Context in Artificial Intelligence:
I. A Survey of the literature

RUNNING TITLE: Context in Artificial Intelligence: A survey

Patrick BRÉZILLON
LIP6, 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@polia.lip6.fr

Abstract:

Context is the challenge for the coming years in Artificial Intelligence. In the companion paper [8], we present the main results of discussions at two workshops and at the first conference focusing on the notion of context. In this paper, we present a view of how context is considered in knowledge acquisition, machine learning, communication, and databases and ontologies. We describe the way in which context is modeled and represented in the logic formalism and a rule-based formalism. We present briefly after some of the other approaches, and sum up the different points that may be of interest for modeling effectively context.

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

1 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. However, there is no clear definition of this word (references may be found in [7]): a set of preferences and/or beliefs, a window on a screen, an infinite and only partially known collection of assumptions, a list of attributes, the product of an interpretation, a collection of context schemata, paths in information retrieval, slots in object-oriented languages, buttons which are functional, customizable and shareable, possible worlds, assumptions under which a statement is true or false, a special, buffer-like data structure, an interpreter which controls the system's activity, the characteristics of the situation and the goals of the knowledge use, entities (things or events) related in a certain way, the possibility that permits to listen what is said and what is not said.

In Artificial Intelligence, the lack of explicit representation of context is one of the reasons of the failures of many Knowledge-Based Systems (KBSs) [11,12]. Studies of the use of KBSs in real-world applications permit to point out four main failures. The main problems arise mainly because the users that work with computer systems are not taken into account. For instance, Vanwelkenhuysen and Mizoguchi [54] 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). Here, the lack of context use appears more as a mechanism for presenting knowledge rather than for modeling knowledge.

Nowadays, one notes an increasing interest for the context. Since 1991, several scientific events in Artificial Intelligence dealt specifically with context (see the reports [40, 41, 9, 13]). Results on these events focusing on context are discussed in the companion paper [8] and [7] give an extended version both of them. The modeling, representation and use of context appears to be the challenge of the coming years, particularly when we now face very complex problems, large knowledge bases and multimedia.

Hereafter, the paper is organized in the following way. Section 2 presents a view of how context is considered in knowledge acquisition, machine learning, communication, and databases and ontologies with a brief presentation of the CYC project. We give in section 3 the way in which context is modeled and represented in the logic formalism and a rule-based formalism, and we present some of the other approaches. The section 4 sum up the different points that may be of interest for modeling effectively context.

2 CONTEXT IN SOME DOMAINS

2.1 Context in 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. Context in 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 al. [55] 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: Each context contains a set of concepts (also called schema, frame, or structure) that describes 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, the latter alternative, leads to contextualize knowledge acquisition. Knowledge is then encoded into the system when it is needed, i.e. in its context of use. Contextualizing the knowledge acquisition process helps ensure that relevant knowledge is put in the knowledge base, because one doesn't know what is really needed until one is in the design process. We are now in the realm of **incremental**

knowledge acquisition.

There are different ways to acquire knowledge in context:

- Compton and Jansen [19] 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 that produced the wrong interpretation has fired first. Thus the new rules are tested precisely in the context in which the expert provided them. 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' [36].)

- Gruber [33] 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 is 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 [51].)

- Bloom and al. [5] present a task driven approach to acquiring, analyzing and representing knowledge for an intelligent tutoring system (ITS). The approach provides the context to ensure that the knowledge acquired and represented is the knowledge required to support task performance.

- O'Hara and al. [45] 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 contextualizes 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. The context is largely determined by the case which prompted the change to the knowledge base [21].

Knowledge has not to be generalized when acquired. It is fundamental to record the context in which the knowledge is acquired.

However, the acquisition of knowledge in context is still a challenge. 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 [17]. Such a situated knowledge is not acquired with the classical tools prior its use. Acquiring the knowledge when needed implies that the knowledge is compatible with the computational medium of the system. 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 that is not known by the machine.

Moreover, making explicit context permits to examine knowledge in any other context. One advantage is that it allows different contexts to be compared, in particular how correcting knowledge in a particular context may be used to review conclusions drawn in another context.

Although the approaches described in this section yet present serious weaknesses, they are ascribed in the realm of the incremental knowledge acquisition in context [10].

2.2 Context in 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 concepts. Incremental concept learning in such domains requires the ability to recognize and adapt to such changes. Widmer [56] presents a general two-level learning model, and its realization in the METAL(B) system. The system learns to detect certain types of contextual clues, and reacts accordingly with context changes. The model consists of a basic learner that perfoms the regular on-line learning and classification task, and a meta-learner that identifies potential contextual clues. The operational definition of contextual attributes is based on the notion of predictive features. A feature is considered as a contextual clue if it does not directly determine or influence the class of an object. (Thus, it does not intervene in the learning explicitly.) However, 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. (Thus, contextual clues contrain the learning.) In a similar spirit, Turney [53] 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 [46] 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 17% and increases the overall amount of improvement to the expert system by 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. 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. Again, the context does not intervene directly in the learning, but constrains it.) 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.

With such approaches, machine learning is considered as part of the task at hand and appears very close of incremental knowledge acquisition.

2.3 Context in communication

In a report of a workshop on context held at CHI'89, Maskery and Meads [40] 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. For this community, 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 al. [41] 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. This is 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 permits to individualize the user interface** according to the current task. For providing a context-sensitive help, a system must be able to answer a 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 [43] 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 acknowledge the importance of 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, but **context is used in an implicit way**.

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 [14], 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 communication failure consists of the illusion of sharing a context that creates misunderstandings.

Making context explicit would also permit to revised some well-known paradigms in AI. For instance, Reichman [49] 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 previous context space is clear. If all previous topics have been fully developed (all criteria for the development of particular conversational move satisfied), then the new move

> may begin an entirely topic. Conversely, if 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 [31] 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). One thus considers **context 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. 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 [32] considers two different kinds of transition between contextual modules: learned (context-specific transitions) and general (associative transitions).

2.4 Context in databases and ontologies

The main role of context here 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 [3].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 [2]. 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 al. [30] 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. A context mediator is used to compare the source and receiver contexts and detect any conflicts. **The context mediator acts as a contextualization process.** 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 et al. [50] 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 a vehicle for users who are interested in modifying their import contexts dynamically as queries are formulated.

Walther et al. [55] 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 the context-definition language called MODEL. Then, 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.

The context mechanism was introduced into the CYC system in May 1990 to simplify the construction of the commonsense-knowledge base. In November 1991, CYC contains over 1.5 million sentences and covers a wide range of phenomena [34]. Contexts are considered as rich objects in a first-order framework, extending the logic as required [38]. The basic change is that formulas are not just true or false; **formulas 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 shared. 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. **The use of different contexts permits to use different languages**.

Huhns et al. [35] 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: the process of representing predefined groupings of concepts; (3) its knowledge representation mechanisms: a global context can be constructed, represented, and maintained, and (4) its typing mechanism: the integration and checking of the consistency of query results are ensured.

Based on McCarthy's work on context logic, Farquhar et al. [26] 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.

3 CONTEXT, MODELING AND REPRESENTATION

3.1 Context in logic representation

A number of applications exist along the ATMS approach developed by de Kleer [22]. 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.

More recently, McCarthy [42] defined a context as a generalization of a collection of assumptions. Contexts are formalized as first class objects (formal objects), and the basic relation is *ist(c,p)*. It asserts that the proposition p is true in the context c, where c is meant to capture all that is not explicit in p that is required to make p a meaningful statement representing what it is intended to state. Formulas ist(c,p) are always asserted within a context, i.e., something like ist(c', ist(c,p)): c': ist (c, p). The consequences are:

(1) **a context is always relative to another context**,

(2) **contexts have an infinite dimension**,

(3) **contexts cannot 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 a 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 that 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 that represents the focus of the reasoning--the set of assumptions made or path taken by a reasoner in evaluating a current hypothesis.

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 [15]. For instance, the diagnostic expert system SEPT dealt with pieces of equipment as circuit breakers and protective relays [6]. 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 [16,23] 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 [6] 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 [16]:

IF
1. The infection that 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 triggering of the rule. The clause acts as a screening clause and implies that this rule is only valid for an adult. Such **a screening clause acts as a context**: it does not intervene directly in the infection identification, just constrains it.

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.

It seems that this process of contextual instantiation of the variables plays an important role. In linguistics, Récanati [48] 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 [34], the bridging rules [29], the pragmatic rules [28], the ripple-down rules [20].

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 represent context (i.e. account for context), not model 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 considered on the basis of the Situation Theory [1]. Situation theory is a unified mathematical theory of meaning and information content that is applied to specific areas of language, computation and cognition. The theory provides a system of abstract objects that make it possible to describe the meaning of both expressions and mental states in terms of the information they carry about the external world. Surav and Akman [52] approach context as an amalgamation of grounding situation and the rules that govern the relations within the context. They represent a context by a situation type that 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, Nardi [44] presents a study of context from a comparison of the activity theory [39], the situation action models [37] and the distributed cognition [27].

Ezhkova [25] 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. Ezhkova [26] 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 contextual system 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].

4 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 cooperation, 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 domains where activities implies reasoning and interpretation, and can only be caught by experience. This interest for the use of context implies that there is no clear and general definition of context. Context seems to possess, according to the domain, 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

[9]. 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 [42] 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. Along a different approach, Giunchiglia [29] considers a set of discrete contexts (then, at a same level) and the relationships between them (through bridging rules for enter and leave contexts), the context making reasoning local.

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

McCarthy [42] 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 (e.g., see [29]). A solution for ensuring a correct transfer of information from one context to another may be a context manager. A context manager makes compatible the interpretations (or reasonings) in the source and destination contexts. This is indeed more **a process of contextualization** that a part of the knowledge. The context manager 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 rather than its representation (or 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. However, it stays to make the different contexts compatible.

Contexts define when the knowledge should be considered and thus simplifies the construction of the knowledge base by imposing requirements on the representation language. These requirements 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. Thus, context-encapsulated knowledge appears as a chunk of reasoning. A challenge here is how knowledge in its context of use may be examined in other contexts (a kind of knowledge decontextualization).

However, 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. Contextual knowledge intervenes implicitly in the problem solving, most often as constraints. At the following step of the problem solving, contextualized knowledge may become contextual knowledge at the new step or external to the step. An operational definition of context requires to take into account if which speak of the problem solving as a whole or of a given step of the problem solving.

In the formula ist(C P), MacCarthy [42] 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. A first step towards an operational definition of context is to define context as what constrains a problem solving without intervening in it explicitly.

In artificial intelligence, the efforts concern mainly the representation of context nested with the knowledge. 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, context and knowledge representation, context and time, context and decision? What are the relationships between contextualization process and control knowledge? The discussion stays still open.

REFERENCES

[1] Barewise, J.—Perry, J.: Situations and Attitudes. Cambridge (USA), MIT Press 1983.

[2] Barthe, M.: ERGO-METH: Principles of a methodology of computerization aiming at integrate results of cognitive ergonomy in an approach of design to improve the utility of interactive software in management. (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 1991.

[3] Bastien, C.: The shift between logic and knowledge. (Le décalage entre logique et connaissances) Le Courrier du CNRS, Numéro Spécial “Sciences Cognitives”, 1992, No.79, pp. 38.

[4] Bianco, L.—Dell'omo, P.—Ezhkova, I.V.: Application of contextual technology for

supporting decision making in transportation. Proceedings of the 7th IFAC/IFORS Symposium on Transportation Systems, Tianjin, China, 1994, pp. 363-368.

[5] Bloom, C.—Bullemer, P.—Chu, R.—Villano, M.: 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, 1992, Vol.1, pp. 509-514.

[6] Brézillon, P.: METAL: a language for structured knowledge-based systems. Proceedings of the IJCAI-91 Workshop Software Engineering for Knowledge-Based Systems, Sydney, Australia 1991, pp. 11-22.

[7] Brézillon, P.: Context in human-machine problem solving: A survey. Technical Report 96/29, LAFORIA, October 1996, 37 pages. (The paper can be retreived at ftp://ftp.ibp.fr/ibp/ reports/laforia.96/laforia.96.29.ps)

[8] Brézillon, P.: Context in Artificial Intelligence: II. Key elements of contexts. Computer and AI (submitted).

[9] Brezillon, P.—Abu-Hakima, S.: Using Knowledge in its context: Report on the IJCAI-93 Workshop. AI Magazine, Vol.16, 1995, No.1, pp. 87-91.

[10] Brézillon, P.—Cases, E.: Cooperating for assisting intelligently operators. Proceedings of the International Workshop on the Design of Cooperative Systems, INRIA, France 1995, pp. 370-384.

[11] Brézillon, P.—Pomerol, J.-Ch.: Misuse and nonuse of knowledge-based systems: The past experiences revisited. In: Humphreys P. et al. (Eds.): Implementing Systems for Supporting Management Decisions, Chapman and Hall, ISBN 0-412-75540-8, 1996 pp. 44-60.

[12] Brézillon, P.—Pomerol, J.-Ch.: User acceptance of interactive systems: Lessons from knowledge-based and decision support. International Journal of Failure & Lessons Learned in Information Technology Management, Vol. 1 1996, No.1, pp. 67-75.

[13] Brézillon, P.—Cavalcanti, M.: Modeling and using context: Report on the First

International and Interdisciplinary Conference (CONTEXT-97). The Knowledge Engineering Review (forthcoming).

[14] Cahour, B.—Karsenty, L.: 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 1993, pp. 20-29.

[15] Chandrasekaran, B.—Johnson, T.R.—Smith, J.W.: Task-structure analysis for knowledge modeling. Communications of the ACM, Vol. 3, 1992, No. 9, pp. 124-137.

[16] Clancey, W.J.: The epistemology of a rule-based expert system: A framework for explanation. Artificial Intelligence Journal, Vol. 20, 1983, No. 3, pp. 197-204.

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

[18] Clancey, W.J.: Notes on “Epistemology of a rule-based expert system”. Artificial Intelligence Journal, 59, 1993, pp. 197-204.

[19] Compton, P.—Jansen, B.: Knowledge in context: A strategy for expert system maintenance. In: J.Siekmann (Ed): Lecture Notes in Artificial Intelligence, Subseries in Computer Sciences, Vol. 406, 1988.

[20] Compton, P.—Jansen, R.: A philosophical basis for knowledge acquisition. Knowledge Acquisition, 2, 1990, pp. 241-257.

[21] Compton, P.—Yang, W.—Lee, M.—Jansen, B.: Cornerstone cases in a dictionary approach to rule maintenance. Proceedings of the IJCAI'91 Workshop on Software Engineering for Knowledge-Based Systems, August 1991, pp. 24-40.

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

[23] Eklund, P.: Prospects for conceptual graphs in acquisition interfaces. Proceedings of EKAW89, Paris, France, July 1989, pp. 169-179.

[24] Ezhkova, I.: (1989) "Knowledge formation through context formalization", Computers and Artificial Intelligence, Vol. 8, 1989, No. 4, pp. 305-322.

[25] Ezhkova, I.V.: Contextual systems: Is it a way of a universal expert system development?. In: General systems, G. Klir Publisher, New Jersey, USA 1992.

[26] Farquhar, A.—Dappert, J.—Fikes, R.—Pratt, W.: Integrating information sources using context logic. KSL-95-12, Knowledge Systems Laboratory (USA), January 1995.

[27] Flor, N.—Hutchins, E.: 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 1991, pp. 36-59.

[28] Girotto, V.: Reasoning on deontic rules: the pragmatic schemas approach. Intellectica, 1991, 1, 1991, pp. 15-52.

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

[30] Goh, C.H.—Madnick, S.E.—Siegel, M.D.: Context interchange: Overcoming the challenges of large-scale interoperable database systems in a dynamic environment. 1995, In Proceedings of the Third International Conference on Information and Knowledge Management, Gaithersburg (USA) 1994, pp. 337--346.

[31] Grant, A. S.: Mental models and everyday activities. Proceedings of the 2nd Interdisciplinary Workshop on Mental Models, Cambridge, UK, March 1992, pp. 94-102.

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

[33] Gruber, T.: Justification-based knowledge acquisition. In: Motoda, H.—Mizoguchi, R.—Gaines B. (Eds.): Knowledge Acquisition for Knowledge-Based Systems, IOS Press, 1991.

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

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

[36] Jansen, B.—Grosz, G.: The knowledge dictionary: representing contextual information. Technical Report TR-FD-90-4, CSIRO Division of Information Technology, PO Box 1599, North Ryde, NSW, Australia 1990.

[37] Lave, J.: Cognition in Practice. Cambridge University Press 1988.

[38] Lenat, D.: Context dependence of representations in CYC. Colloque ICO'93, Montréal, Canada May 1993.

[39] Leont'ev, A.: The problem of activity in psychology. Soviet Psychology, Vol. 13, 1978, No. 2, pp. 4-33.

[40] Maskery, H.—Meads, J.: Context: In the eyes of users and in computer systems. SIGCHI Bulletin, Vol. 24, 1992, No. 2, pp. 12-21.

[41] Maskery, H.—Hopkins, G.—Dudley, T.: Context: What does it mean to application design. SIGCHI Bulletin, Vol. 24, 1992, No. 2, pp. 22-30.

[42] McCarthy, J.: Notes on formalizing context. Proceedings of the 13th IJCAI, Vol. 1, 1993, pp. 555-560.

[43] Mittal, V.O.—Paris, C.L.: Use of context in explanations systems. International Journal of Expert Systems with Applications, Vol. 8, 1995, No. 4, pp. 491-504.

[44] Nardi, B.A.: 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, 1992, pp. 352-359.

[45] O'Hara, K.—Shadbolt, P.—Laublet, P.—Zacklad, M.—Leroux, B.: VITAL, A methodology-based workbench for KBS life cycle support. ESPRIT II Project intermediary report VITAL/DD212, September 1992.

[46] Park, Y.-T.—Wilkins, D.C.: Establishing the coherence of an explanation to improve refinement of an incomplete knowledge base. Proceedings of the AAAI90 1990.

[47] Paton, R.—Shave, M.—Bench-Capon, T.—Nwana, H.: 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 1993.

[48] Récanati, F.: The linguistic pragmatics. (La pragmatique linguistique) Le Courrier du CNRS, Numéro Spécial “Sciences Cognitives”, 79, 1992, pp. 21.

[49] Reichman, R.: Getting computers to talk like you and me. Discourse context, focus, and semantics. A Bradford Book, The MIT Press, Cambridge, MA 1985.

[50] Sciore, E.—Siegel, M.—Rosenthal, A.: Context interchange using meta-attributes. Proceedings of the 1st International Conference on Information and Knowledge Management, 1992, pp. 377-386.

[51] Srinivasan, A.—Compton, P.—Malor, R.—Edwards, G.—Lazarus, L.: Knowledge acquisition in context for a complex domain. Pre-print of Proceedings of the Fifth EKAW91 1991.

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

[53] Turney, P.: The management of context-sensitive features: A review of strategies. Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains, 1996, pp. 53-69.

[54] Vanwelkenhuysen, J.—Mizoguchi, R.: 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 1995 , pp. 167-177.

[55] Walther, E.—Eriksson, H.—Musen, M.A.: Plug-and play: Construction of task-specific

expert-system shells using sharable context ontologies. Proceedings of the AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition, San Jose, CA, pp. 191-198. AAAI, 1992. Technical Report KSL-92-40, Stanford (USA) 1992.

[56] Widmer, G.: Recognition and exploitation of contextual clues via incremental meta-learning (Extended version). Oesterreichisches Forschungsinstitut fuer Artificial Intelligence, Wien, TR-96-01, 1996

Patrick BRÉZILLON

LIP6, Case 169

University Paris 6,

4 place Jussieu

75252 Paris Cedex 05

France