meaningfulness of a formula may depend on the context it occurs in. Different contexts make it possible to use different languages.

Huhns et al. (1993) use the CYC knowledge base as a context. Here, a model is a set of frames and slots in a CYC context created especially for it. The mapping between each model and the global context (the CYC knowledge base) is captured in a set of articulation axioms. The models of individual resources are compared and merged with CYC but not with each other, making a global context much easier to construct and maintain. The authors find that using CYC is significant, because of (1) its size: it covers a large portion of the real world and the subject matter of most information resources; (2) its rich set of abstractions, which ease the process of representing predefined groupings of concepts; (3) its knowledge representation mechanisms, which are needed

to construct, represent, and maintain a global context, and (4) its typing mechanism, which is used to integrate and check the consistency of query results.

3.7 The ORCA project

Turner (1993, 1995, 1997) has developed a system—an adaptive reasoner—to make context explicit for autonomous underwater vehicles to tackle unanticipated events in complex environments. Contextual information helps the agent to focus its attention on appropriate goals to achieve in the current situation. Thus, context intervenes in at least five different ways: (1) to make predictions about the situation; (2) to modulate an agent's behavior; (3) to focus agent's attention; (4) to influence an agent's choice of actions; and (5) to determine how an agent should handle unanticipated events. An agent should be able to recognize its current context as an instance of a class of contexts it knows about. It should be able to reason about its context, bringing to bear knowledge that is explicitly known to be contextual in nature.

Contextual knowledge is represented as a set of contextual schemas (c-schemas). The system then retrieves the most appropriate of those and uses them to help the reasoner behave appropriately in its current context. Thus, c-schemas contain information not only describing the context they represent, but also information prescribing how to behave in situations that are instances of that context. Schemas provide a natural way to represent contexts which should facilitate knowledge acquisition and potentially provide a tie to established machine learning approaches such as case-based reasoning.

An agent's context manager retrieves the best c-schemas from its memory based on features of its current situation, then merges them to form a view of the current context, the current c-schema. Thus, relatively few contexts are represented as c-schemas, but they are combined as necessary to adequately represent a particular situation. The major difference with case-based reasoning is how c-schemas are used: generalized cases are usually used as indexing structures, while c-schemas are problem-solving structures in addition to their role in memory organization. Context is mainly considered as a way to cluster knowledge for search efficiency, for representing counter-factual or hypothetical situations, for circumscribing the effects of particular actions to particular situations, and for directing an agent's focus of attention to salient features of a situation.

4 Representation of context

4.1 Logical representations of context

Recently, McCarthy (1993) defined a context as a generalization of a collection of assumptions. Contexts are formalized as first class objects (formal objects), and the basic relation is $ist(c, p)$. It asserts that the proposition p is true in the context c , where c is meant to capture all that is not explicit in p that is required to make p a meaningful statement representing what it is intended to state. Formulas $ist(c, p)$ are always asserted within a context, i.e., something like $ist(c', ist(c, p)) : c' : ist(c, p)$. The consequences are:

1. a context is always relative to another context;
2. contexts have an infinite dimension;
3. contexts can not be described completely;
4. when several contexts occur in a discussion, there is a common context above all of them into which all terms and predicates can be lifted.

There are many other relations among contexts and context valued functions. Each context has a vocabulary associated with it. So, there might be some contexts in which P might not be stated (in the vocabulary of that context) and there might be yet other contexts in which P is stated differently. Different expressions might be used by different contexts for stating the same fact or the same

expression might be used by different contexts for stating the same fact or the same expression might mean different things in different contexts. Thus context makes reasoning local (Giunchiglia, 1993).

However, the logical machinery is only a small fraction of the effort involved in building a context-based system. The bulk of the effort lies in writing the axioms describing and interrelating contexts. The structure and content of these axioms—the *lifting rules*—are heavily dependent on the kind of use. The most common operation on contexts is to lift a formula from one context into another. Doing this requires relative (partial) *decontextualization*, i.e., the differences between the origin and target contexts have to be taken into account to obtain a formula with the same truth conditions as the original formula had in the original context. (Note that a formula relating two contexts could involve contextual assumptions and is therefore itself in a context.)

The context of the system is the current context of the problem solving activity. All interactions with the system take place in this context, and information must be lifted from other contexts into this current Problem Solving Context. The current context is the physical/real memory, the other contexts are the virtual memory. Contexts of a problem solving task are usually created dynamically by the system and are ephemeral.

Mechanisms for relating and translating between contexts are acknowledged as vital to the effective reuse of domain theories in new problem-solvers. There are two classes of context: (1) Representational context captures the total set of qualifications relative to which the symbols in the language of a theory are abstracted at a pertinent level of relevance; and (2) Computational context represents the focus of the reasoning—the set of assumptions made or path taken by a reasoner in evaluating a current hypothesis. Sharma (1995) gives a list of desirable properties for contexts in a formal language and distinguishes four approaches for formalizing contexts: (1) incrementing arity; (2) variation on implication; (3) modal operator forms; and (4) syntactic treatment. Guinchiglia (1993), Sevakini and Ghidini (1997), Guinchiglia and Ghidini (1997), follow such approach by trying to mechanize contextual reasoning mainly with datasets.

Extending McCarthy's work on context logic, Farquhar et al. (1995) present an approach to integrating disparate heterogeneous information sources. They show that the use of a context logic reduces the up-front cost of integration, and allows semantic conflicts within a single information source or between information sources to be expressed and resolved. Two contexts are used to represent each information source. The information source context is a direct translation of a database schema into logic without resolving semantic conflicts, so that the translation can be done automatically. The semantic context holds the translation with the semantic conflicts resolved. An integrating context contains axioms that lift sentences from several semantic (or integrating) contexts. The consequences of using a context logic to integrate information sources are that: it integrates new information sources incrementally; it shares assumptions among information sources without making them explicit; it exploits shared ontologies; and it provides a richer model of integration that goes beyond global schema or federated schema methodologies.

There are a number of applications of the ATMS developed by de Kleer (1987), which can be considered as a logical method of representing context. For example, in design applications, multiple design solutions or alternative design solutions are due to under-constrained design variables and parameters. Design solutions must be derived from an environment (i.e., a context) where design requirements, design methods and design evaluation criteria are subject to frequent change. In particular, Tang (1995) proposes ATMB, a Lisp-based software architecture for an intelligent design support. ATMB is based on the integration of a blackboard control system and an ATMS to support design context exploration. The main advantages of ATMS-based context management systems are the capabilities of maintaining consistency of knowledge and multiple contexts. A component of the system is a context manager that: retains as much of the knowledge generated as possible; provides easy access to, and a good explanation of this knowledge; makes the best use of the knowledge already held in the dynamic knowledge base to enable it to generate new knowledge without performing redundant inference; and helps the user to compare different, sometimes conflicting solutions.

4.2 Rule-based representation of context

In a rule-based representation, context may be expressed either in the knowledge itself (if explicitly represented) or the functionality of the chosen representation formalism.

When the knowledge is viewed at the appropriate level, we can often see the existence of organizations of knowledge that bring up only a small, highly relevant body of knowledge without any need for conflict resolution (Chandrasekaran et al., 1992). For instance, Brézillon (1990) presents the diagnostic expert system SEPT that deals with pieces of equipment as circuit breakers and protective relays. Checking the internal behavior of a circuit breaker implies the use of expertise that is independent of any expertise about the internal behavior of a protective relay. Thus, the reasoning is local and does not need to tackle the overall expertise, and in the context of the circuit-breaker diagnosis only knowledge structures are represented.

In rule-based formalism, knowledge structures are rule packets represented either at the level of the rules or at the level of the knowledge base. The former is managed by screening clauses, which are controlled by special rules (Clancey, 1983; Eklund, 1989) and the latter organizes the knowledge base in a set of distinct small knowledge bases managed either directly by rules that call rule packets in their THEN part (Brézillon, 1991) or by interactions among rule packets for exchanging information.

For a representation at the rule level, the well-known example of a screening clause is the following rule in MYCIN (Clancey, 1983, 1993):

```
IF
  1. The infection which requires therapy is meningitis,
  2. Only circumstantial evidence is available for this case,
  3. The type of meningitis is bacterial,
  4. The age of the patient is greater than 17 years old, and
  5. The patient is an alcoholic,
THEN
  There is evidence that the organisms, which might be causing
  the infection, are diplococcus-pneumoniae (.3) or e.coli (.2)
```

Such a rule is composed of different types of knowledge (strategic knowledge, causal knowledge, etc.). Clause 4 had been added to control the interaction with the user. The clause acts as a screening clause and implies that this rule is only valid in the context of an adult. It is not a clause for the identification of the infection. Such knowledge does not intervene directly in the problem solving, just constrains it. (Note that mixing different types of knowledge—including context—in an implicit way implies that the system is unable to explain its reasoning.) Screening clauses are stated by rules that act as meta-rules.

At the knowledge-base level, an example is given by the following rule of the SEPT expert system:

```
! check_cb
"Checking a circuit breaker of the protection system $name"
> if failure freeze check_pw, check_teac
IF
  equipment_piece (cos) := (cb) ,
  nature (cb) := circuit_breaker .
THEN
  call the rule packet 'Circuit-Breaker_Diagnosis(cos, cb)' .
```

The rule *check_cb* (written here in pseudo natural language) is used to trigger the diagnosis of a circuit breaker *cb* in a cut-off system *cos*. The rule belongs to a rule packet that checks all the equipment in the cut-off system. The rule packet represents the diagnosis expertise at the level of the cut-off system, and local diagnosis expertises on pieces of equipment are in other rule packets as *Circuit-Breaker_Diagnosis*. Firing the rule *check_cb*, the inference engine will enter the rule packet

Circuit-Breaker_Diagnosis with the instances fixed in the IF part for the variables *cos* and *cb* when the rule packet may be applied to all the circuit breakers in the substation (around 20 circuit breakers in an extra high voltage substation).

In such a rule, the context is expressed at two levels: (1) by a specialization of the circuit-breaker expertise for the given instances of the variables; and (2) management of the expertise at the cut off system level by the meta-knowledge *if failure, freeze choice-pw, choice_teac* that says that if a failure is found on the circuit breaker it is not necessary to check the equipment pieces *pw* and *teac* of the cut-off system. (See Mulet-Marquis and Gondran (1985) for more detail about ALOUETTE. Note that ALOUETTE has been extended to incorporate the METAL language to handle context better.)

It seems that this process of contextual instantiation of the variables plays an important role. In linguistics, Récanati (1992) proposes representing linguistic meaning as a formula where a number of variables are instantiated in context by assigning them a value. Among the other approaches of context management in rule-based formalism, there are: lifting rules (Guha, 1991), bridge rules (Giunchiglia, 1993), pragmatic rules (Giroto, 1991) and ripple-down rules (Compton & Jansen, 1990).

4.3 Other representations of context

There are several theoretical approaches, in addition to those presented above, which consider context explicitly or implicitly. Most of these approaches try to represent context, not model it. Note that we distinguish between modelling and representation. The goal of a model is to give a coherent picture of context that can be used for explaining and predicting by simulation. The goal of a representation of context is only to account for what is observed regardless of how this is done. A model is embodied in a theory and a representation depends upon the representation formalism chosen. As the two goals are different, here we consider modeling to be at the theoretical level and representation to be at the programming level. A successful model is a model that is used in applications.

Context has been modeled using Situation Theory (Barewise & Perry, 1983). Situation theory is a unified mathematical theory of meaning and information content that is applied to specific areas of language, computation and cognition. The theory provides a system of abstract objects that make it possible to describe the meaning of both expressions and mental states in terms of the information they carry about the external world. Surav and Akman (1995) approach context as an amalgamation of a grounding situation and the rules that govern the relations within the context. They represent context by a situation type that supports two types of infon. Parameter-free infons are used to state the facts and the usual bindings. Parametric infons (which corresponds to parametric conditionals) aim to capture the if-then relations and axioms within the context. In Computer-Human Interaction, Nardi (1992) presents a study of context from a comparison of activity theory (Leont'ev, 1978), situation action models (Lave, 1988) and distributed cognition (Flor & Hutchins, 1991).

Ezhkova (1989) replaces knowledge base descriptions by semantic spaces, and then uses them for generating new knowledge bases as well as for making comparisons with, identifying intersections with, and performing integration with knowledge from other sources. Thus, context is considered as a semantic background. Ezhkova (1992) defines context in knowledge representation techniques on the basis of the concept of contextual system (CS). The main purpose of a CS is stated from the viewpoint of a decision making problem. According to this viewpoint, a global problem is reduced to a set of local problems where each one is stated for a specific set of decision alternatives and requires formation of context for its own problem area. A CS has two types of memory: a long-term memory (a primary database and a base of contexts) and a short-term memory (intracontext knowledge processing and intercontext knowledge processing). Ezhkova proposes an algebra of contexts which involves contraction, extension, immersion, coupling and intersection of contexts. Contexts are then stored or dynamically generated. For example, by contracting a context, one may

focus on certain sections of its description. There are different types of contraction of a context, since contraction can take place with respect to attribute significance, a set of attributes, the number of the most significant attributes, or a set of basic concepts. Conversely, context expansion operations are required for further learning, adaptation, introduction of new concepts, and immersion in a larger context.

Behind this concept, there is the idea of Problem Area Context (PA), which is a meta-notion relative to knowledge bases. The PA makes it possible to determine distances between the concepts of the context, the proximity of concepts behind decision making schemes being largely dependent on the problem area context. Two concepts may be close in one context and diametrically opposite in another one. The introduction of the contextual space makes it possible to determine a distance between these different concepts of the context. The distance is described in a universal manner. The distance between concepts in the context space is what lies behind the intra- and intercontext processing. The intercontext knowledge processing employs context algebra and logic. The former supports processes such as context contraction and extension, submersion of the given context in a wider one, integration and intersection of contexts that offer an interesting interpretation and formalize different non-traditional knowledge processing schemes. The CS technique has been applied in a problem of decision making in transportation (Bianco et al., 1994).

Other models of context have been proposed:

1. For Jaeger (1993), a context is a momentary mental setting that acts as a frame within which the intensional meaning of concepts is evaluated. It is composed of concepts in a dynamic way. Interactions among concepts have three aspects. First, they are the “gluing force” that holds the context together. Second, they modify the intensional meanings of the concepts taking part in them. Third, these interactions govern the temporal development of the context.
2. Leong (1992) presents contexts as relations that delimit the scope of the description of a concept in the conceptual network. They induce a context network which encompasses all concepts defined in the knowledge base. This gives a set of higher-order relations that constrain the interpretation of all other relation types in the framework. Thus, each concept is defined in some context that specifies the condition in which the description of the denoted concept is valid, and allows this description to vary, if necessary, from the basic entity.
3. Hovy (1990) has developed the PAULINE system which models context with a set of pragmatic parameters, such as how much time is available, how much the speaker and the hearer know about the topic at hand, and whether the speaker and hearer are calm, angry, or happy. A context-sensitive generator should interpose an intermediate decision level between communicative goals and realization decisions, and interleave the tasks of planning and realization.
4. For Politzer (1991), a context is a set of propositions with various origins (prior discourse, memory content, observations, inference, etc.). Joining a proposition P to a context, it is possible to derive a new proposition I when it is not possible from only either the proposition P or the context. The proposition I is a contextual implication of the proposition P. Thus, a proposition is pertinent when either it has at least one contextual implication or it confirms (or disconfirms) some propositions of the context. A pertinent proposition makes it possible to make inferences in a context.
5. Maurer (1992) uses contexts for modulation in MOLKE. A context represents a rough, intermediate, or final diagnosis. If its precondition is true, the associated failure is said to be proven and the related correction is executed. For example, a context (without rules) is defined by the following statement:

Context name: LIGHT-BULB
precondition: (SWITCH = CLOSED) & (LIGHT = OUT)
correction: “Change the light-bulb”

Contexts are organized in a context graph where arcs have the semantics “is-refinement-of” (e.g., the context ‘failure-in-electrical-subsystem’ is a refinement of ‘failure-in-car’). The diag-

nostic process goes through the context graph by testing symptoms according to the ordering rules of the actual context and switching to a refinement when its preconditions become (the logical value) true. If a leaf of the context graph is reached, the system prints the diagnosis and terminates.

6. Zuckermann (1992) proposes a Student Model for representing contextual information. A relationship is represented as:

[concept1 relation concept2 context)

The meaning of this representation is that the relation between concept1 and concept2 holds in a particular context, where the context is either the global context or an arbitrary sequence of nested relationships. For example, the representation:

beating has_goal get_whipped_cream (beating apply_to pure_cream))

means that beating will produce whipped_cream when the beating action is applied to pure_cream. In this manner, we provide a uniform representation to chain conditions.

7. Wu et al. (1992) propose a plan reuse framework in which restricted conceptual graphs are used as the internal representations of these skeletal plans and reusing these skeletal plans. The domain-specific relation set R only includes three relations, $R = \text{IS_PART_OF}, \text{IS_CONTEXT_OF}, \text{and IS_INSTANCE}$. For example, the IS_CONTEXT_OF relation represents that the labeled node has meaning only under the context of the linked labeled node.

In the ORCA project described in section 3.7, Turner developed a mechanism as part of his work on schema-based reasoning that uses contextual schemas (c-schemas) to explicitly represent contexts that an agent may encounter. C-schemas are organized into generalization-specialization hierarchies and represent relatively few contexts, but they are combined as necessary to adequately represent a particular situation that is more complex. The agent's context manager retrieves the best c-schemas from its memory based on features of its current situation, then merges them to form a view of the current context, the current c-schema. Each c-schema has several parts: a situation description, "standing orders"; attention-focusing information; action-selection information; and event-handling information.

Up to now, most software has used context in a static way (thus it has generally been imposed upon users). The preference file (.profile for example) under UNIX makes it possible for the user to specify his preferred working context for processes such as .xinitrc that records the properties of the screen/display desired by the user. Microsoft makes use of some contextual-sensitive help (for example balloons and scrolling menus). As another example, in the Macintosh operating system, the items in a popup menu are visible but are only active in some contexts, e.g., the item 'Copier' (Copy) is not active in Figure 2a because no file has been selected, but when the file "A gardier" (to keep) is selected as in Figure 2b, it becomes active. The list of items of a given popup menu may change from one working context to another.

Searching the Internet reveals some programming languages which make it possible to handle some aspects of context:

- <http://www.cis.ohio-state.edu/man/hpux/context.5.html>—The context is a set of character strings associated with each process. Each string corresponds to a characteristic of the machine the process is running on. The process context is used to access context dependent files. For example, since the MC68020 processor supports a superset of the MC68010 instruction set, processes running on a system with an MC68020 processor will have HP-MC68010 in their context, as well as HP-MC68020.
- http://quest.cc.purdue.edu/Paragon/Docs/Intel/Manuals/pipdman_6.html—The instruction 'context ((nodelist:ptypelist))' makes it possible to set the debug context to compute partition processes.

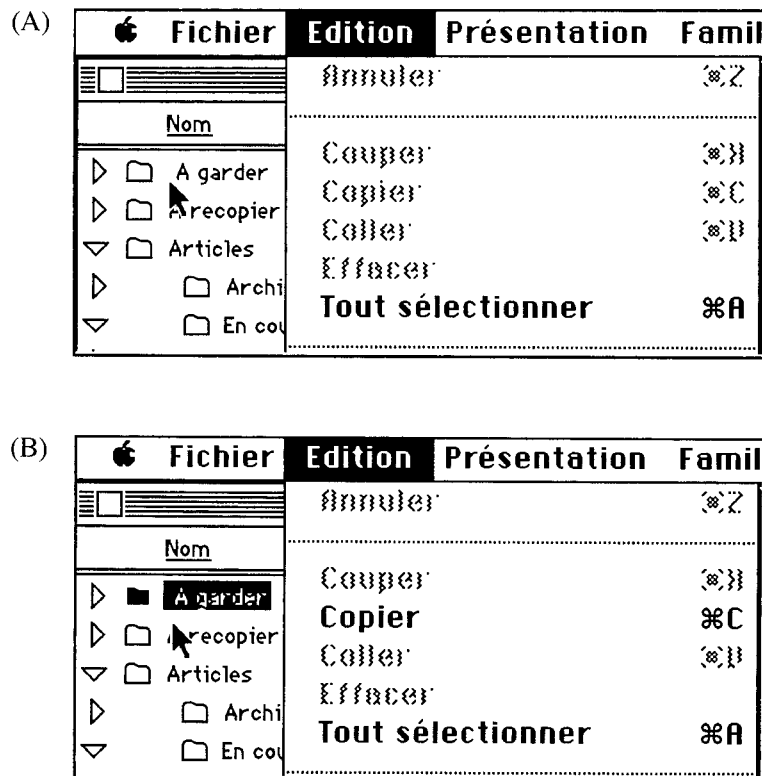


Figure 2 Context in an operating system. (a) The Edit popup menu without file selected; (b) the same Edit popup menu with a file selected.

- http://www.ci.com/support/ca/techtips/CONTEXT_RESTRI.html—Here context is represented as a set of restrictions that limit the access to parts of a system.

Most of these representations of contexts are used to manage information in some kind of heirarchy.

4.4 Relationships between these different views on context

In logic, the focus is on the infinite dimension of context. As a consequence, a given context cannot be described completely, one reason being that it is always relative to another context. A context is a type of representation with its semantics and its syntax. Thus, a given proposition P can have different expressions in two contexts. A context is associated with the focus of attention: “contextual reasoning is local reasoning” (Giunchiglia, 1993). However, the main problem is how to move from one context to another.

In a rule-based formalism, context is assimilated with control knowledge. Context is coded, generally implicitly, according to the representation formalism. For example, context is coded as pre- and postconditions that define when a rule is fired and what is possible after. The goal, as in logic, is to focus the system reasoning on the relevant pieces of knowledge. However, this is made with the technical performance in mind, limiting the list of facts and rules to check.

More generally, one tries to represent context in an ad hoc way, not to model it. The goal is to include in problem representations information on some characteristics of agents as mental state, beliefs, preferences, intentions, etc. The goal is to define the validity domain of concepts, and especially their relationships (Pomerol & Brézillon, 1997).

4.5 Example of context representation in database interrogation

We are currently studying¹ how to make context explicit in the interactions of a user and a database in order to make the system have a more active role in the dialogue. We guessed that if the designer can improve the interrogation language with feedback from the user, there would be a part that depends on the particularities of each user. Thus a system must be able to improve the user's requests dynamically when interacting with the user in a way that cannot be planned beforehand. Such contextual knowledge must be obtained empirically from the user, not from the designer.

Context may be considered at different levels. These levels include: the database itself (e.g. the conceptual schema used), the use of the database (e.g., frequency of the use of a field or combination of fields), the user's activity during interactions (e.g., stopping the database exploration to submit the same request with a change), and the knowledge the system has about the user (e.g., a novice needs to be guided through popup menus when an experienced user prefers a text window).

The main results of our study are:

- The conceptual model is a designer's view of the database. The user possesses another view and thus another conceptual model. Thus, user has to make his conceptual model compatible with his interpretation of the designer's conceptual model. The work contexts of the designer and the user are different.
- Some information is implicit in the coding of a database. For example, prices are given in French currency, the film-maker is a human. Such implicit information relies on general knowledge that is supposed shared by everybody. If made explicit, such a contextual knowledge would permit the system to complete automatically user's entry.
- An experienced user prefers to write an SQL request himself rather than in a pseudo-natural language with the help of a sophisticated tool. His reason is that the translation of the SQL request by the tool is not optimal (for instance it might count the same item several times when it appears several times in the database).
- A request may be written in different ways (e.g., using a COUNT or NOT EXISTS clause). Not all are as efficient as each other, and their efficiency depends upon the characteristics of the database. A solution would be that the system identifies rapidly what the user is writing, finds alternatives and proposes the optimal expression according to some criteria.
- To use the interaction history as an element of the context, one must find a compromise between recording enough information pieces for a useful history and recording the minimal quantity of information to avoid an overload of the computer memory.
- If there are different types of contexts, the context of the user-database interaction cannot be considered in isolation of the other types of context evoked above. There is an interdependency between the different types of context. For example, the tailoring of a user's request by the system depends on the conceptual schema of the database.

We now give an example of a rough representation of context in a rule-based formalism that concerns the different ways to write a query containing a division. The chosen example is the choice between the clauses COUNT and NOT EXISTS. Generally, the clause NOT EXISTS is preferred for a simple query. However, a query either with a COUNT clause or with a NOT EXISTS clause does not take the same amount of time to complete. The COUNT method is about 2–3 times faster than the NOT EXISTS method.

This empirical observation may be expressed as the following piece of contextual knowledge:

“Divisions with the COUNT clause are faster than with the NOT EXISTS clause,”

and its operational expression (in a rule-based expression) may be:

¹This work has been carried out with Philippe Charron during the preparation of his Master's degree at the *Institut d'Informatique d'Entreprises (Evry, France)* in 1996.

(1) General rules:

IF user sends a request,
 THEN identify pieces of the request for which alternative expressions exist.
 IF a user's request can be satisfied by several expressions,
 THEN find the expression that is the fastest.
 IF the user's request is not the fastest expression,
 THEN propose using the fastest expression.
 IF user wants an explanation,
 THEN give:

- the identified goal of the user's request,
- the list of alternatives
- the criteria chosen to classified the alternative
- the best alternative.

(2) and the specific rule for the problem at hand:

IF the user's request contains the NOT EXIST clause,
 THEN propose a faster expression with COUNT to the user.

5 Context and related elements

5.1 Introduction

This section presents some topics discussed at four of the workshops dealing directly with context. The first two workshops were held at the Computer–Human Interaction (CHI) Conference in 1989 and 1990. Reports on these workshops are given in (Maskery & Meads, 1992; Maskery et al., 1992). The two other workshops were held at the International Joint Conferences on Artificial Intelligence in 1993 and 1995 and were organized by the author. (The proceedings are available as Research Report 93/13 and 95/11 at author's laboratory, and a report on the IJCAI-93 workshop is given in Brézillon and Abu-Hakima (1995).)

The four workshops addressed partial but complementary aspects of the notion of context. The workshop at CHI'89 focused on the context of a relationship between a human and a computer-based system. The workshop at CHI'90 moved on to look at the goals and benefits of including context in applications with in perspectives the building of the tools which take advantage of context, and the building and maintenance of context. The IJCAI-93 workshop mainly addressed computational principles and mechanisms for using the contextual aspect of knowledge as well as techniques and tools for building context-sensitive systems. The IJCAI-95 workshop focused on modeling context in knowledge representation and reasoning.

Even in specific domains as CHI, a general observation is that most of the time at the workshop was spent in trying to establish what context is, how it can be defined, and how it can be distinguished from other notions such as the notion of situation. The vigorous discussions at the IJCAI workshops, which had an interdisciplinary nature, show that an agreement on the notion of context is not yet achieved. Hereafter, we present the results of the two IJCAI workshops, the results of the CHI workshop having been discussed in section 2.3.

5.2 The two sides of context

The lack of consensus appears when one considers the nature of context that is considered either as static or dynamic, discrete or continuous, knowledge or process. An important question is: Is context known *a priori* or *a posteriori*? Considering context known *a priori* supposes that it may be modeled in a discrete representation and is static. Conversely, considering context known *a posteriori* implies that context is dynamic and can be modeled only during problem solving (or interaction). The two diverging positions arise from the consideration of context according either to the use of context or its formalization.

The notion of context is interpreted differently from a *cognitive science* and an *engineering* (or *system building*) point of view. One might consider this to be the practical viewpoint versus the theoretical one. The cognitive science view is that context is used to model interactions and situations in a world of infinite breadth, and human behavior is key in extracting a model. The engineering view is that context is useful in representing and reasoning about a restricted state space within which a problem can be solved. On closer examination, one realizes that the engineering view is subsumed by the cognitive science view (Brézillon & Abu-Hakima, 1995). Thus, participants from the different disciplines face similar problems in defining and using context and can share ideas in researching a solution. In identifying the two points of view, the participants of the IJCAI-93 workshop permitted us to go one step further than the two CHI workshops. Once such a distinction is made on the viewpoint, one can achieve a kind of consensus on the aspects of context.

One must account for both the *static aspect* (knowledge that remains constant throughout the interaction) and the *dynamic aspect* (knowledge that changes throughout the interaction) of context. The changing knowledge of a context and the movement between contexts would be managed by independent but related mechanisms.

From the engineering viewpoint, context is static and considered at the level of the knowledge representation. As a consequence, there is a discrete number of contexts and the interest is in the management of contexts (e.g., see the lifting and bridging rules, and the algebra on contexts above). Static contextual knowledge is attached to the domain knowledge, and thus may be described in knowledge bases. The static part of the context is what may be coded at the design time or at the beginning of a session (e.g., the file ‘.profile’ under UNIX). With its dynamic aspect, part of the problem is linked to the changing nature of context in time. For example, the context of a problem-solving episode evolves from one step of the problem-solving to the following one when new elements enter the context. Similar results can be found in Kokinov (1995) for the cognitive science viewpoint. If it is (relatively) easy to represent the static aspect of context, the dynamic aspects of context must be considered during its use, say, a problem solving.

Contextual knowledge acts as a filter that defines, at a given time, what knowledge pieces must be taken into account (*explicit context*) from those that are not necessary or already shared (*implicit context*). A context is a structure, a frame of reference, that makes it possible not to say all the things in a story. For example, “At his birthday party, Paul blew out the candles.” It is not said here there was a birthday cake because it is clear for everybody. Such a piece of knowledge is supposed to be a part of our social inheritance. There is a French movie called ‘Le Chat’ (the cat) presenting the life of a husband and wife who have lived together for 40 years. Knowing each other very well, they have been able to severely prune their spoken conversation. For instance, with a small movement of his chin towards the cupboard, the husband says “Darling, can you please give me the salt which is in the cupboard.” With a computer system, however, it is necessary to find a compromise between the need to store a large number of information pieces and the need to have a tailored presentation of the answer to the user’s question.

In the SEPT application (Brézillon, 1990), the knowledge engineer discovered very late that some equipment in a substation (e.g. a protective relay) only functions when there is a fault on the network. It was so evident to the expert that he had never expressed this contextual information until a conflict arose with the knowledge engineer. The expert had compiled this knowledge so that for him it was implicit when explaining (making explicit) how a piece of equipment functioned. Other implicit knowledge in the SEPT application is that protective relays effectively identify the faults, transmission cables correctly transmit signals from equipment to the central printer, and faults detected in one substation occur in that substation rather than any adjacent ones.

Sharing information supposes that people share knowledge. However, it is frequent that the context—the frame of reference—must be made explicit to the receiver. For example, I may say to a person: “I heard a lion roar in my office this morning” (called hereafter CK_o, CK standing for Chunk of Knowledge). It is self-explanatory for a person that knows me. If the person is surprised (i.e., the person has some trouble with CK_o and may suppose that I am speaking of my boss), I must make explicit different pieces of knowledge in order to share them with the other: “I work in a

university near a zoo that I can see from the window of my office. There are lions in the part of the zoo that is near the University. I often hear lions roar. It just was the case this morning.” Here, CK_o is introduced after various CKs are first presented. The former person knows these pieces of knowledge (that I work in a university near a zoo, that I can see the zoo from my office and that there are lions in the zoo). However, I must make clear the implicit knowledge and share it with the person in order to explain my first statement.

Once part of the context of an interaction context only contains shared pieces of knowledge, that is pieces of knowledge which are structured in a well-defined way, these shared pieces of knowledge are compiled into a single piece of knowledge by both participants. That knowledge piece then will be recalled similarly to a pointer.

Contextualized knowledge is knowledge that is explicitly considered in problem solving. Contextual knowledge is knowledge which is used implicitly in problem solving, often in the form of constraints. Consider the two following examples.

- **Example 1.** Operators who ensure the monitoring of the distribution of water in Paris had noted that there was a peak in the water consumption each late evening. The peak was reproducible every day but not predictable because it did not occur at exactly at the same time. After some enquiry, the operators discovered that people use water for domestic needs (to drink a glass of water, to wash dishes, to pour water on flowers, go to the toilet, etc.) during the advertisements in the middle of the TV movie. The timing of advertisements in the movie depends on the plot of the movie. Such knowledge (the link between the peak of consumption and the advertisements at the TV) has a contextual nature for water distribution.
- **Example 2.** A similar situation had been noted for the distribution of electricity. The variations of the electricity consumption have been well studied, the peaks of consumption well identified and can be planned in advance. The main peak corresponds to the time during which people go back home, prepare a meal, and so on. That contextual knowledge has been integrated in the planning of electricity distribution systems and became contextualized knowledge. This is so well known that the European power companies coordinate their electricity production according to the different timing of the peaks in each European country in order to import electricity and export their electricity production.

In both examples, contextual knowledge is used to constrain some form of problem solving – planning water and electricity distribution respectively. In the first example, it is difficult to account for the relevant contextual knowledge since the time of advertisements on TV is rather unpredictable because advertisements are not exactly at the same time every evening, even if the TV schedule itself is rather repetitive. The information is thus difficult to represent and use on a computer. As a result, one needs to look for this contextual knowledge when it is needed in planning water distribution. Such contextual knowledge is dynamic because it is only when the context is modified (when the times of the advertisements change) that something must be done about the water distribution. In the second example, the contextual knowledge—that people going back home affects electricity requirements—is accurately predictable. Such contextual knowledge can be used by the electricity producers in planning the electricity supply. Thus, the contextual knowledge becomes operational (since it is used in the programs which predict electricity consumption) and hence evolves into contextualized knowledge.

When contextual knowledge can be well identified by empirical means, it is possible to include it in the problem solving. In the examples above, the contextual knowledge acts as a constraint. Conversely, a piece of knowledge obtained at one step of the problem solving—a piece of contextualized knowledge—may become a contextual knowledge at some later time. For example, identifying the user of a piece of software restricts the functionality of the software to a level appropriate for that user after the identification has taken place. Such knowledge is first in the focus of attention (contextualized knowledge) and afterwards acts as a filter on what the user may do (contextual knowledge).

This shows that part of the feedback from a user may be encoded by the designer because the

improvements requested by the user are general, repetitive and static. However, there will always be a part of the feedbacks which is specific, depending on aspects such as the task at hand and the conditions in which the task is accomplished. For example, not all users wish to have the same type of warning, whether an alert window, a sound, a special icon, or any combination of them, and this preference may change as the user becomes more familiar with the software. It is therefore desirable for the system to tailor its behavior according to the user's reaction, while the software is in use. In order to do this, the system must manage some contextual information on the interaction.

Successive versions of a piece of software take into account feedback from users and show the importance of a user-centered approach. However, what may please some users will not please others. For example, the Macintosh Operating System 7.5 provides three levels of support for a user who wants to open an item which were not present in previous versions of the operating system. These are: (1) a folder containing the last documents open regardless of the application; (2) a folder containing the last applications opened; and (3) a folder containing the last servers visited. This improvement implies some assumptions about the way in which users work. When one opens a file, say a text, in a given personal folder and one wishes to transfer information from that file to another one in the same folder, one is obliged to follow the path from the root recent document folder to the other document. This is a change from previous versions of the operating system which started from the current folder.

5.3 Applying context

There is a consensus that *context is inseparable from its use*. Context is considered as a shared knowledge space that is explored and exploited by participants in an interaction. Such shared knowledge includes the history of all that ensued over a period of time; the overall state of knowledge of the participating agents at a given moment; the small set of things they are attending to at that particular moment. Other elements present in a context come from the domain (e.g., problem solving, task at hand, events, instantiated objects and constraints, the knowledge inferred by the system), the users (e.g., goals, expertise, beliefs, learner's profile, value assignments), their environment (e.g., organizational knowledge, corporate memory), their interaction with a system (e.g., transaction history, plans for the future, attention-focusing information). However, context lacks a recognizable unifying characteristic and is often the generalization of an infinite and only partially known collection of assumptions (McCarthy, 1993).

Context makes it possible to guide the focus of attention, that is the subset of common ground that is pertinent to the current task. The focus of attention is defined as the immediate context, whilst the common ground (or common context) is seen as being the mutual context that already exists between the agents (Maskery & Meads, 1992). An interesting point is that one can change of context to solve a problem that is not obvious in the current context.

One of the strong statements made by Giunchiglia at the IJCAI-95 workshop was that "contextual reasoning is local reasoning". This stance is compatible with various communications in a number of domains (nonmonotonic reasoning, robotics, pattern recognition). Expressing reasoning through context use may allow the exploitation of various forms of reasoning within distinct contexts, such as nonmonotonic reasoning, reasoning about situations, approximate reasoning, etc. With separate contexts, reasoning in one context may influence reasoning in other contexts (Giunchiglia, 1993).

Entering (and exiting) a context serves two purposes. The import rules allow a fact about an object to be added to the object's description and allow a simulator to import messages into a description. The export rules allow encapsulated information to be taken outside the context of an object. This makes it possible to provide focus to speed up problem solving behavior, and to give a context for the interactions with the system (Guha, 1991). Then the new information could be combined with the previous description to form the updated description (Sowa, 1992).

The general approach for creating a context is to capture local contexts and generalize them (Boy,

1991b). This makes it possible to lift the relevant axioms from local theories in the new context, enter this context, and solve the problem. There are various ways of getting new contexts from old ones: by specializing the time or place, by specializing the situation, by making abbreviations, by specializing the subject matter, and by making assumptions.

Creating a context from existing contexts, it is possible to establish a hierarchy of contexts in which a formula relating two contexts involving contextual assumptions is itself in a context (Guha, 1991). The interest of a context hierarchy is that, working on an object in one context, something may be derived about that object in another context. The two contexts may use different vocabularies, and the treatment of the object may be easier in one context than in another.

It is necessary to find a compromise between the cost and the utility of a context: context analysis must be less time-consuming than the improvement it brings (Desvignes et al., 1991); adding a new context to an existing conceptual graph requires major reorganization of the position orientation of existing contexts (Elkund, 1989); and it may have a potential combinatorial explosion of contexts (Turner, 1993). However, this assumption, which permits interesting theoretical developments, often does not hold in practical situations because if we know that contexts exist, we cannot formalize them, mainly because they recursively refer to other existing contexts. When dealing with a real-world problem, we are obliged to adopt the opposite approach, identifying the current context directly in an empirical way, and then trying to establish links with known contexts.

McCarthy (1993) says that for using knowledge across contexts, one needs a process of *decontextualization*, permitting one to abstract a piece of knowledge from contexts into a more general context that covers the initial contexts. Conversely, Edmondson and Meech (1993) suggest that the concept of “context” would be most preferably understood as a process of *contextualization*. Information is the process of contextualizing data, and this contextualization process involves both the immediate data, the history of data (e.g., what has preceded a word in a text or an utterance, or visual information in the case of a sign), and the knowledge already possessed by the recipient (i.e., mental models, general knowledge, etc.). Context then may typically be viewed as the environment of communication which enables the intended meaning to be ascribed by the recipient of the data. They give the example of a pilot in a cockpit, facing hundreds of instruments. Few data provided by the instruments are transformed (interpreted) by the pilot into information, and which data is used depends upon the current context (for example, a flight in normal conditions requires less information than entering a turbulence zone).

However, although empirical evidence clearly shows that recall is much better in contexts which are similar to the context in which the information was acquired, recall of decontextualised information is possible and even desirable in problem solving. Sandberg and Wielinga (1991) give the example of Newton’s law ($F = ma$) as one which one recalls in its abstract expression rather than in all the situations in which one has applied it.

6 Discussion

Context plays an important role in all domains which involve reasoning and interpretation, and can only be captured through experience. This implies that there is no clear and general definition of context. Context seems to possess a dual nature: it is static or dynamic, discrete or continuous, form of knowledge or a process. This apparent dual nature arises from the fact that the notion of context is dependent upon its interpretation, and this can be from a cognitive science or an engineering (or system building) point of view (Brézillon & Abu-Hakima, 1995). This explains why there is a theory-versus-practice gap, and why it seems difficult to attempt to unify the various notions of context as long as a consensus is not reached. As a consequence, one considers context as a concept with complex topology, an ontology, a shared space of knowledge, a consistent set of propositional assumptions, a semantic background, the environment of communication, a set of restrictions that limit the access to parts of a system, and so on.

Focusing on human-machine problem solving, it appears that the acceptance of computer

systems depends heavily on users, the tailoring of systems to users, the system intervention between the user and the task at hand. This position is quite different from the designer's one, which is yet more usual. Thus, there are several types of context that interact. For instance, McCarthy (1993) considers that there is a common context above all of the contexts which occur during a discussion, into which all terms and predicates can be lifted. We consider that such a description may be presented using the onion metaphor, in which contextual knowledge is organized in layers as the onion skins around contextualized knowledge as the heart of the onion (Agabra et al., 1997; Brezillon et al., 1997). Along a different axis, Giunchiglia (1993) considers a set of discrete contexts (at the same level) and the relationships between them (through bridge rules for entering and leaving contexts). Both of these approaches consider context at the representation level.

McCarthy (1993) points out that the logical machinery is only a small fraction of the effort involved in building a context-based system. The bulk of the effort lies in writing the axioms describing and interrelating contexts. The problem is augmented when we consider that there are different types of interdependent contexts.

A solution for ensuring a correct transfer of information from one context to another is the context manager. A context manager makes the interpretations (or reasonings) in the export and import contexts compatible with one another. It is supposed to: retain as much of the knowledge generated as possible; provide easy access to and a good explanation of this knowledge; make the best use of the knowledge already held in the dynamic knowledge base to enable it to generate new knowledge without performing redundant inference; and help the user compare different, sometimes conflicting solutions. For example, a good context manager would make compatible users' requests and the conceptual schema of a database. However, a context manager acts at the level of the presentation of the knowledge than its representation (or its modelling). This is why one often gives at context the role of filter at the programming level. The double action of a context manager on knowledge at a given step of a problem solving is: (i) to select the knowledge pieces for the focus of attention, and (ii) to keep in stand-by other knowledge pieces.

The situation is different at the human-machine interaction level where different contexts coexist: the user's context, the context of the task at hand, the system context, and the context of the interaction. Indeed, one result of the discussion at the various scientific events is that context cannot be dissociated from its use. One reason is that one doesn't know what is really needed until one is in the design process. A consequence is that context makes it possible to make the reasoning local. Thus, we are immediately in the realm of knowledge-based system, and more recently of intelligent assistant systems that abandon the first paradigm of early expert systems, namely the separation of the knowledge representation from its use.

Contexts define when the knowledge should be considered. They make it possible to structure knowledge bases in trackable units, often organized in a hierarchy. A context contains: (i) sets of concepts (also called schemas, frames, or structures) that describe the basic terms used to encode knowledge in the ontology, and (ii) a set of constraints that restrict the manner in which instances of these concepts may be created and combined. Context-encapsulated knowledge appears as a chunk of reasoning. This is similar to the claim that acquiring knowledge with its context of use, leads to the organization of knowledge representation in episodes (Boy, 1991a; Turner, 1993; Lamming & Flynn, 1994). Thus, contexts simplify the construction of the knowledge base by imposing requirements on the representation language. A challenge here is how knowledge in its context of use may be examined in other contexts (decontextualization of the knowledge).

A contextual system may have two types of memory: a long-term memory (a primary database and a base of contexts) and a short-term memory (intracontext knowledge processing and intercontext knowledge processing). A context may also be generated dynamically and, according to McCarthy, created from old contexts. The difficulty here is to determine if one needs to store all past contexts or, as proposed by Turner (1993), have a set of "elementary" contexts that may be combined to constitute complex contexts to adequately represent a particular situation.

The relationships between context and knowledge are yet to be explored. A piece of knowledge may

be contextual or contextualized according to the stage in the problem solving process. Contextualized knowledge is knowledge that is explicitly considered in the problem solving. Thus, to some extent, contextualized knowledge is operational knowledge. Contextual knowledge intervenes implicitly in the problem solving, most often as constraints. However, a piece of knowledge obtained at one step of the problem solving—a piece of contextualized knowledge—may be a contextual knowledge at a following step. An operational definition of context is needed.

In the formula $ist(C\ P)$, McCarthy (1993) defines context to capture all things that are not explicit in P but that are required to make P a meaningful statement representing what it is intended to state. In other words, a context is a structure, a frame of reference, that makes it possible to omit details from a story. Context makes it possible to leave implicit things that do not intervene directly in the problem solving. This is coherent with the infinite dimension of context advocated by McCarthy too. Such a definition may be easily extended from formal logic using the following claim:

“Context is what constrains problem solving without intervening in it explicitly.”

The claim is left vague here because we think that before we refine the definition of context, we need to position context at the level of the knowledge and its representation, or at the level of the reasoning mechanism, or at the level of the human-machine interaction. We already point out that there is a different model of context at each level. Moreover, all these types of contexts are dependent on the others. For example, the way in which a user defines his request (concerning the interaction context) depends on the conceptual schema of the interrogated database (context at the level of the knowledge representation). In his book *Lector in Fabula*, Umberto Eco (1985) discusses in depth how the writer uses a model of a reader and how the writer may lead the reader into a *cul-de-sac*. This again illustrates the importance of defining context, and the context in which context is defined.

The discussion continues. Some of the questions that must be addressed are: Does part of the context belong to the knowledge base or a particular context base? What are the relationships between context and meta-knowledge? Knowledge representation? Time? What are the relationships between context, tactical decision and strategical decision? What are the relationships between contextualization process and control knowledge?

Acknowledgement

The author would like to thank Simon Parsons for his help in preparing the final version of this paper for publication.

References

- Abu-Hakima, S, 1993. “The use of context in diagnostic systems” *Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context, Research Report 93/13 LAFORIA*, University Paris 6, 13–20.
- Aikins, JS, 1983. “Prototypical knowledge for expert systems” *Artificial Intelligence* **20**(2) 163–210.
- Agabra, J, Alvarez, I and 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, Brasil, Federal University of Rio de Janeiro, 351–362.
- Arbab, B, 1992. “A formal language for representation of knowledge” *Proceedings of the AAAI’92 Workshop on Propositional Knowledge Representation* Stanford, CA, 1–8.
- Barewise, J and Perry, J, 1983. *Situations and Attitudes* MIT Press.
- Barr, A and Feigenbaum, EA, editors, 1981. “Representation of knowledge, representation” *The Handbook of Artificial Intelligence Vol. I* William Kaufmann.
- Barthe, M, 1991. “ERGO-METH: Principes d’une méthodologie d’informatisation visant à intégrer les apports de l’ergonomie cognitive dans la démarche de conception pour améliorer l’utilité et la maniabilité des logiciels interactifs de gestion” *Mémoire d’ingénieurs du CNAM* Paris, France.
- Bastien, C, 1992. “Le décalage entre logique et connaissances” *Le Courrier du CNRS*, Numéro Spécial “Sciences Cognitives”, **79** 38.

- Bianco, L, Dell'omo, P and Ezhkova, IV, 1994. "Application of contextual technology for supporting decision making in transportation" *Proceedings of the 7th IFAC/IFORS Symposium on Transportation Systems* Tianjin, China, 363–368.
- Bloom, C, Bullemer, P, Chu, R and Villano, M, 1992. "A task-driven approach to knowledge acquisition, analysis and representation for intelligent training systems" *Proceedings of the International Conference on Systems, Man, and Cybernetics* Chicago, USA, 1, 509–514.
- Bobick, AF and Pinhanez, C, 1995. "Using Approximate Models as Source of Contextual Information for Vision Processing" *Proceedings of the IEEE Workshop on Context-Based Vision (CBVIS '95)*.
- Boy, G, 1991a. *Intelligent Assistant Systems* Academic Press,.
- Boy, G, 1991b. "Computer integrated documentation" *NASA Technical Memorandum 103870*, NASA Ames Research Center, Moffett Field, CA 94035–1000, USA.
- Boy, G, 1995. "Design rationale and supportability: An agent-oriented corporate memory approach" *Proceedings of the International Workshop on the Design of Cooperative Systems*.
- Brémond, F and Thonnat, M, 1997. "Issues in representing context illustrated by scene interpretation applications" *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT-97* Rio de Janeiro, Federal University Publisher, 323–332.
- Brézillon, P, 1990. "Interpretation and rule packet in expert systems. Application to the SEPT expert system" In: S Ramani, R Chandrasekar and KSR Anjaneyulu, editors, *Lecture Notes in Artificial Intelligence 444: Knowledge Based Computer Systems* Springer-Verlag, 78–87.
- Brézillon, P, 1991. "METAL: a language for structured knowledge-based systems" *Proceedings of the IJCAI-91 Workshop Software Engineering for Knowledge-Based Systems* Sydney, Australia, 11–22.
- Brézillon, P, 1994. "Context needs in cooperative building of explanations" *Proceedings of the First European Conference on Cognitive Science in Industry* Luxembourg, 443–450.
- Brézillon, P and Abu-Hakima, S, 1995. "Using Knowledge in its context: Report on the IJCAI-93 Workshop" *AI Magazine* 16(1): 87–91.
- Brézillon, P and Cases, E, 1995. "Cooperating for assisting intelligently operators" *Proceedings of the International Workshop on the Design of Cooperative Systems* INRIA, 370–384.
- Brézillon, P and Pomerol, J-Ch, 1996a. "Misuse and nonuse of knowledge-based systems: The past experiences revisited" In: P Humphreys, L Bannon, A McCosh, P Migliarese and J-Ch Pomerol, editors, *Implementing Systems for Supporting Management Decisions* Chapman & Hall, 44–60.
- Brézillon, P and Pomerol, J-Ch, 1996b. "User acceptance of interactive systems: Lessons from knowledge-based and decision support" *International Journal of Failure & Lessons Learned in Information Technology Management* (to appear).
- Brézillon, P, Bau, DY, Fauquembergue, P, Hertz, A and Maizener, A, 1990. "Elaboration of the SEPT expert system as a coupling of a simulator and a diagnostician" *Proceedings of the Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* Charleston, SC, 54–60.
- Brézillon, P, Gentile, C, Saker, I and Secron, M, 1997. "SART: A system for supporting operators with contextual knowledge" *First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)* Rio de Janeiro, Brasil, Federal University of Rio de Janeiro, 209–222.
- Burlina, P, Chellappa, R, Lin, CL and Zhang, X 1995. "Context-Based Exploitation of Aerial Imagery", *Proceedings of the IEEE Workshop on Context-Based Vision (CBVIS '95)*.
- Cahour, B and Karsenty, L, 1993. "Context of dialogue: a cognitive point of view" *Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Technical Report 93/13 LAFORIA*, University Paris 6, France, 20–29.
- Carenini, G and Moore, JD, 1993. "Generating explanations in context" *International Workshop on Intelligent User Interfaces* Orlando, FL.
- Cavalcanti, M, 1993. "Les mondes possibles dans les systèmes de production: un métalangage pour la gestion d'hypothèses et le raisonnement non-monotone" *PhD Thesis*, LRI, Université Paris-Sud, France.
- Chandrasekaran, B, Johnson, TR and Smith, JW, 1992. "Task-structure analysis for knowledge modeling" *Communications of the ACM* 35(9) 124–137.
- Clancey, WJ, 1983. "The epistemology of a rule-based expert system: A framework for explanation" *Artificial Intelligence Journal* 20(3) 197–204.
- Clancey, WJ, 1991. "Israel Rosenfield, The Invention of Memory: A New View of the Brain (Book Review)" *Artificial Intelligence* 50 241–284.
- Clancey, WJ, 1993. "Notes on 'Epistemology of a rule-based expert system'" *Artificial Intelligence Journal* 59 197–204.
- Compton, P and Jansen, B, 1988. "Knowledge in context: A strategy for expert system maintenance" In: J Siekmann, editor, *Lecture Notes in Artificial Intelligence 406* Springer-Verlag.
- Compton, P and Jansen, R, 1990. "A philosophical basis for knowledge acquisition" *Knowledge Acquisition* 2 241–257.

- Compton, P, Yang, W, Lee, M and Jansen, B, 1991. "Cornerstone cases in a dictionary approach to rule maintenance" *Proceedings of the IJCAI'91 Workshop on Software Engineering for Knowledge-Based Systems* 24–40.
- de Kleer, J, 1987. "An assumption based truth maintenance system" In: M Ginsberg, editor, *Nonmonotonic Reasoning* Morgan Kaufmann.
- Desvignes, M, Porquet, C and Spagnou, P, 1989. "A tool for studying context in image sequences" *Proceedings of the 3rd IEE International Conference on Image Processing and Applications* 467–472.
- Desvignes, M, Revenu, M and Porquet, C, 1991. "The use of context in image sequences interpretation" *Proceedings of the 8th Congrès AFCET-RFIA* 55–61.
- Eco, U, 1985. *Lector in Fabula* Grasset, Paris.
- Edmondson, WH and Meech, JF, 1993. "A model of context for human-computer interaction" *Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context, Technical Report 93/13 LAFORIA*, University Paris 6, 31–38.
- Eklund, P, 1989. "Prospects for conceptual graphs in acquisition interfaces" *Proceedings of EKAW89* Paris, France, 169–179.
- Ezhkova, IV, 1989. "Knowledge formation through context formalization" *Computers and Artificial Intelligence* 8(4) 305–322.
- Ezhkova, IV, 1992. "Contextual systems: Is it a way of a universal expert system development?" *General Systems*, G. Klir, NJ.
- Farquhar, A, Dappert, J, Fikes, R and Pratt, W, 1995. "Integrating information sources using context logic" *Technical Report KSL-95-12* Knowledge Systems Laboratories, Stanford University.
- Fischer, G, 1990. "Communication requirements for cooperative problem solving systems" *Information Systems* 15(1) 21–36.
- Flor, N and Hutchins, E, 1991. "Analyzing distributed cognition in software teams: A case study of team programming during perfective software maintenance" In: J Koenemann-Belliveau et al., editors, *Proceedings of the Fourth Annual Workshop on Empirical Studies of Programmers Ablex*, 36–59.
- Forman, AV, Rowland PJ and Pemberton, WG, 1984. "Contextual analysis of tactical scene" *SPIE* 485, *Applications of Artificial Intelligence* 189–197.
- Frege, G, 1985. "On sense and meaning" In: AP Martinich, editor, *The Philosophy of Language* Oxford University Press, 212–220. (Originally published in 1892.)
- Garfinkel, A, 1981. *Forms of Explanation: Rethinking the Questions in Social Theory* Yale University Press.
- Giroto, V, 1991. "Reasoning on deontic rules: the pragmatic schemas approach" *Intellectica* 1 15–52.
- Giunchiglia, F, 1993. "Contextual reasoning" *Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context, Research Report 93/13 LAFORIA*, 39–48.
- Giunchiglia, F and Ghidini, C, 1987. "A local model semantics for propositional attitudes" *Proceedings of CONTEXT-97* Federal University of Rio de Janeiro, 363–372.
- Goh, CH, Madnick, SE and Siegel, MD, 1994. "Context interchange: Overcoming the challenges of large-scale interoperable database systems in a dynamic environment" *Proceedings of the Third International Conference on Information and Knowledge Management* Gaithersburg, MD, 337–346.
- Grant, AS, 1992. "Mental models and everyday activities" *Proceedings of the 2nd Interdisciplinary Workshop on Mental Models* Cambridge, UK, 94–102.
- Grant, AS, 1994. "Modeling complex cognition: contextual modularity and transitions" *Proceedings of the Fourth Int. Conf. on User Modeling* Hyannis, USA, The MITRE Corp., 157–162.
- Gruber, T, 1991. "Justification-based knowledge acquisition" In: H Motoda, R Mizoguchi, J Boose and B Gaines, editors, *Knowledge Acquisition for Knowledge-Based Systems* IOS Press.
- Grusenmeyer, C, 1995. "Les dialogues coopératifs en phase de relève de poste: Rôle dans la sûreté des systèmes de production" *Proceedings of the XXX^e Congrès de la SELF "L'homme dans les nouvelles organisations"* Biarritz, France.
- Guha, RV, 1991. "Contexts: a formalization and some applications" *MCC Technical Report ACT-CYC-423-91*.
- Guha, RV, 1993. "Context dependence of representations in CYC" *Colloque ICO'93* Montréal,
- Hatchuel, A and Weil, B, 1992. "L'Expert et le Système" *Economica* Paris.
- Hendrix, G, 1975. "Expanding the utility of semantic networks through partitioning" *Proceedings of the Fourth IJCAI* 115–121.
- Henninger, S, 1992. "The knowledge acquisition trap" *Proceedings of the IEEE Workshop on Applying Artificial Intelligence to Software Problems: Assessing Promises and Pitfalls (CAIA-92)* Monterey, CA, 51–57.
- Hovy, EH, 1990. "Pragmatics and natural language generation" *Artificial Intelligence* 43(2) 153–198.
- Huhns, MN, Jacobs, N, Ksiezyk, T, Shen, W-M, Singh, MP and Cannata, PE, 1993. "Enterprise information modeling and model integration in Carnot" *Proceedings of the International Conference on Intelligent and Cooperative Information Systems* Rotterdam, Netherlands, 12–14.

- Huuskonen, P and Korteniemi, A, 1992. "Explanation based on contexts" *Proceedings of the 8th Conference on Artificial Intelligence for Applications* Monterey, CA, 179–185.
- Intille, SS and Bobick, AF, 1995. "Exploiting Contextual Information for Tracking by Using Closed-Worlds" *Proceedings of the IEEE Workshop on Context-Based Vision (CBVIS '95)*.
- Iwanska, L, 1995. "Summary of the IJCAI-95 Workshop on Context in Natural Language Processing" <http://www.cs.wayne.edu/context>.
- Jaeger, H, 1993. "A type-free semantics for concepts in contexts" *IJCAI-93 Workshop on "Using Knowledge In Its Context"*, Research Report 93/13 LAFORIA, University Paris 6, France.
- Jansen, B, 1993. "Context: A real problem for large and sharable knowledge bases" *Proceedings of the 1st International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KB&KS'93)* Tokyo, 177–183.
- Jansen, B, 1995. "Context in context" <http://mac145.syd.dit.csiro.au/Context/context.html>. (Working Draft V4).
- Jansen, B and Grosz, G, 1990. "The knowledge dictionary: representing contextual information" *Technical Report TR-FD-90-4* CSIRO Division of Information Technology, PO Box 1599, North Ryde, NSW, Australia.
- Jansen, B and Bray, G, 1992. "IntelliText" *ECHT'92*.
- Johnson, H and Johnson, P, 1992. "Acceptable explanations: theory before implementation" *Proceedings of the AAAI Spring Symposium on Producing Cooperative Explanation* Stanford, CA, 95–102.
- Kamp, H, 1981. "A theory of truth and semantic representation" In: JAG Groenendijk, TMV Janssen and MJB Stokhof, editors, *Formal Methods in the Study of Language* Mathematical Centre Tracts, Amsterdam, 277–322.
- Karsenty, L, 1994. "L'explication d'une solution dans les dialogues de conception" *PhD Thesis* University Paris 8.
- Karsenty, L and Falzon, P, 1992. "Spontaneous explanations in cooperative dialogues" *Proceedings of the ECAI'92 Workshop on Improving the Use of KBS with Explanation*, Technical Report 92/21 LAFORIA, University Paris 6, France, 115–124.
- Karsenty, L and Brézillon, P, 1995. "Cooperative problem solving and explanation" *International Journal of Expert Systems With Applications* 8(4) 445–462.
- Kjeldsen, R and Kender, J, 1995. "Knowledge Based Hand Gesture Recognition" *Proceedings of the IEEE Workshop on Context-Based Vision (CBVIS '95)*.
- Kokinov, B, 1995. "A dynamic approach to context modeling" Working Notes of the IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning. In: P Brézillon and S Abu-Hakima, editors, *Research Report 95/11*, LAFORIA, University Paris 6, 199–209.
- Laird, JE, Newell, A and Rosenbloom, PS, 1987. "Soar: An architecture for general intelligence" *Artificial Intelligence* 33 1–64.
- Lamming, M and Flynn, M, 1994. "'Forget-Me-Not', Intimate computing in support human memory" *Proceedings of FRIEND21: International Symposium on Next Generation Human Interface* Japan.
- Lashkari, Y, Metral, M and Maes, P, 1994. "Collaborative Interface Agents" *Proceedings of the AAAI'94 Conference* Seattle, WA.
- Lave, J, 1988. *Cognition in Practice* Cambridge University Press.
- Leake, DB, 1992. *Evaluating Explanations* Lawrence Erlbaum.
- Leong, T-Y, 1992. "Representing context-sensitive knowledge in a network formalism: a preliminary report" In: D Dubois, MP Wellman, MD D'Ambrosio and P Smets, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference* Morgan Kaufmann, 166–173.
- Leont'ev, A, 1978. "The problem of activity in psychology" *Soviet Psychology* 13(2) 4–33.
- Lester, JC and Porter, BW, 1991. "Generating context-sensitive explanations in interactive knowledge-based systems" *Proceedings of the AAAI'91 Workshop on Comparative Analysis of Explanation Planning Architectures* 27–41.
- Mackie, J, 1965. "Causes and conditions" *American Philosophical Quarterly* 2(4) 245–264.
- Mark, B, 1988. "Explanation and interactive knowledge acquisition" *Proceedings of AAAI'88, Workshop on Explanation*.
- Maskery, H and Meads, J, 1992. "Context: In the eyes of users and in computer systems" *SIGCHI Bulletin* 24(2) 12–21.
- Maskery, H, Hopkins, G and Dudley, T 1992. "Context: What does it mean to application design", *SIGCHI Bulletin*, 24(2): 22–30.
- Maurer, F, 1992. "Knowledge base maintenance and consistency checking in MOLKE/HyDi" *Proceedings of the EKA'92: Lecture Notes in Artificial Intelligence* 599 Springer-Verlag, 337–352.
- McCarthy, J, 1979. "First-order theories of individual concepts and propositions" In: JE Hayes, D Michie and LI Mikulich, editors, *Machine Intelligence* 9 Ellis Horwood, 129–147.
- McCarthy, J, 1993. "Notes on formalizing context" *Proceedings of the 13th IJCAI Vol 1*, 555–560.

- McDermott, J, 1982. "R1: A rule based configurer of computer systems" *Artificial Intelligence* **19** 39–88.
- Minsky, M, 1986. *The Society of Mind* Picador.
- Mittal, VO and Paris, CL, 1995. "Use of context in explanations systems" *International Journal of Expert Systems with Applications* **8**(4) 491–504.
- Mohr, R and Masini, G, 1979. "Utilisation du contexte en analyse d'images" *Rapport ADI 19/79* CRIN, France.
- Moore, JD, 1995. *Participating in Explanatory Dialogues. Interpreting and Responding to Questions in Context* MIT Press.
- Mulet-Marquis, D and Gondran, M, 1985. "Un langage pour les systèmes experts: ALOUETTE" *Bulletin de la Direction des Etudes et Recherches d'EDF Série C* (3/4).
- Mylopoulos, J, Bordiga, A, Jarke, M and Koubarakis, M, 1990. "Telos: representing knowledge about information system" *ACM Trans. on Information Systems* **8**(4) 325–362.
- Nardi, BA, 1992. "Studying context: A comparison of activity theory, situated action models, and distributed cognition" *Proceedings of the East–West International Conference on Human–Computer Interaction* St.-Petersburg, 352–359.
- Newman, WM, Eldridge, MA and Lamming, MG, 1994. "PEPYS: Generating autobiographies by automatic tracking" *Research Repport* Rank Xerox.
- Ogden, CK and Richards, IA, 1946. *The Meaning of Meaning* Routledge & Kegan Paul.
- O'Hara, K, Shadbolt, P, Laublet, P, Zacklad, M and Leroux, B, 1992. "VITAL, A methodology-based workbench for KBS life cycle support" *ESPRIT II Project intermediary report VITAL/DD212* September.
- Oracle (1996) *Oracle7 TM ConText Option*, <http://www.oracle.com/>.
- Paris, CL, Wick, MR and Thompson, WB, 1988. "The line of reasoning versus the line of explanation" *Proceedings of AAAI Workshop on Explanation* 4–7.
- Park, Y-T and Wilkins, DC, 1990. "Establishing the coherence of an explanation to improve refinement of an incomplete knowledge base" *Proceedings of the AAAI'90*.
- Paton, R, Shave, M, Bench-Capon, T and Nwana, H, 1993. "Domain characterisation in context" *Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13* LAFORIA, University Paris 6, France.
- Pereira, FC and Pollack, ME, 1991. "Incremental interpretation" *Artificial Intelligence* **50**(1) 37–82.
- Politzer, G, 1991. "Introduction du numéro spécial 'Pragmatique et Psychologie du Raisonnement'" *Intellectica* **1** 7–13.
- Pomerol, J-Ch and Brézillon, P, (1997) "Contextual issues in the framework of multicriteria decision making" *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)* Federal University of Rio de Janeiro, 256–268.
- Récanati, F, 1992. "La pragmatique linguistique" *Le Courrier du CNRS Numéro Spécial "Sciences Cognitives"* **79** 21.
- Reichman, R, 1985. *Getting computers to talk like you and me. Discourse context, focus, and semantics* MIT Press.
- Sandberg, J and Wielinga, B, 1991. "How situated is cognition?" *Proceedings of the 12th IJCAI'91* Sydney, Australia, Vol 1, 341–346.
- Serafini, L and Ghidini, C, 1997. "Context-based semantics for federated databases" *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)* Federal University of Rio de Janeiro, 33–45.
- Schmidt, CT, 1995. "Information processing, context creation, setting minds in public arenas: investigative techniques for client/automaton dialogue design" *Proceedings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Technical Report 95/11* LAFORIA, University Paris 6, France, 121–131.
- Sciore, E, Siegel, M and Rosenthal, A, 1992. "Context interchange using meta-attributes" *Proceedings of the 1st International Conference on Information and Knowledge Management* 377–386.
- Sharma, N, 1995. "On formalising and reasoning with contexts" *Research Report 352* Department of Computer Science, University of Queensland, Australia.
- Sowa, JF, 1984. *Conceptual Structures: Information Processing in Mind and Machine* Addison-Wesley.
- Sowa, JF, 1991. "Toward the expressive power of natural language" *Principles of Semantic Networks—Exploration in the representation of Knowledge* Morgan Kaufmann, 157–189.
- Sowa, JF, 1992. "Representing and reasoning about contexts" *Proceedings of the AAAI'92 Workshop on Propositional Knowledge Representation* Stanford, 133–142.
- Srinivasan, A, Compton, P, Malor, R, Edwards, G and Lazarus L, 1991. "Knowledge acquisition in context for a complex domain" Pre-print of *Proceedings of the Fifth EKA'91*.
- Surav, M and Akman, V, 1995. "Modeling context with situations" *Proceedings of the IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning, Research Report 95/11* LAFORIA, 145–156.

- Suthers, D, 1993. "Influences of the epistemic context on explanation" *Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13* LAFORIA, University Paris 6, Paris, France.
- Tang, MX, 1995. "Exploring design solution using a context management system" *Proceedings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Technical Report 95/11* LAFORIA, University Paris 6, France, 157–166.
- Terssac (de), G, 1992. *Autonomie dans le travail* Série Sociologie d'Aujourd'hui, Presses Universitaires de France.
- Toussaint, GT, 1978. "The use of context in pattern recognition" *Pattern Recognition* **10** 189–204.
- Turner, RM, 1993. "Context-sensitive reasoning for autonomous agents and cooperative distributed problem solving" *Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13* LAFORIA, University Paris 6, Paris, France.
- Turner, RM, 1995. "Using contextual knowledge in autonomous real-world systems", *Proceedings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning* Montreal, *Technical Report 95/11* LAFORIA, University Paris 6, France, 189–196.
- Turner, RM, 1997. "Determining the context-dependent meaning of fuzzy subsets" *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)* Federal University of Rio de Janeiro, 233–242.
- Turney, P, 1996. "The management of context-sensitive features: A review of strategies" *Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains* 53–69.
- Vanwelkenhuysen, J and Mizoguchi, R, 1995. "Adaptation of reusable knowledge for workplace integration" *Proceedings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Technical Report 95/11* LAFORIA, University Paris 6, France, 167–177.
- Walther, E, Eriksson, H and Musen MA, 1992. "Plug-and play: Construction of task-specific expert-system shells using sharable context ontologies" *Technical Report* <http://www.stanford.edu>.
- Wick, MR and Thompson, WB, 1992. "Reconstructive expert system explanation" *Artificial Intelligence Journal* **54** 33–70.
- Widmer, G, 1996. "Recognition and exploitation of contextual clues via incremental meta-learning (Extended version)" *Proceedings of the 13th International Conference on Machine Learning (ML-96)* Morgan Kaufmann.
- Winograd, T and Flores, F, 1989. *L'intelligence Artificielle en question* Presse Universitaire de France, Ablex.
- Wu, Z, Ansgar, B and Christophe, K, 1992. "Skeletal Plans Reuse: a restricted conceptual graph approach" *Proceedings of the 7th International Workshop on Conceptual Graph* New Mexico, USA.
- Zukerman, I, 1992. "Content planning based on a model of a user's beliefs and inferences" *Proceedings of the User Modelling Conference, UM'92*.