Context in Human-Machine Problem Solving: A Survey

Patrick Brézillon

Abstract:
Context appears in AI as a challenge for the coming years as shown by the various scientific events focusing on context held since 1995. However, context is already considered in other domains, as Natural Language, although through few aspects of context. We present in this paper a survey of the literature dealing directly and explicitly with context whatever the domain is. This permits to have a clear view on the context in AI. One of the conclusions of this survey is to point out the existence of different types of context along knowledge representation, the mechanisms of reasoning on the knowledge, and the interaction of the computer system with a human.

Keywords: Context, knowledge representation, knowledge modeling, explanation, knowledge acquisition, machine learning

Résumé:
Le contexte apparaît en IA comme un des challenges des années qui viennent, comme le montrent la tenue de manifestations scientifiques sur le contexte depuis 1995. Cependant, le contexte est déjà considéré depuis longtemps dans d'autres domaines tel que le Langage Naturel, bien que sur des aspects limités du contexte. Nous présentons dans ce papier une étude de la littérature qui s'intéresse directement et explicitement au contexte quelque soit le domaine. Ceci permet d'avoir une vue plus clair sur le contexte en IA. Une des conclusions de cette étude est l'existence de différents types de contexte suivant que l'on s'intéresse aux connaissances, aux mécanismes de raisonnement exploitant ces connaissances et aux interactions homme-machine.

Mots clés: contexte, représentation des connaissances, modélisation des connaissances, explication, acquisition des connaissances, apprentissage par la machine

Research Report 96/29, LAFORIA, October 1996

Context in Human-Machine Problem Solving: A Survey

Patrick Brézillon

LAFORIA, Case 169, University Paris 6,
4 place Jussieu, 75252 Paris Cedex 05, France
Tel: +33 1 44 27 70 08 - Fax: +33 1 44 27 70 00
E-mail: brezil@laforia.ibp.fr

Abstract:
Context appears in AI as a challenge for the coming years as shown by the various scientific events focusing on context held since 1995. However, context is already considered in other domains, as Natural Language, although through few aspects of context. We present in this paper a survey of the literature dealing directly and explicitly with context whatever the domain is. This permits to have a clear view on the context in AI. One of the conclusions of this survey is to point out the existence of different types of context along knowledge representation, the mechanisms of reasoning on the knowledge, and the interaction of the computer system with a human.

Keywords: Context, knowledge representation, knowledge modeling, explanation, knowledge acquisition, machine learning

INTRODUCTION

Context plays an important role in a number of domains since a long time. This is especially true for activities as predicting context changes, explaining unanticipated events and helping to handle them, and helping to focus attention. (See [Jansen, 1995] for philosophical considerations on context.) Context also is widely used. When looking for *context* by AltaVista on the Web, one found more than 710000 pages containing *context*, when *concept* gives only 639000 references (*system* occurs in 9612000 pages). The main problem is the lack of clear definition of this word: a set of preferences and/or beliefs [Cahour and Karsenty, 1993], a window [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 [Aussenac and Dieng, 1991], buttons which are functional, customizable and shareable [Maskery and 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 and 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 and Richards, 1946], the possibility that permits to listen what is said and what is not said [Winograd and Flores, 1989].

The first works considering context explicitly are in Natural Language. For instance:

(1) Frege (1892) 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: word name (denote) other words. (In writing, quotation marks or italics are used for direct contexts.) In an indirect context, names denote their customary denotation and have an indirect sense which is different from their customary sense [Frege, 1985].

(2) Russel (1905) 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 & Holyoak (1985) 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 their 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].

More recently, one of the main conclusion of the IJCAI-95 Workshop on Context in Natural Language Processing (NLP) is that context in NLP stays rather unexplored [Iwanska, 1995]. This has serious consequences. First, there is a theory-versus-practice gap. Second, context appears to be a huge bag of issues, virtually always closely related to the specific task at hand, domain, application, and the research problems of immediate interest to authors. Third, works addressing the context problems clearly are too vague and do not address the specific questions about context. Fourth, one should not even attempt to unify the various notions of context as long as a consensus is not reached. However, there is an agreement on the fact that context provides constraints on reasoning, increases information content of natural language utterances, and facilitates learning.

In Artificial Intelligence, the lack of explicit representation of context is one of the main reasons of the failures of most of the Knowledge-Based Systems (KBSs) [Brézillon and Pomerol, 1996a and 1996b]. Studies of the use of KBSs in real-world applications permit one 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 as novice followers. However, unexpected problems to solve are the norm rather than the exception. KBSs cannot solve such unexpected problems when users, with their practical experience, are not given the opportunity. This failure is due to the lack of user-system cooperation and the context in which a problem is solved in relation to the original knowledge acquisition for the problem solving.

The second failure arises when KBSs do not correctly use their knowledge. Knowledge being 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 needed 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 and 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 give now two examples.

Schmidt (1995) gives an example of 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 a machine as automatic dispensers. Such machines impose a forced interview, a prior determination of all the interaction, and an amount of instructions that is the same for all the end-users. However, the more instructions 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 context and its evolution during interaction with the system.

Vanwelkenhuysen and Mizoguchi (1995) describe work practice of two troubleshooters--testers and engineers--performing 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 on priority on 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 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 coming operator must adjust his understanding and operational representation of the current situation with those of the leaving operator. Leaving and coming operators need to share knowledge and build common references. Functional communication increases crucially when the changing occurs during a malfunctioning 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 working places, executive humans develop working ways (strategies, relationships among them, etc.) to reach the efficiency that decision-makers wait when they design the work. Part of their practical solving of problems is not coded [Hatchuel and Weil, 1992]. Such a know-how is generally elaborated case-by-case in nonwritten rules. A nonwritten 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 followed is. The validation of such nonwritten rules is more linked to the result than to the procedure to reach it. This is a logic of the efficiency [de Terssac, 1992].

Nowadays, one notes an increasing interest for the context. In 1995, four scientific events in Artificial Intelligence dealt specifically with context, and a number of works consider the context explicitly from the level of the modelling of the domain knowledge to the level of the programming languages. The modeling, representation and use of context appears to be the challenge of the coming years, especially when we now face very complex problems, large knowledge bases and multimedia.

Hereafter, the paper is organized in the following way. Section 1 gives the needed elements on the SEPT application from which are taken the examples that are used all along this document. Section 2 presents a view of how context is used in several domains and in three applications. We give in section 3 the way in which context is modeled and represented in the logic formalism and a rule-based formalism, and present different other approaches. The section 4 is a synthesis of discussions on context held at different workshops. The section 5 sum up the different points that may be of interest for modeling effectively context.

1. THE CONTEXT OF OUR STUDY

Our study arises from a need to represent context explicitly in systems that must cooperate with a human for solving problems. The main aspects that are considered in this large framework are explanation and incremental knowledge acquisition. This may lead to present here a partial view that is amplified by the fact that the literature on context is very large and disseminated in a number of places.

We have chosen to exemplify our purposes all along the paper with a practical application, to facilitate the description of the use of context and show its practicality. This application concerns the diagnosis and explanation through simulation of the behavior of equipment components in Extra-High Voltage (EHV) substations operated in power plants using SEPT at Electricité France [Brézillon et al., 1990; Bau and Brézillon, 1992]. In this application, context is not made explicit but is represented.

The French national power system is composed of controlled systems and control systems. Elements of the controlled system are lines and busbars that constitute a network. Elements of the control system are controllers. A controller links a busbar to a line or another busbar. It contains pieces of equipment like protective relays that act in parallel and a circuit breaker. The appearance of the fault on a line or a busbar triggers the protective relay. The fault elimination stops the functioning of all the equipment pieces. A protective relay detects the appearance of a fault in the controlled system and automatically sends a trip order to the circuit breaker after a delay. Actions of the equipment are associated with signals that are sent to a central printer from which the operator diagnoses the events that occur.

SEPT is a Knowledge-Based System (KBS) that supports an operator diagnosing various types of equipment in an extra high voltage substation. The types of equipment rapidly isolate a faulty element of the power network to ensure a continuous distribution of electricity. SEPT is designed to detect functional faults in equipment in a systematic manner.

The overall objectives of SEPT are twofold: to provide a diagnosis quickly; and to *a posteriori* explain the diagnostic results, mainly by simulation. SEPT diagnoses equipment in three successive steps: (1) The identification of abnormal behaviors of equipment; (2) The diagnosis of fault location in the power system; and (3) The inter-relations between abnormal equipment behaviors and fault location. The diagnosis in SEPT is an exhaustive analysis of all the types of equipment that is different from the diagnosis made by the operator.

The METAL language in which SEPT is described, is based on first-order predicate logic within a formalism using rule packets and the notion of interpretation [Brézillon 90]. The system has approximately 800 rules partitioned across 150 rule packets. Rule packets explicitly represent the structural and functional organization of the domain knowledge and of the diagnosis expertise. They are managed dynamically, and a rule packet is called in the action part of a rule and some instances of variables in the antecedent of such a meta-rule may be set in the rule packet. At any time of the reasoning, the system only considers the relevant rules at the current step of the problem solving, and thus the system reasoning is made local.

SEPT offers the user two mechanisms through which to explore the diagnostic tree, namely interruption points in the reasoning and basic interpretations to control the execution of actions when a rule is fired [Brézillon, 1990]. The user may either start or inhibit the various points of interruption mechanism to suspend the reasoning and look, with a combination of basic interpretations, for various events at various levels of detail on different slices of a reasoning branch.

The first explanations provided by SEPT to operators were a failure because of the strategy of attaching canned text to individual rules. Clancey has pointed out the problems of such poor explanation mechanisms that simply attach text to rules [Clancey 86]. Operators do not want definition of equipment types but want to compare two indirectly but not explicitly related pieces of knowledge. The retained solution in SEPT has been to provide the operator with the ability to explore parts of the line of reasoning of SEPT through varying levels of detail. These levels are obtained by the attachment of basic interpretations to actions to selectively execute some of them when a rule is fired.

Encountering an active point of interruption, the inference engine asks the user to define the context of the desired observation by selecting basic interpretations. The user defines the length of the path to explain and chooses a combination of elementary interpretations according to the desired level of detail. SEPT then executes the simulation of the corresponding part of the reasoning and returns to the point of interruption. This cycle ends when the user inhibits the point of interruption.

The meta-rule that calls a rule packet in SEPT defines a context, e.g., the context of a circuit breaker. The context may be refined by the operator, e.g., the context of the circuit breaker of a given cut-off system. This static aspect of context is directly linked to the structural and functional organization of the rule base. There is also a dynamic aspect of context who is established by the user that builds an explanation with path of the reasoning in SEPT.

As the system reasons and the user asks for explanation, the user discovers that the context of reasoning of the system does not match what the user expected. The problem solving is then expanded to attempt to align the reasoning of the system with that of the user through clarification dialogue, and explanations are co-constructed by the system and the user. This clarification dialogue with the system may result in the incremental acquisition of new knowledge.

2. CONTEXT IN DOMAINS AND APPLICATIONS

2.1 Context in Artificial Intelligence

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. In the formula ist(C P), C is meant to capture all that is not explicit in P but that is required to make P a meaningful statement representing what it is intended to state. Each context has a vocabulary associated with it. So, there might be some contexts in which P might not be stateable (in the vocabulary of that context) and there might be yet other contexts in which P is stated differently. 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]. (McCarthy maintained an extended version of his paper entitled "Formalizing context" at IJCAI-93, which can be found on his www page at http://www-formal.stanford.edu/jmc/index.html.)

There are other approaches. For example, in ATMS context is a consistent set of propositional assumptions [de Kleer, 1987]. De Kleer defines context of a consistent environment as the set of all assertions derivable from the propositional assumptions. The ATMS associates with each assertion the minimal set of environments from which it is derivable and thus has meanings. This set is the label of the assertion. Given the label, it is very easy to calculate the presence of an assertion in a context: an assertion is in a context when the assumptions of the context are a subset of any environment of the label of the assertion. This permits reasoning to be carried out simultaneously on several contexts. Several programming languages (included ART and Knowledge Craft) provide means to manage some types of context.

Another reason for considering context is the possibility to tackle large knowledge bases and complex reasonings. For instance:

(1) 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.

(2) 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.

(3) 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].

(4) Hendrix proposes a partition of semantic networks by managing context [Hendrix, 1975].

(5) Guha and Lenat use context in CYC for partitioning a knowledge base in more manageable modules [Guha 1991; Lenat, 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) Aikin's CENTAUR system explored the use of context-like prototypes to partition the ruleset. [Turner, 1993].

2.2 Context in databases and ontologies

The main role of context is to provide humans with a much greater control over the knowledge. Context permits to define which 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, the plumber, the chemist, and the painter.

As an alternative to the integration approaches in the literature, Goh et col. (1995a; 1995b) 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 which 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 permits to distinguish the source (export) and receiver (import) contexts, and 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, and 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 col. (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 interaction 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 module's role 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 call MODEL. As a result, it is shown that shareable ontologies are a fundamental precondition for reusing knowledge, serving as a 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 requirement of the component, and (4) encapsulates the component's behavior so that the component can be reused and shared.

Along with the research on the implementation of context in database, some products appear 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 text, documents, manuals, e-mail and news [Oracle 1996]. The software builds on previous generations of Oracle text and database products to provide intelligently management of all corporate information. The product is supposed to "understands" 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 to 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. "Oracle ConText Option is expected to be available by the end of Q2 1996." 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[1]

In a report of a workshop on context held at 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 thing 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 col. (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 a 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 query, to provide concise, pertinent information immediately, to help user to define questions, and to anticipate users' need 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, the user's profile/system access allowed by the security.

The main point underlines by Mittal and Paris (1995) is that communication (and mainly explanation as 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 of 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 as the problem solving situation, the participants involved, the mode of interaction in which communication is occurring, the discourse taking place, and the external world. 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.

---

[1] The workshops at CHI'89 and CHI'90 are presented in the section 4.1.

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 to define 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 of communication consists of the illusion of sharing a context which creates misunderstandings.

Making context explicit would also permit to revised some well-known paradigms in AI. For instance, Reichman (1985) propose to restate two Grice's conversational maxim, 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) via the development of a distinct conversational move, whose relationship to some preceding context space is clear. If all preceding topics have been fully developed (all criteria for the development of particular conversational move 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 either will constitute a temporary interruption or will have to signal clearly to which portion of the preceding discourse it is related.

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 contextual modular 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 and explanation

Leake (1992) consider 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 occurs the event. 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 relies 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 explainer beliefs and goals-- is crucial in deciding an explanation's goodness, and a theory of contextual influences can be used to determine which explanations are appropriate. Deciding explanations' goodness makes sense only in context of what triggered explanation and how the resultant explanation will be used. This leads to view explanation as 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 sortes 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.

Moore (1995) presents a view similar to that of Leake in Natural Language. She states the explanation problem as follows: given a communicative goal, find information from the expert system's knowledge sources that is relevant to achieving this goal, and organize this knowledge into a coherent multisentential text. In order 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 and 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 said. It always points to the node in the text plan representing the last clause uttered by the system.

Always 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

Karsenty (1994) argues that communication implies the sharing of a linguistical 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 the actor 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 [Edmonson and Meech, 1993; Karsenty and 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 and Falzon, 1992].

Most of researchers focus on explanations considering them as a transfer of knowledge **from** the system **to** the user. Feedbacks are used by the system to tailor its explanation to user's needs. However, users rarely may intervene in the generation of explanations. Conversely, the approach taken in SEPT lets the user build alone his explanation [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. We discuss now this evolution on the research on explanation generation and the need of making context explicit in it.

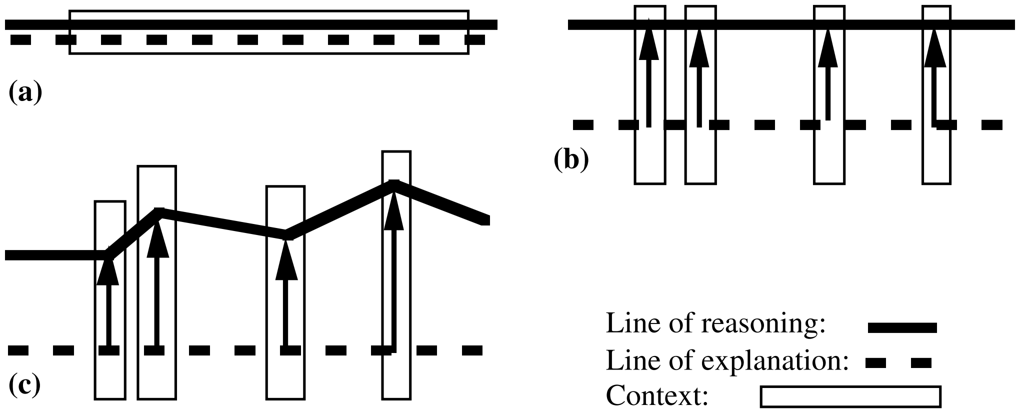


FIGURE 1: Line of reasoning versus line of explanation

As represented in Figure 1-a, 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 traversed contexts 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. 88; Wick and Thompson, 1992], an explanation facility is aligned only periodically with the reasoning of the system (see Figure 1-b). In Wick's system, only some of the contexts that the system reasons with are explained to the user. In this approach, additional explanatory knowledge (knowledge on the domain and the expertise that 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 is here 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 some information on users through a model. 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 as in the case of SEPT. Thus, the user and the system may have different interpretations on 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 95]. In order 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 the line of reasoning of the system may be modified by explanation (see Figure 1-c). 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 in order 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 of the context in which are the two actors in the explanation generation. 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 which 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 and Porter, 1991].

Context appears to guide the interactions among actors and search for suitable explanations. Interaction is seen as navigation in context space. For instance, Huuskonen and Korteniemi (1992) propose to follow different steps for producing an explanation accounting for the context:

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

With such a method, it seems that the user must accept easily 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 and 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. Firstly, the co-building of explanations is an accepted idea but rather very few studies consider it. Secondly, context being not a mature domain of research, its dependency upon explanations is not really considered. 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.

2.5 Context and knowledge acquisition

Knowledge acquisition is a difficult and time-consuming task. The difficulty raises because experts do not report on how they reach a decision. As a consequence, the decision is acquired out of its context when experts rather 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 and col. (1992) describe a context-definition language call MODEL for the PROTEGE-II system. MODEL is a metatool architecture that associates an explicit data model with every reusable module in the tool's 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 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 obtain knowledge acquisition 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 that comes before this rule is exactly the same. (This representation is used in a 'Knowledge Dictionary' [Jansen and Grosz, 1990].)
- Gruber (1991) propose to consider a 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 and col. (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 and col. (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 which prompted the change to the knowledge base [Compton et al., 1991; Paton et al., 1993]. Knowledge has not to be generalized when it is acquired. It is fundamental to record the context in which the knowledge is acquired [Compton and Jansen, 1990; Guha, 1991; Turner, 1993].

However, the acquisition of knowledge in context is still a challenge. Firstly, 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 a 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 deal 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 a 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 and Jansen, 1990]. One advantage is that is 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 knowledge that is acquired in a given context, may be used out of this context?

Although the approaches described in this section yet present serious weaknesses, as presented, they are ascribed in the realm of the incremental knowledge acquisition in context [Boy, 1991b; Brézillon and Cases, 1995].

2.6 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 perfoms 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 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 two lost (implicit) contextual information.

Park and Wilkins (1990) describe a failure driven learning with a context analysis mechanism as a method to constrain explanations and thereby increase the number of learning opportunities by about 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 similar if their strategy axes are the same and their focus axes can be grouped by a known relation. For example, consider context1 = (clarify-finding, surgery) and context2 = (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 that is proposed by Lashkri and col. (1994) achieves a level of personalization that is impossible with the knowledge engineer, and 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 thus collaborate. When a user takes an action, it is paired with the corresponding situation and the situation-action pair is record in the agent's memory. For example, if the user reads a message M, the pair <M', read-action> is memorized, where M' contains details about M and relevant context information.

2.7 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” which 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 which 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.

Jansen and Grosz (1990) proposed 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 maintenance. 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 extract 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 author's organized their discussion about context around the idea of K_lines of Minsky [Minsky, 1986].

2.8 Context in vision

Context is an important factor in different applications in the vision area: character recognition [Toussaint, 1978], target recognition [Forman et al., 1984], image recognition [Mohr and 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, learning. Desvignes et coL. (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) held in 1995 on using context in vision. Context is considered for change detection (site updates or activities, and irrelevant changes due to illumination differences or seasonal variations) [Aaron et al., 1995; Burlina et al., 1995], for mapping from sequences of image events to actions the system is to perform [Kjeldsen, 1995], for tracking between frames [Intile and Bobick, 1995]. Globally, it is acknowledged that the use of context allows for the design of less complex and more robust image understanding algorithms.

2.9 The CYC project

One of the primary reason for introducing contexts is to simplify the construction of the knowledge base. The context mechanism was introduced into the CYC system in May 1990. In November 1991, CYC contains over 1.5 million sentences and covers a wide range of phenomena [Guha, 1991]. Contexts are considered as rich objects in a first-order framework, extending the logic as required [Lenat, 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.

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 make different attributions of an object, but these attributions are about the same object. So, there might be some contexts in which P might not be stateable (in the vocabulary of that context) and there might be yet other contexts in which P is stated differently. 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. The meaningfulness of a formula may depend on the context it occurs in. Different contexts permit to use different languages.

Huhns et col. (1993) use the CYC knowledge base as a context. Then, a model is a set of frames and slots in a CYC context created specially 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.

On McCarthy's work on context logic, Farquhar et col. (1995) present an approach to integrating disparate heterogeneous information sources. They show that the use of context logic reduces the up-front cost of integration path, and allow 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 semantics (or integrating) contexts. The consequences of using context logic to integrate information sources are: integrate new information sources incrementally; share assumptions among information sources without making them explicit; exploit shared ontologies; provide a richer model of integration that goes beyond global schema or federated schema methodologies.

2.10 The ORCA project

Turner (1993, 1995) 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 focus its attention on appropriate goals to achieve in the current situation. Thus, context intervenes in at least five different ways: (1) make predictions about the situation; (2) modulate agent's behavior; (3) focus agent's attention; (4) influence an agent's choice of actions; (5) determine how an agent should handle unanticipated events. An agent should be able to recognize its current context as an instance of a class of contexts it knows about. It should be able to reason about its context, bringing to bear knowledge that is explicitly known to be contextual in nature.

Contextual knowledge is represented as a set of contextual schemas (c-schemas), then retrieving the most appropriate of those and using them to help the reasoner behave appropriately for its current context. Thus, c-schemas contain information not only describing the context they represent, but also information prescribing how to behave in situations that are instances of that context. Schemas provide a natural way to represent contexts 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 combining as needed to adequately represent a particular situation. The major difference with case-based reasoning is how c-schemas are used: generalized cases are usually used as indexing structures, while c-schemas are problem-solving structures in addition to their role in memory organization. Context is mainly considered as a way to cluster knowledge for search efficiency, for representing counter-factual or hypothetical situations, for circumscribing the effects of particular actions to particular situations, and for directing an agent's focus of attention to salient features of a situation.

2.11 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 for past events into episodes, and the location of the episode, who was there, what was going on, and what happened before or after, are all strong cues for recall. Thus, the authors propose a 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 and col. (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--concretized by an active badge that each employee has--pays particular attention to meetings and other episodes involving two or more people. The episodes are presented to the user as diary generated at the end of each day and distributed by electronic mail. After reading their diaries, users remembered events that they had previously forgotten.

3. CONTEXT, MODELING AND REPRESENTATION

3.1 Context in logic representation

A number of applications exist along the approach developed by de Kleer (1987). 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.

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 compare different, sometimes conflicting solutions.

More recently, McCarthy (1993) defined a context as a generalization of a collection of assumptions. Contexts are formalize 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.

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

There are many other relations among contexts and context valued functions. For instance, there is a general relation *specializes* between contexts, specialize(c1, c2), that indicates c2 involves no more assumptions than c1 and every proposition meaningful in c1 is translatable into one meaningful in c2 (inheritance of ist in both from the subcontext to the supercontext and vice versa).

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.

3.2 Context in rule-based formalism

In a rule-based representation, context may be expressed on the basis of either the knowledge structures (if explicitly represented) or the functionalities 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, the diagnostic expert system SEPT dealt with pieces of equipment as circuit breakers and protective relays. Checking the internal behavior of a circuit breaker implies an expertise that is independent of the expertise on the internal behavior of a protective relay. Thus, the reasoning is local and needs not to tackle the overall expertise, and in the context of the circuit-breaker diagnosis only knowledge structures are represented.

In a 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 is the well-known example of screening clause is the following rule in MYCIN [Clancey, 1983]:

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.). The 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. 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.)

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 pieces 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 EHV 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) a 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 details about ALOUETTE that has been extended to constitute the METAL language concerning this part.)

It seems that this process of contextual instanciation of the variables plays an important role. In linguistics, Recanti (1992) proposes to represent a linguistical 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: the lifting rules [Guha, 1991], the bridging rules [Giunchiglia, 1993], the pragmatic rules [Girotto, 1991], the ripple-down rules [Compton and Jansen, 1990].

3.3 Other representations of context

There are several theoretical approaches (other than those described before) that consider explicitly or not context. Most of these approaches try to model context, not only represent it. Note that we distinguish model 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 whatever the way is. A model is endowed in a theory and a representation lies on the representation formalism chosen. As the two goals are different, we consider here modeling at the theoretical level and representation at the programming level. A successful modeling is a modeling that is used in applications.

Context has been modeled on the basis of the Situation Theory [Barewise and 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 Ackman (1995) approach context as an amalgamation of grounding situation and the rules which govern the relations within the context. They represent a context by a situation type which supports two types of infons: parameter free infons to state the facts and the usual bindings. Parametric infons (which corresponds to parametric conditionals) aim at capture the if-then relations and axioms within the context. In Computer-Human Interaction, Bonnie Nardi (1992) presents a study of context from a comparison of the activity theory [Leont'ev, 1974], the situation action models [Lave, 1988] and the distributed cognition [Flor and Hutchins, 1991].

Ejkova (1989) replaces knowledge base descriptions by semantic spaces, and then uses them for generating new knowledge bases as well as for comparison, intersection, integration with knowledge from other sources. Thus, the context is considered as a semantic background. Ejkova (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 the 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 the 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). An algebra of contexts is proposed to involve 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 with respect to: attribute significance, a set of attributes, the number of the most significant attributes, a set of basic concepts. Conversely, the context expansion operations are required for further learning, adaptation, introduction of new concepts, immersion in a larger context.

Behind this concept is the **problem area context (PA)**, which is a metanotion relative to knowledge bases. The PA permits 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 the concepts of the context. The distance is described in a universal manner. The distance between concepts in the context space is 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 which 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” which 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 (1993) presents contexts as relations that delimit the scope of the description of a concept in the conceptual network. They induce a context network among all concepts defined in the knowledge base. These are higher-order relation that constrains 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 allow this description to vary, if necessary, from the basic entity.

(3) Hovy (1990) has developed the PAULINE system that 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 (1990), a context is a set of propositions with various origins (prior discourse, memory content, observations, inference, etc.). Jointing 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 leads to confirm (or infirm) some propositions of the context. A pertinent proposition permits 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”

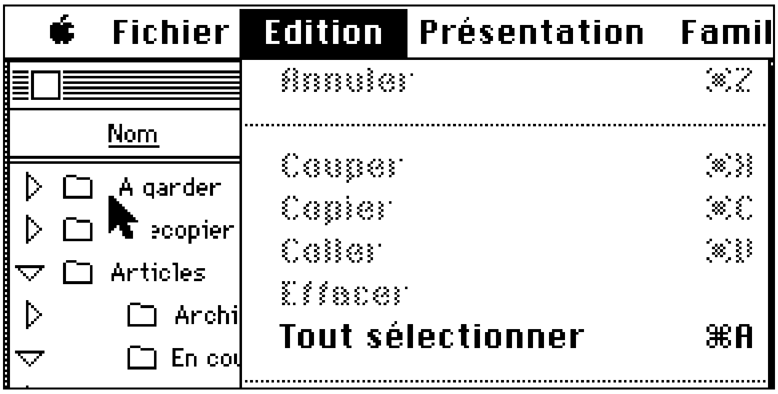
Contexts are organized in a context graph where arcs have the semantics “is-refinement-of” (e.g., the context 'failure-in-electric' is a refinement of 'failure-in-car'). The diagnostic process goes through the context graph by testing symptoms according to the ordering rules of the actual context and 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 follows: [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.

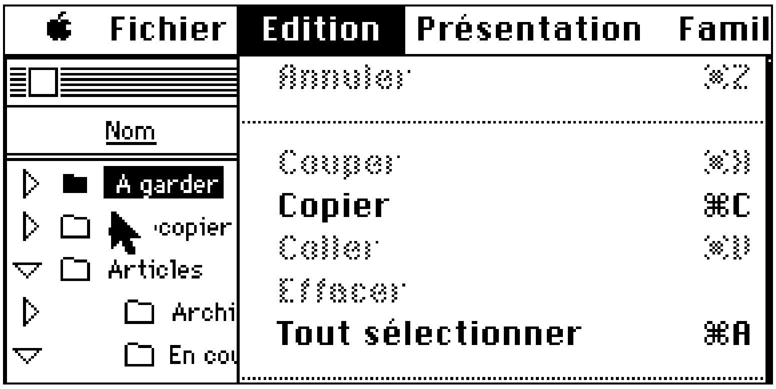
(7) Wu and col. (1992) propose a plan reuse framework in which restricted conceptual graphs are used as the internal representations of these skeletal plans and reusing these skeletal plans. The domain-specific relation set R only includes three relations, R= IS_PART_OF , IS_CONTEXT_OF, and IS_INSTANCE. 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 the section 2.10, 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 combining as needed 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 of software use context in a static way (thus generally imposed at users). The preference file (.profile) under UNIX permits the user to specify his working context by defining his favorite organization of the screen. MicroSoft proposes in software some contextual-sensitive help (the balloons, the scrolling menus, etc.). On MacIntosh, the items in a popup menu are visible but are only activable in some contexts, e.g., the item 'Copier' (Copy) only is activable if a file is selected (see the figures A and B below). The list of items of a given popup menu may change from one working context to another.



(A) The Edit popup menu without file selected



(B) The same Edit popup menu with a file selected

Some programming languages are proposed on Internet to represent some types 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)]' permits to set the debug context to compute partition processes.

- http://www.ci.com/support/ca/techtips/CONTEXT_RESTR.html

  Here context is represented as a set of restrictions that limit the access to parts of a system.

Most of these contexts are used to managed information in an arborescence.

3.5 Representing context in database interrogation

We lead a study[2] on how making context explicit in the interactions of a user and a database to make more relevant the system participation. We guessed that if the designer can improve the interrogation language with feedbacks from the user, there would be a part that depends on the particularities of each user. Thus a system must propose to improve user's requests dynamically when interacting with the user in a way that can not be planned beforehand. Such contextual knowledge may be obtain by an empirical way from the user, not from the designer.

The context intervenes in database at different levels: 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 system knowledge on 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.
* A number of 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 and that may use. If made explicit, such a contextual knowledge would permit the system to complete automatically user's entry.
* An experienced user prefers to write himself a SQL request 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 (e.g., counting the same item when this one appears several times in the database).
* A request may be written by different methods (e.g., using a COUNT or NOT EXISTS clause). Each method does not need the same CPU time and depends on the characteristics of the database. A solution would be that the system identifies rapidly what the user is writing, find alternatives and propose the most optimal expression according to some criteria.
* For using in practice 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 of 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 request containing a division. The chosen example is the choice between the clauses COUNT and NOT EXISTS. Generally, the clause NOT

[2] This work has been realized with Philippe Charron during the preparation of his Master degree at the *Institut d'Informatique d'Entreprises (Evry, France).*

EXISTS is preferred for a simple writing of the request. However, a request either with a COUNT clause or with a NOT EXISTS clause does not give the same results in CPU time. 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:

(1) General rules:

IF user sends a request,
THEN identify pieces of the request for which alternative expressions exist.
IF a user's request possesses several expressions,
THEN find the expression that is the faster.
IF the user's request is not under the faster expression,
THEN propose to the user the faster 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 better alternative.

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

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

4. CONTEXT AND RELATED ELEMENTS

4.1 Introduction

This section presents some topics discussed at four of the workshops dealing directly with context. The two first workshops held at the Computer-Human Interaction (CHI) Conference in 1989 and 1990. Reports on these workshops are given in [Maskery and Meads, 1992; Maskery et al., 1992]. The two other workshops 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 addressed mainly 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. Electronic discussions went before the two IJCAI workshops around topics drawn from accepted papers.

Even in more specific domains as CHI, a general observation is that most of the time was spent in trying to establish what context is, how it can be defined, and how it can be distinguished from other notions as 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 be discussed in section 2.3.

4.2 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 a problem solving (or interaction). The two diverging positions arise from the consideration of context at the level of the interactions among agents or at the level of the knowledge representation.

4.2.1 Cognitive and engineering points of view

The notion of context is dependent in its interpretation on a cognitive science versus an engineering (or system building) point of view, the practice viewpoint versus the theory 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 and 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.

4.2.2 Static versus dynamic contexts

According to the engineering viewpoint, the 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 on 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). Along its dynamic aspect, part of the problem is linked to the changing nature of context in time, by elaboration and shift [Clancey, 1991]. 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.

Thus, 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.

4.2.3 Implicit versus explicit contexts

Contextual knowledge acts as a filter that defines, at a given time, what knowledge pieces must be taken into account (explicit) from those that are not necessary or already shared. A context is a structure, a frame of reference, that permits to do not say all the things in a story. For example, "At his birthday's party, Paul blew up 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 call 'Le Chat' (the cat) presenting the life a husband and his wife living together since 40 years. Knowing very well the other, they had severely limited their communication. For instance, with a light movement of the chin toward the cupboard, the husband said "Can you please dearling give me the salt that is in the cupboard." With a computer system, however there is a compromise to find between the need to store a large number of information pieces and a tailored presentation of the answer to the user's question.

In the SEPT application, the knowledge engineer discovered very late that equipment in a substation function only when there exists a fault on the network. It was so evident that the expert had never expressed this contextual information before a conflict arose with the knowledge engineer. The expert had compiled this knowledge that became implicit when explaining the functioning of a piece of equipment. Other implicit knowledge in the SEPT application is: protective relays see effectively the fault to function, transmission cables transmit correctly signals from equipment to the central printer, the fault concerns the substation and does not occur in next substations.

This supposes that people share knowledge. However, it is frequent that the context--the frame of reference--must be built for the listener. For example, I may say to a person: "I heard a lion roars in my office this morning" (called hereafter CK_0, 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_0 and may suppose that I am speaking of my boss), I must introduce different pieces of knowledge that I suppose shared 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_0 is introduced after various CKs are first presented.

The former person knows these knowledge pieces (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 for explaining my first statement.

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

4.2.4 Contextual knowledge and contextualized knowledge

Contextualized knowledge is knowledge that is explicitly considered in the problem solving. Contextual knowledge intervenes implicitly in the problem solving, often as constraints. Consider the two following examples.

Example 1. Operators that ensure the monitoring of the distribution of water in Paris had noted that there was a peak in the water consumption late each evening. The peak was reproducible every day but not predictable because not exactly at the same time. After an inquiry, they discovered that persons use water for domestic needs (drink a glass of water, wash dishes, pour water on flowers, go to the toilets, etc.) during the ads introduced in the movie at the TV. The introduction of ads in the movie depends on the organization of the scenario. Such a knowledge (the link between the peak of consumption and the ads at the TV) has a contextual nature for the 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 interval of time during which people go back at home, prepare the meal, and so on. This is so well known that the European power systems companies play with the different timing of the peaks in each country of Europe to import electricity and export their electricity production according to the off-peak hours of each European country.

In both examples, contextual knowledge is used to constrain the problem solving (water and electricity distribution). In the first example, it is difficult to account for (and use in a computer) the contextual knowledge--the time of ads. at the TV--has an unpredictable nature because ads. is not at exactly the same time every evening, even if it is (too much) repetitive each day. Thus, one needs to look for this contextual knowledge when needed in the problem solving. Such a contextual knowledge concerns the dynamic aspect of context. In the second example, the contextual knowledge--people go back at home--is predictable with a good accuracy. Such contextual knowledge can be used in the planning programs of the electricity production. Thus, the contextual knowledge becomes operational (use in a system) and then contextualized knowledge.

When contextual knowledge can be well identified empirically, it is possible to include it in the problem solving. In the examples above, the contextual knowledge acts as a constraint. Thus, contextual knowledge becomes contextualized knowledge. Conversely, 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. For example, the user's identification constrains after the available functionalities of the system for that user. Such a knowledge piece is first in the focus of attention (contextualized knowledge) and after acts as a filter of user's accesses at functionalities of the system (contextual knowledge).

This show that part of the users' feedbacks may be encoded by the designer because required improvements are general, repetitive and static. However, it will have always a part of users' feedbacks that are specific, depending in time on the task at hand, the conditions in which the task is accomplished, etc. For example, users do not all wish the same type of warning (and may change of advice along the software use becoming increasingly knowledgeable with the software): an alert window, a sound, a special icon, or any combination of them. Only the system, during its interaction with the user may tailor its behavior according to the user's reaction. For doing this, the system must manage some contextual information on the interaction.

The successive versions of a piece of software take into account for users' feedbacks and show the importance of a user-centered approach. However, what may please at a part of users will not please at another part. For example, the MacIntosh Operating System 7.5 provides a support at three levels when the user wants open an item: (1) a folder containing the last documents open whatever the application is; (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 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, when previous versions started from the current folder.

4.3 Applying context

There is a consensus on the fact that context is inseparable from its use. Context is considered as a shared knowledge space that is explored and exploited by participants in the interaction. Such shared knowledge includes the history of all that ensured over an interval of time; the overall state of knowledge of the participating agents at a given moment; the small set of thing they are attending to at that particular moment. Other elements intervening 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]. This is partly because the world cannot be objectively and exhaustively described; cultural or social circumstances cannot be reduced to a set of facts and procedures [Lave, 1988; Clancey, 1992].

The context permits to guide the focus of attention, i.e., 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) was seen as being the mutual context that already exists between the agents [Maskery and 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 stance of Giunchiglia at the IJCAI-95 workshop was that "contextual reasoning is local reasoning." This stance is compatible with the various communications in various 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 serve two purposes. The import rules allow a fact about an object to be added to the object's description. It enables a simulator to import messages into a description. The export rules allow encapsulated information to be taken outside the context of an object. This permits to provide focus to speed up problem solving behavior, and 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 permits 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, by making assumptions and by specializing to the context of a conversation [McCarthy, 1993].

Creating a context from existing contexts, it is possible to establish a hierarchy of contexts where 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 another.

However, there is a compromise to find between the cost and the utility of a context: Context analysis must be less time-consuming than the brought improvement [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].

Moreover, this assumption, which permits interesting theoretical developments, is difficult to hold in practical situation because if we know that there exists contexts, we cannot formalize them, mainly because they refer themselves to other existing contexts recursively. Facing a real-world problem, we are obliged to have an opposite approach, identifying the current context directly in an empirical way, and try 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 cover 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 captors. Few data provided by captors are transformed by the pilot into information, according to the current context (e.g., a flight in normal conditions and entering a turbulence zone). Context as a process of contextualization is considered in electronic documentation where the path follows by a user to reach an information is used as a context to guide the search [Boy, 1991a; Jansen and Gray, 1992].

However, although empirical evidence clearly shows that recall is much better in contexts similar to the context where the information was acquired, recall of decontextualised information is possible and even desirable in problem solving. Sandberg and Wielinga (1991) give the example of the Newton's law (F=ma) that one recalls under its abstract expression, not in all the situations where one have applied it.

5. DISCUSSION

Our interest for context is for the 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. Our survey of the literature on context has been made according to this point of view when the literature on context is very large and considered in a number of domains.

Context plays an important role in all domains with activities as reasoning and interpretation, and can only be caught by experience. This interest by the use of context implies that there is no clear and general definition of context. Context seems to possess a double nature: static or dynamic, discrete or continuous, knowledge or process. This apparent double nature arises from the fact that the notion of context is dependent in its interpretation on a cognitive science versus an engineering (or system building) point of view [Brézillon and 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, etc.

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 several contexts occurring in a discussion, into which all terms and predicates can be lifted. We consider that such a description may be presented in the onion metaphor, where contexts are in layers as the onion skins around the heart of the onion [Agabra et al., 1007; Brezillon et al.; 1997]. Along a different approach, Giunchiglia (1993) considers a set of discrete contexts (then, at a same level) and the relationships between them (through bridging rules for enter and leave contexts). Both of them consider context at the representation level.

A contextual system must 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). The managing of these two types of memory may be organized with an algebra of contexts to involve contraction, extension, immersion, coupling and intersection of contexts [Ejkova, 1992].

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 compatible the interpretations (or reasonings) in the export and import contexts. 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 coexist different contexts: 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 don't know what is really needed until one is in the design process. A consequence is that context permits 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 permit to structure knowledge bases in tracktable 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 claim that acquiring knowledge with its context of use, leads to organize knowledge representation into episodes (Boy, Turner, Lamming & Flynn). 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).

The context is considered as 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. Moreover, a contextual system must 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 combine to constitute complex contexts to adequately represent a particular situation.

The relationships between context and knowledge are yet to explored. A piece of knowledge may be contextual or contextualized according to the step of the problem solving where we are. Contextualized knowledge is knowledge that is explicitly considered in the problem solving. At this title, 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), MacCarthy (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. Say with other words, a context is a structure, a frame of reference, that permits to do not say all the things in a story. Context permits to let 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 extend out of formal logic by the following claim:

"Context is what constrains a problem solving without intervening in it explicitly."

The claim is let here vague because we think that before refine the context definition, 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 context are dependent of the others. For example, the way in which a user defines its 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 discusses widely how the writer needs a model of a reader and how the writer may plays to leads the reader in a cul-de-sac. This again illustrates the importance to well define context, and the context in which context is defined.

The discussion stays still open. 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?

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, pp. 13-20.

Arbab B. (1992) "A formal language for representation of knowledge", Proceedings of the AAAI'92 Workshop on Propositionnal Knowledge Representation, Stanford, CA, pp. 1-8.

Aussenac N. and R. Dieng R. (1991) "Models of problem solving for knowledge acquisition: comparison of MACAO and 3DKAT", Part II of Sisyphus Project, EKAW-91, Glasgow, UK, May 1991, pp. 1-23.

Barewise J. and Perry J. (1983) Situations and Attitudes, MIT Press, Cambridge, MA.

Barr A. and E.A. Feigenbaum, Eds. (1981) "The Handbook of Artificial Intelligence", William Kaufmann, Inc., Vol. 1, Chap III: Representation of Knowledge, représentation".

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, novembre.

Bastien C. (1992) "Le décalage entre logique et connaissances", Le Courrier du CNRS, Numéro Spécial “Sciences Cognitives”, 79, pp. 38.

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

Bianco L., Dell'omo P., & Ezhkova I.V. (1994) "Application of contextual technology for supporting decision making in transportation", Proceedings of the 7th IFAC/IFORS Symposium on Transportation Systems, Tianjin, China, pp. 363-368.

Bloom C., Bullemer P., Chu R. & 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, pp. 509-514.

Bobick A.F. 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, London, Knowledge-Based Systems Series, Vol. 6.

Boy G. (1991b) "Computer integrated documentation ", NASA Technical Memorandum 103870, NASA Ames Research Center, Moffett Field, CA 94035-1000, USA, September.

Boy G. (1995) "Design rationale and supportability: An agent-oriented corporate memory approach", Proceedings of the International Workshop on the Design of Cooperative Systems, January.

Brézillon P. (1990) "Interpretation and rule packet in expert systems. Application to the SEPT expert system", Lecture Notes in Artificial Intelligence, N° 444, Knowledge Based Computer Systems, S. Ramani R. Chandrasekar K.S.R. Anjaneyulu (eds.), Springer-Verlag, pp. 78-87.

Brézillon P., Bau D.Y., Fauquembergue P., Hertz A. & 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, South Carolina, Vol. I, pp. 54-60.

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, pp. 11-22.

Brezillon P. (1994) "Context needs in cooperative building of explanations", Proceedings of the First European Conference on Cognitive Science in Industry, Luxembourg, September 28-30, pp. 443-450.

Brezillon 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., & Cases, E. (1995) "Cooperating for assisting intelligently operators", Proceedings of the International Workshop on the Design of Cooperative Systems, INRIA ed., pp. 370-384.

Brézillon P. and Pomerol J.-Ch. (1996a) "Misuse and nonuse of knowledge-based systems: The past expreiences revisited", In: Implementing Systems for Supporting Management Decisions, Eds P. Humphreys, L. Bannon, A. McCosh, P. Migliarese and J.-Ch. Pomerol, Capman and Hall, ISBN 0-412-75540-8, pp. 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).

Burlina P., Chellappa R., Lin C. L., 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, pp. 20-29.

Carenini G. and J.D. Moore (1993) "Generating explanations in context", International Workshop on Intelligent User Interfaces, Orlando, Florida.

Cavalcanti M. (1993) "Les mondes possibles dans les systèmes de production : un métalangage pour la gestion d'hypothèses et le raisonnement non-monotone", Ph. D. Thesis, LRI, Université Paris-Sud, Orsay, France, June.

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

Compton P. and Jansen R. (1990) "A philosophical basis for knowledge acquisition", Knowledge Acquisition, 2, pp. 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, August, pp. 24-40.

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

Clancey W.J. (1986) "From GUIDON to NEOMYCIN and HERACLES in twenty short lessons: ORN final report 1979-1985", The AI Magazine, pp.40-60.

Clancey W.J. (1991) "Israel Rosenfield, The Invention of Memory: A New View of the Brain (Book Review)", Artificial Intelligence, 50, pp. 241-284.

Clancey W.J. (1992) "The knowledge level reinterpreted: modeling socio-technical systems", Proceedings of the AAAI'92 Worshop on Cognitive Aspects of Knowledge Acquisition, Stanford, CA, March, pp. 47-56.

Clancey W.J. (1993) "Notes on “Epistemology of a rule-based expert system”", Artificial Intelligence Journal, 59, pp. 197-204.

Compton P. and Jansen B. (1988) "Knowledge in context: A strategy for expert system maintenance", Lecture Notes in Artificial Intelligence, J.Siekmann (ed), Subseries in Computer Sciences, Vol. 406.

Coutaz J. & Caelen J. (1993) "Projet Grenoblois: Interface multimodale", Rapport d'activité du GDR-PRC CHM, pp. 61-123.

de Kleer J. (1987) "An assumption based truth maintenance system", In: M Ginsberg ed., Readings In Nonmonotonic Reasoning, Morgan Kaufmann, Los Altos, CA, USA.

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, pp. 467-472.

Desvignes M., Revenu M. and Porquet C. (1991) "The use of context in image sequences interpretation", Proceedings of the 8e Congrès AFCET-RFIA, November, pp. 55-61.

Edlund C., Weise C. and Lewis M. (1995) "Context engineering for human problem solving", Proceedings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Technical Report 95/11, LAFORIA, University Paris 6, France, March, pp. 79-88.

Edmondson W.H. and Meech J.F. (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, pp. 31-38.

Eklund P. (1989) "Prospects for conceptual graphs in acquisition interfaces", Proceedings of EKAW89, Paris, France, July, pp. 169-179.

Ezhkova I. (1989) "Knowledge formation through context formalization", Computers and Artificial Intelligence, 8(4): 305-322.

Ezhkova I.V. (1992) “Contextual systems: Is it a way of a universal expert system development?”", In: General systems, G. Klir Publisher, New Jersey, USA.

Farquhar A., Dappert J., Fikes R. and Pratt W. (1995) "Integrating information sources using context logic", http://www.ksl.stanford.edu.

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. (Eds.), Proceedings of the Fourth Annual Workshop on Empirical Studies of Programmers, Norwood, NJ, Ablex Publishing, pp. 36-59.

Forman A.V., Rowland P.J. and Pemberton W.G. (1984) "Contextual analysis of tactical scene", SPIE 485, Applications of Artificial Intelligence, pp. 189-197.

Frege G. (1985) “On sense and meaning”, In: A.P.Martinich (ed.), The Philosophy of Language, Oxford University Press, pages 212-220 (Originally published in 1892).

Garfinkel A. (1981) "Forms of Explanation : Rethinking the Questions in Social Theory", Yale University Press, New Haven and London.

Girotto V. (1991) "Reasoning on deontic rules: the pragmatic schemas approach", Intellectica, 1991, 1, pp. 15-52.

Giunchiglia F. (1993) "Contextual reasoning", Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context, Research report 93/13, LAFORIA, pp. 39-48.

Goh C.H., Madnick S.E., and Siegel M.D. (1995) "Context interchange: Overcoming the challenges of large-scale interoperable database systems in a dynamic environment", http://context.mit.edu.

Goh G.H., Madnick S.E., Siegel M.D. (1995) "Ontologies, contexts, and mediation: Representing and reasoning about semantic conflicts in heterogeneous and autonomous systems", http://context.mit.edu.

Grant A. S. (1992) "Mental models and everyday activities", Proceedings of the 2nd Interdisciplinary Workshop on Mental Models, Cambridge, UK, March, pp. 94-102.

Grant A. S. (1994) "Modeling complex cognition: contextual modularity and transitions", Proceedings of the Fourth Int. Conf. on User Modeling, Hyannis, USA, The MITRE Corp., August, pp. 157-162.

Gruber T. (1991) "Justification-based knowledge acquisition", In: Knowledge Acquisition for Knowledge-Based Systems, IOS Press, H. Motoda, R.Mizoguchi, J. Boose and B. Gaines (Eds.).

Grusenmeyer C. (1995) "Les dialogues coopératifs en phase de relève de poste: Rôle dans la sureté des systèmes de production", Proceedings of the XXX° Congrès de la SELF “L'homme dans les nouvelles organisations”, Biarritz, France.

Guha R.V. (1991) "Contexts: a formalization and some applications", MCC Technical Report ACT-CYC-423-91, December.

Hatchuel A. and Weil B. (1992) "L'Expert et le Système", Economica, Paris, France.

Hendrix G. (1975) "Expanding the utility of semantic networks through partitioning", Proceedings of the Fourth IJCAI, pp. 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, March, pp. 51-57.

Hovy E.H. (1990) "Pragmatics and natural language generation", Artificial Intelligence, 43(2): 153-98.

Huhns M.N., Jacobs N., Ksiezyk T., Shen W.-M., Singh M. P., and Cannata P. E. (1993) "Enterprise information modeling and model integration in Carnot", Proceedings of the International Conference on Intelligent and Cooperative Information Systems, Rotterdam, Netherlands, May, pp. 12-14.

Huuskonen P. and Korteniemi A. (1992) "Explanation based on contexts", Procedings of the 8th Conference on Artificial Intelligence for Applications, Monterey, CA, March, pp. 179-185.

Intille S.S. and Bobick A.F. (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. and Bray G. (1992) "IntelliText", ECHT'92.

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, pp. 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.

Johnson H. and Johnson P. (1992) "Acceptable explanations: theory before implementation", Proceedings of the AAAI Spring Symposium on Producing Cooperative Explanation, Stanford, CA, March, pp. 95-102.

Kamp H. (1981) "A theory of truth and semantic representation", In: JAG Groenendijk, TMV Janssen & MBJ Stokhof (eds.), Formal Methods in the Study of Language, Mathematical Centre Tracts, Amsterdam, pp. 277-322.

Karsenty L. (1994) "L'explication d'une solution dans les dialogues de conception", Ph.D. Thesis, University Paris 8, juillet .

Karsenty L. and Brézillon P. (1995) "Cooperative problem solving and explanation", International Journal of Expert Systems With Applications, 8(4): 445-462.

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, June, pp. 115-124.

Kjeldsen R. and John Kender J. (1995) "Knowledge Based Hand Gesture Recognition", Proceedings of the IEEE Workshop on Context-Based Vision (CBVIS '95).

Laird J.E., Newell A. and Rosenbloom P.S. (1987) "Soar: An architecture for general intelligence", Artificial Intelligence, 33, pp. 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. & Maes P. (1994) "Collaborative Interface Agents", Proceedings of the AAAI'94 Conference, Seatle, Washington.

Lave J. (1988) "Cognition in Practice", Cambridge University Press.

Leake D.B. (1992) "Evaluating explanations", Lawrence Erlbaum Associates Inc., 1992.

Lenat D. (1993) "Context dependence of representations in CYC", Colloque ICO'93, Montréal, May.

Leong T.-Y. (1992) "Representing context-sensitive knowledge in a network formalism: a preliminary report", In: Uncertainty in Artificial Intelligence: Proceedings of the Eight Conference, Dubois D, Wellman MP, & D'Ambrosio MD and Smet P (eds.), Morgan Kaufmann, pp. 166-173.

Leont'ev A. (1978) "The problem of activity in psychology", Soviet Psychology, 13(2): 4-33.

Lester J.C. and Porter B.W. (1991) "Generating context-sensitive explanations in interactive knowledge-based systems", Proceedings of the AAAI'91 Workshop on Comparative Analysis of Explanation Planning Architectures, July, pp. 27-41.

Longino H. (1990) "Science as Social Knowledge: Values and Objectivity in Scientific Inquiry", Princeton, NJ: Princeton University Press.

Mackie J. (1965) "Causes and conditions", American Philosophical Quaterly, 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 EKAW'92, Lecture Notes in Artificial Intelligence, Springer Verlag, 599, pp. 337-352.

McCarthy J. (1979) "First-order theories of individual concepts and propositions", In: JE Hayes, D Michie & LI Mikulich (eds.), Machine Intelligence, 9, Ellis Horwood Ltd., Chichester, England, pp. 129-147.

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

McDermott J. (1982) "R1: A rule based configurer of computer systems", Artificial Intelligence, 19, pp. 39-88.

Minsky M. (1986) "The Society of Mind", Picador Publisher.

Mittal V.O. & C.L. Paris (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 J.D. (1995) "Participating in Explanatory Dialogues. Interpreting and Responding to Questions in Context", A Bradford Book, The MIT Press, Cambridge, MA, USA, Series Natural Language Processing.

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 B.A. (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, Russia, International Centre for Scientific and Technical Information Publisher, Vol. 2, pp. 352-359.

Newman W.M., Eldridge M.A. and Lamming M.G. (1994) "PEPYS: Generating autobiographies by automatic tracking", Research Repport, Rank Xerox.

Ogden C.K. and I.A. Richards (1946) "The Meaning of Meaning", Routledge & Kegan Paul LTD, Broadway House, 68-74 Carter Lane, London, E.C.4, 8th edition.

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 C.L., Wick M.R. and Thompson W.B. (1988) "The line of reasoning versus the line of explanation", Proceedings of AAAI Workshop on Explanation, pp. 4-7.

Park Y.-T. and Wilkins D.C. (1990) "Establishing the coherence of an explanation to improve refinement of an incomplete knowledge base.", Proceedings of the AAAI90.

Paton R., Shave M., Bench-Capon T. and Nwana H. (1993) "Domain characterisation in context", Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France.

Pereira F.C.N. and Pollack M.E. (1991) "Incremental interpretation", Artificial Intelligence, 50(1): 37-82.

Politzer G. (1991) "Introduction du numéro spécial “Pragmatique et Psychologie du Raisonnement", Intelleticta, 1, pp. 7-13.

Récanati F. (1992) "La pragmatique linguistique", Le Courrier du CNRS, Numéro Spécial “Sciences Cognitives”, 79, p. 21.

Reichman R. (1985) "Getting computers to talk like you and me. Discourse context, focus, and semantics", A Bradford Book, The MIT Press, Cambridge, MA.

Sandberg J. and Wielinga B. (1991) "How situated is cognition?", Proceedings of the Twelveth IJCAI'91, Sydney, Australia, Vol. 1, 341-346.

Schmidt C.T. (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, pp. 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, pp. 377-386.

Sharma N. (1995) "On formalising and reasoning with contexts", Research Report 352, Dpt of CS, Univ. of Queensland, Brisbane, Australia, September.

Sowa J.F. (1984)

Sowa J.F. (1991) "Toward the expressive power of natural language", In: Principles of Semantic Networks - Exploration in the representation of Knowledge, Morgan Kaufmann, San Mateo, CA, pp. 157-189.

Sowa J.F. (1992) "Representing and reasoning about contexts", Proceedings of the AAAI'92 Workshop on Propositional Knowledge Representation, Stanford, pp. 133-142.

Srinivasan A., P. Compton, R. Malor, G. Edwards and L. Lazarus (1991) "Knowledge acquisition in context for a complex domain", Pre-print of Proceedings of the Fifth EKAW91.

Surav M. and Akman V. (1995) "Modeling context with situations", Proceedings of the IJCAI-95 Workshop on Modeling Contex in Knowledge Representation and Reasoning, Research Report 95/11, LAFORIA, pp. 145-156.

Suthers D. (1992) "Preferences for comprehensible explanations under multiples models", Proceedings of the AAAI Spring Symposium on Producing Cooperative Explanation, Stanford, CA, March 1992, pp. 46-52.

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, Box 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France.

Tang M.X. (1995) "Exploring desing 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, March, pp. 157-166.

Terssac (de) G. (1992) "Autonomie dans le travail", Série Sociologie d'Aujourd'hui, Presses Universitaires de France.

Toussaint G.T. (1978) "The use of context in pattern recognition", Pattern Recognition, 10, pp. 189-204.

Turner R.M. (1993) "Context-sensitive reasoning for autonomous agents and cooperative distributed problem solving", Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context, Research Report 93/13, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France, May.

Turner R.M. (1995) "Using contextual knowledge in autonomous real-world systems", Proceddings of the IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Montreal, Technical Repport 95/11, LAFORIA, University Paris 6, France, March, pp. 189-196.

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, pp. 53-69.

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

Walther E., Eriksson H., and Musen M.A. (1992) "Plug-and play: Construction of task-specific expert-system shells using sharable context ontologies", Technical Report, http://www.stanford.edu.

Wick M.R. and Thompson W.B. (1992) "Reconstructive expert system explanation", Artificial Intelligence Journal, 54, pp. 33-70.

Widmer G. (1996) "Recognition and exploitation of contextual clues via incremental meta-learning (Extended version)", e-mail to “gerhard@ai.univie.ac.at”.

Winograd T. and F. Flores (1989) "L'intelligence Artificielle en question", Presse Universitaire de France, Ablex Publishing Company.

Walther E., Eriksson H., and Musen M.A. (1992) "Plug-and play: Construction of task-specific expert-system shells using sharable context ontologies", Technical Report, http://www.stanford.edu.

Wu Z., B. Ansgar and K. Christophe (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.