*2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,*
*Proceedings, Companion Volume, pp. 27-38*

# Context in Artificial Intelligent and Information Modeling

Manos Theodorakis[1] and Nicolas Spyratos[2]

[1] FIT-Fraunhofer Institute for Applied Information Technology,
D-53754, Sankt Augustin, Germany
manos.theodorakis@fit.fraunhofer.de
[2] Universite de Paris-Sud, LRI-Bat 490, 91405 Orsay Cedex, France
spyratos@lri.fr

**Abstract.** The modeling, representation and use of context is the challenge for the coming years in Artificial Intelligence, especially when we now face very large knowledge bases, complex problems and multimedia means. The notion of context is important since it can capture many of the interesting aspects of the way we understand the world, such as relativity, locality, partiality, and context-dependence. In Artificial Intelligence, a number of formal or informal definitions of some notion of context have appeared in several areas. However, all these notions of context are very diverse and serve different purposes. In this paper, we give an overview of the notion of context in Artificial Intelligence and we focus in information modeling to present our approach of how to model context. In particular, we present a context-based model for structuring and accessing information in large information bases.

**Keywords**: Context, contextualization, knowledge representation, information modeling, contextual knowledge, contextualized knowledge, abstraction mechanisms.

## 1 Introduction

The notion of context plays an important role in a number of domains and it is of fundamental importance in cognitive psychology, linguistics, and computer science. In computer science, a number of formal or informal definitions of some notion of context have appeared in several areas, such as artificial intelligence [17, 20, 15], software development [25, 14], databases [2, 13], machine learning [23], and knowledge representation [24, 35, 39, 34, 40].

However, all these notions of context are very diverse and serve different purposes. In software development the notion of context appears in the form of views [2, 13], aspects [25], and roles [14], for dealing with data from different perspectives, or even in the form of workspaces which are used to support cooperative work [19]. In machine learning, context is treated as environmental information for concept classification [23]. In the so called multiple databases, context appears as a collection of meta-attributes for capturing class semantics [18]. In artificial intelligence, the notion of context appears as a means of partitioning knowledge into manageable sets [17], or as a logical construct that facilitates reasoning activities [20, 15]. In particular, in the area of knowledge representation, the notion of context appears as an abstraction mechanism for partitioning an information base into possibly overlapping parts (e.g. [24, 34, 39, 40, 11, 31, 9, 8]).

The modeling, representation and use of context is the challenge for the coming years, especially when we now face very large knowledge bases, complex problems and multimedia means. The notion of context is important since it can capture many of the interesting aspects of the way we understand the world, such as relativity, locality, partiality, and context-dependence. In Artificial Intelligence, the lack of explicit representation of context is one of the reasons of the failure of many Knowledge-Based Systems [5].

In this paper, we examine different representations of the notion of context in Artificial Intelligence and we present our approach of how to model context. Finally, we compare all these approaches based on specific criteria on modeling contexts.

In this paper, we review the notion of context in different areas in Artificial Intelligence and we present an approach of modeling the notion of context. In particular, we present a context-based model for structuring and accessing information in large information bases. A context is seen as a set of objects within which each object has a set of names and optionally a reference: the reference of the object is just another context which "hides" detailed information about the object.

In Section 2, we review different formalizations of context in Artificial Intelligence. In Section 3, we focus on the area of information modeling and we present our approach of modeling the notion of context, and in Section 4, we compare these approaches according to specific criteria for modeling context.

## 2 Context in Artificial Intelligence

### 2.1 Situation Theory

Contexts has been considered in the basis of Situation Theory [3]. 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 deals with situations and meaning (as a relation between situations). The notion of context is represented by *situations*. Situations are first-class citizens of the theory and are defined intentionally. A situation is considered to be a structured part of the reality, that an agent (somehow) manages to pick out, and to which sentences in a theory are stated. Specifically, the work of Barwise and Perry on situated reasoning and semantics [3] is motivated by interest in dealing with efficiency of language (all qualifications for ”John can run” are captured in the context of the sentence's utterance), partiality of information (the symbol ”flower” is meaningless in a theory of computer network), and self-reference (the utterance ”This is an embarrassing moment” would be made in some situation $s$ but clearly refers to the same situation $s$).

Surav and Akman [30] (mainly inspired by Barwise and Perry's work) approaches context as an amalgamation of grounding situation and the rules that govern the relations within the context. They present 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.

### 2.2 Situation Calculus

The simplest approach to representing that the value of some predicate or function symbol is dependent on some situation or context is to add a context argument to the list of arguments for each predicate and function in the theory [22]. For example, the predicate $on(block_1, block_2)$ is written as $on(block_1, block_2, s_1)$. This allows one to stay within a classical first-order framework and capture a simple form of context-relativity for formulae. While adequate for many pragmatic cases for contexts, this approach is inadequate for dealing naturally with structured spaces of contexts, context-specific vocabularies and context inheritance. A problem, for example, of this approach is that it does not allow a predicate to have different arguments in different contexts as it does not provide adequate namespace separation between theories.

### 2.3 Context in first-order logic

Motivated by the observation that one can never represent an object in complete generality, McCarthy [20] introduced the notation $ist(c, p)$ (pronounced as "is true") meaning that a logical sentence $p$ holds 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 represented what it is intended to state. Formulas $ist(c, p)$ are always asserted within a context, i.e., something like $ist(c', ist(c, p))$. The most important in this theory the writing of axioms describing and interrelating contexts. Then, the most common operation is to lift a formula from one context into another. Doing this requires the differences between the origin and target contexts to be taken into account to obtain a formula with the same truth conditions as the origin formula had in the origin context. There are many other relations among contexts, for example, the relation $specialize(c_1, c_3)$, that indicates $c_2$ involves no more assumptions than $c_1$ and every proposition meaningful in $c_1$ is translatable into one meaningful in $c_2$.

The consequences are: (i) contexts cannot be described completely, (ii) propositions and contexts are always relative to another context, (iii) contexts can be nested in any depth, (iv) contexts are related among others with different relations, (e.g., lifting axioms from one context to another, specialization of contexts).

One of the first attempts at formalizing context under McCarthys supervision was presented by Guha [15]. Motivated largely by his work in the Cyc project, an attempt to build an extremely large knowledge base to support common-sense reasoning, Guha observed that without structuring the vocabularies and sentences of the system, it was nearly impossible to construct Cyc without running into inconsistency and a proliferation of qualifications for non-logical symbols when stating sentences. Contexts also enabled the Cyc designers to state theories at different levels of detail and to have Cyc employ the appropriate theory. Thus, Guha treated a context as having a vocabulary and a set of possible truth assignments associated with it so that the formula $ist(c, p)$ holds if $p$ is true in all the valid truth assignments for context $c$. Contexts are formalized as first class objects.

Since then, contexts have found uses in various artificial intelligence applications, including: translating knowledge [6], modeling knowledge and belief [12], integrating heterogeneous databases [10], planning [7], common sense reasoning [21].

Coherently with the notion of context described above, Attardi [1] uses viewpoints to represent the notion of relativized truth such as beliefs, situations and knowledge. Viewpoints denote sets of sentences which represent the assumptions of a theory.

### 2.4 Context in categorization

Categorization is one of the basic mental processes in cognition. We, as human beings, can categorize various types of objects, events, and states of affairs, and our categorizations depend on the circumstance and perspective (i.e., how things are depend on one's point of view on them). Barwise and Sligman [4] use *natural regularities* to study the role of context in categorization. To be more specific consider the following example of regularity as it appears in [27]: 'swans are white', which express an intuitive sense that all swans are white. Although this is the general intuition about swans, there might be exceptions and we can find swans which are black (this can happen in Australia). Therefore, this sentence can be evaluated only in appropriate contexts, such as in Europe, outside zoos, and so forth. The appropriate context wouldn't be a problem if we could completely specify all contextual factors. However, in many cases it is impossible to state all the relevant contextual factors. In [4], a notion of context is captured through the notion of *perspective*. Different perspectives simply give rise to different ways of classifying things. For example, distances can be classified using either inches from Manos' perspective or centimeters from Nicolas' perspective.

### 2.5 Graphical Representation of Context

A precursory idea of context can be traced back to Peirce's *existential graphs* [26]. Existential Graphs use a logical form of context called a *cut* which shows in a topological manner the scope of a negative context on a sheet of paper (the *sheet of assertion*). Sowa [28, 29] introduced *conceptual graphs* (CG) as an extension of the existential graphs and defined *contexts* as concepts whose referent contains one or more CG. The contexts are treated as first-class objects and are embeddable in graphs. Hendrix [16, 17] expanded semantic networks (based on existential graphs) through *partitioning* contexts. Unlike Peirce, Hendrix allowed overlapping contexts.

Some problems with CG contexts are that inheritance can only follow the tree structure formed by the embedding of the contexts and runs from the child to the parents (collapsing of contexts). Furthermore, lifting axioms for contexts can only be written by resorting to the meta-level facilities of CGs, writing meta-level rules.

## 3 Context in Information Modeling

### 3.1 The notion of context

Suppose we want to talk about Greek islands by simply using their names without further description. Let us consider the island of Crete. We can represent this island by an *object identifier*, say $o_1$ associated with the name $\texttt{Crete}$. We write $names(o_1) = \{\texttt{Crete}\}$ and we denote this as follows[1]:

$$\texttt{Crete} : o_1$$

Next, let us consider the island of Santorini. Following a similar approach, we represent this island by an object identifier $o_2$ associated with the name $\texttt{Santorini}$. However, the island of Santorini is also known under the name Thera. So this time, we associate $o_2$ with the set of names $\{\texttt{Santorini}, \texttt{Thera}\}$, i.e. $names(o_2) = \{\texttt{Santorini}, \texttt{Thera}\}$, and we denote this as follows:

$$\texttt{Santorini}, \texttt{Thera} : o_2$$

Finally, let us consider one of those tiny, uninhabited islands of Greece that happen to be nameless. We represent such an island by an object identifier $o_3$ associated with no name, i.e. $names(o_3) = \{\}$, and we denote this as follows:

$$: o_3$$

Continuing in the same way, we can represent every Greek island in a similar manner. The set of all such representations is a *context* that we represent by a *context identifier*, say $c_1$.

Suppose next we want to talk about the Greek mainland by simply using the names of each region of Greece without further description. Proceeding in a similar way as in the case of Greek islands, we can create a second context, say $c_2$, as shown in Figure 1.

Suppose now that we want to talk about the geography of Greece seen as a division of Greece into islands and mainland. First, let us consider the islands. We can represent the islands by an object identifier, say $o$, and associated with the name $\texttt{Islands}$. However, the object $o$ is a higher level object that collectively represents all Greek islands, i.e. the object $o$ collectively represents the contents of context $c_1$. In other words, if we want to see what $o$ means at a finer level of detail, then we have to "look into" the contents of $c_1$. Thus we call context $c_1$ the *reference* of object $o$, and we write $ref(o) = c_1$. Summarizing our discussion on islands, we write $names(o) = \{\texttt{Islands}\}$ and $ref(o) = c_1$, and we denote this as follows:

$$\text{Islands} : \text{o} \cdots\cdots\triangleright\ \text{c}_1$$

Following a similar reasoning, we can represent the mainland by an object identifier, say $o'$ associated with the name $\texttt{Mainland}$ and the reference $c_2$. We can now group together the islands and the mainland to form a context $c$, as shown in Figure 1. Then, geography of Greece can be represented by an object identifier $o''$ associated with context $c$, as shown in Figure 1.

The previous examples suggest the following definition of a context [35, 32]:

**Definition 31 Context.**

A context $c$ is a set of objects such that each object $o$ is associated with

1. a set of names, called *the names of $o$ in $c$*, and denoted by $names(o, c)$;

[1] In this paper, the terms *object* and *object identifier* will be used interchangeably.

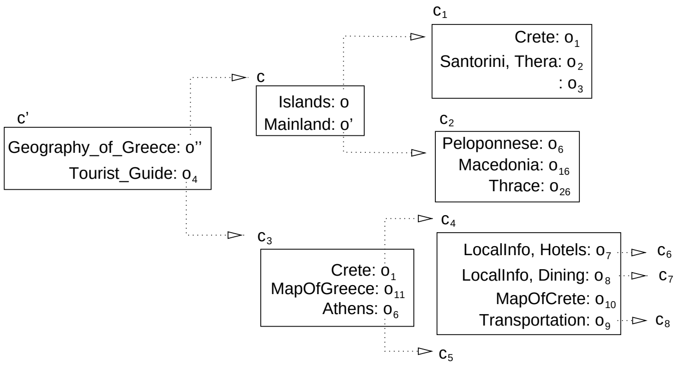


**Fig. 1.** An example of context structure.

2. zero or one context, called *the reference of* $o$ *in* $c$, and denoted by $ref(o, c)$.

The reason why we use the symbols $names(o, c)$ and $ref(o, c)$ in the above definition, instead of the symbols $names(o)$ and $ref(o)$ used in the previous examples, is that an object can belong to different contexts and may have different names and/or reference in each context. That is, *names and references of an object are context-dependent.*

In our previous examples, while explaining the construction of a context, we followed a bottom-up approach. That is, we started from simple objects and built up contexts which were later on referenced by higher level objects ("moving" from right to left in Figure 1). Clearly, we could have followed the opposite construction, i.e. a top-down approach ("moving" from left to right in Figure 1). In fact, we can even follow a mixed approach, i.e., creating each context independently, then connecting them through references. This flexibility is important in conceptual modeling and implies (among other things) the possibility of *modular design*, i.e. retaining at each level of abstraction the essential information and hiding inessential details (by putting them in a referenced context).

It is important to note that an object can have two or more names within a context and that two different objects can have one or more names in common. Moreover, an object can have no name within a context. In other words, our contextualization mechanism supports *synonyms*, *homonyms*, and *anonyms*. It is also important to note that our contextualization mechanism does not depend in any particular data model, but it can be easily embedded in any one [35, 32]. For example, the objects of a context can be objects from an object-oriented database, or tuples from a relational database, or the results of queries in an object-oriented or a relational database, and so on. This is why we do not specify the nature of the objects in a context.

Let us now consider another example, shown in Figure 2, which represents the views of two persons regarding an institute. Context $IB$ contains two objects, $o_{10}$ and $o_{11}$, namely `ManosView` and `AnastasiaView`, respectively. These objects represent the views of Manos and Anastasia regarding the same institute. These views are presented in detail within contexts $c_1$ and $c_4$ (the references of objects $o_{10}$ and $o_{11}$, respectively).

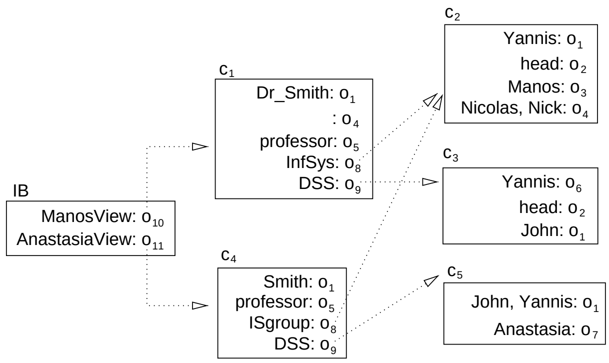


**Fig. 2.** An example of context structure: Two views of the same institute.

Note that object $o_1$ (which represents a person) is shared by several contexts ($c_1$, $c_2$, $c_3$, $c_4$, and $c_5$) and is assigned different names within each of these contexts. Note also that object $o_8$ (which represents the Information Systems Lab of the institute) refers to the same context $c_2$ either within context $c_1$ or $c_4$. Intuitively, this expresses that Manos and Anastasia have the same view for the Information Systems Lab. Whereas object $o_9$ (which represents the Decision Support Systems Lab) refers to context $c_3$ within context $c_1$ and to context $c_5$ within context $c_4$. Intuitively, this expresses that Manos and Anastasia have different views for the Decision Support Systems Lab.

Let us summarize the features of context supported by context definition:

1. *Object sharing or overlapping contexts*: An object can belong to one or more different contexts. When contexts share objects we say that contexts overlap. This feature is useful when we want to view an object under different perspectives.
2. *Context-dependent object names*: The same object can have different names in different contexts (in which it belongs). This is very convenient, because a name which has a clearly understood meaning in one context may not do so in another.
3. *Synonyms*: The same object can have different names in the same context. That is, alternative ways for naming the same object are supported. This is the case of synonymous objects.
4. *Homonyms*: Two different objects can have the same name within a context. This is the case of homonymous objects.
5. *Anonyms*: An object may have no name within a context. This is the case of anonymous objects.
6. *Context-dependent references*: The same object can have different references within different contexts. In other words, references are context-dependent.
7. Two different objects, whether or not they belong to the same or different contexts, can have the same reference. This is convenient, as a given context can be reachable through different object paths.

8. From within a given context, we can “reach” any object that belongs to the reference of an object within that context (and, recursively, any object that lies on a path).

A prominent feature of our contextualization mechanism is that it allows users to focus on a specific context at a time (call it *current context*), thus delimiting a portion of interest in the information base. As a consequence, the scope of user queries is localized to that portion, i.e., to the set of objects and contexts that are accessible from the current context. In turn, query evaluation is performed with respect to that portion of interest — and *not* with respect to the whole information base. As a result, users can find speedily the needed information.

An other important feature is that users are able to make cross references of an object from one context to another in order to obtain alternative representations of that object. Note that, in first-order logic, to make cross references it is needed to defined in an outer context lifting axioms. In this theory, this is done through the object identity.

### 3.2 Operations on context

Moreover, we define a set of operations for manipulating contexts [38, 34, 33]. These operations support context creation, update, copy, union, intersection, and difference. In particular, our operations of context union, intersection, and difference are different from these of set theory as they keep track of the context involved. However, they also satisfy the important properties of commutativity, associativity, and distributivity. Our model contributes to the efficient handling of information, especially in large information systems, and in distributed, cooperative environments, as it enables (i) representing (possibly overlapping) partitions of an information base; (ii) partial representations of objects, (iii) flexible naming (e.g. relative names, synonyms and homonyms), (iv) focusing attention, (v) handling inconsistent information, and (vi) combining and comparing different partial representations.

### 3.3 Structuring the contents of a context

Then, we show how a particular semantic data model (the Telos data model) can be incorporated into the proposed contextualized framework [35, 36]. Thus, we enhance our notion of context by structuring its contents through the traditional abstraction mechanisms, i.e., classification, generalization, and attribution. We show that, depending on the application, our notion of context can be used either as an alternative way of modeling or as a complement of the traditional abstraction mechanisms. It is important that we study the interactions between contextualization and the traditional abstraction mechanisms as well as the constraints that govern such interactions. In that study, we define also the relation *refinement* between context to support a kind of inheritance of their contents.

### 3.4 Querying contextualized information

Contextualized information bases need a special treatment in order to answer queries. Thus, we propose a general framework for querying information bases which supports contextualization [37]. In particular, we focus on the following issues: (i) accessing information in a context structure using paths of names or paths of references, (ii) retrieval of contextualized information by defining useful fundamental query operations on contexts such as select, project, generate (which allows the reorganization of contexts structure), and path select.

In addition to the fundamental operations there are several other derived operations, such as *context union*, *context intersection*, and *context difference*. We extend the functionality of querying by allowing traversal of the context hierarchy using complex path expressions. Finally, we illustrate the usefulness of our contextualization mechanism by presenting higher level query operations that enable users to explore a contextualized information base. These higher level operations include focusing on a context of interest, searching the context structure for specific information, and making cross references of a concept from one context to another in order to obtain alternative representations of that object.

## 4 Comparison

It is unlikely that there will be a single context formalism that will suffice for all forms of context, just as there is no universal knowledge representation language. The following criteria have been identified for comparing the different approaches for formalizing context:

CFCO: contexts as first class objects

CSV: context specific vocabulary

CSS: context specific semantics

Nest: nesting of contexts (subcontexts)

Inh: inheritance through contexts

SR: self-reference

Table 1 shows a comparison of the main approaches for formalizing the notion of context examined in this paper (ST: Situation Theory, SC: Situation Calculus, FOL: first-order logic, CG: Conceptual Graphs, IM: Information Modeling).

| | ST | SC | FOL | CG | IM |
|---|---|---|---|---|---|
| CFCO | ✓ | ✓ | ✓ | | ✓ |
| CSV | ✓ | | ✓ | | ✓ |
| CSS | ✓ | ✓ | ✓ | ✓ | ✓ |
| Nest | ✓ | | | ✓ | ✓ |
| Inh | ✓ | | | | ✓ |
| SR | ✓ | | | | ✓ |

**Table 1.** Comparison of properties of context

## 5 Conclusions

In this paper, we review different formalizations of contexts in Artificial Intelligence and we present our approach of modeling the notion of context in information modeling. Finally, we give a comparison of the formalizations presented according to specific criteria for context modeling.

## References


1. G. Attardi and M. Simi. A Formalization of Viewpoints. *Fundamenta Informaticae*, 23(2-4):149–174, 1995. Also Technical Report, International Computer Science Institute, Berkeley, TR-93-062, October, 1993.
2. F. Bancilhon and N. Spyratos. Update Semantics of Relational Views. *ACM Trans. Database Syst.*, 6(4):557–575, Dec. 1981.
3. J. Barwise and J. Perry. *Situations and Attitudes*. MIT Press, Cambridge, Mass.: Bradford Books, 1983.
4. J. Barwise and J. Seligman. The Rights and Wrongs of Natural Regularity. *Philosophical Perspectives*, 8:331–365, 1994.
5. P. Brezillon and J. Pomerol. Misuse and nonuse of knowledge-based systems: The past expreriences revisited. In P. Humphreys, L. Bannon, A. McCosh, P. Migliarese, and J.-C. Pomerol, editors, *Implementing Systems for Supporting Management Decisions*, pages 44–60. Capman and Hall, 1996.
6. S. Buvač and R. Fikes. A Declarative Formalization of Knowledge Translation. In *Proc. of the ACM CIKMK: The 4th Int. Conf. on Information and Knowledge management*, 1995.
7. S. Buvač and J. McCarthy. Combining Planning Contexts. In A. Tate, editor, *Advanced Planning Technology - Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, pages 84–88. AAAI Press, 1996.
8. L. Campbell, T. Halping, and H. Proper. Conceptual Schemas with Abstractions: Making Flat Conceptual Schemas More Comprehensible. *DKE*, 20(1):39–85, June 1996.
9. B. Czejdo and D. Embley. View Specification and Manipulation for a Semantic Data Model. *IS*, 16(6):585–612, 1991.
10. A. Farquhar, A. Dappert, R. Fikes, and W. Pratt. Integrating Information Sources Using Context Logic. In *Proc. of AAAI Spring Symposium on Information Gathering from Distributed Heterogeneous Environments*, 1995. Also Technical Report, Knowledge Systems Laboratory, KSL-95-12, January 1995.
11. P. Feldman and D. Miller. Entity Model Clustering: Structuring a Data Model by Abstraction. *Computer J.*, 29(4):348–360, Apr. 1986.
12. F. Giunchiglia. Contextual Reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, XVI:345–364, 1993.
13. G. Gottlob, P. Paolini, and R. Zicari. Properties and Update Semantics of Consistent Views. *ACM Trans. Database Syst.*, 13(4):486–524, December 1988.
14. G. Gottlob, M. Schrefl, and B. Röck. Extending Object - Oriented Systems with Roles. *ACM Trans. Inf. Syst.*, 14(3):268–296, July 1996.
15. R. Guha. *Contexts: A Formalization and Some Applications*. PhD thesis, Stanford University, 1991. Also published as Technical Report STAN-CS-91-1399-Thesis, and MCC Technical Report Number ACT-CYC-423-91.
16. G. Hendrix. Expanded the Utility of Semantic Networks through Partitioning. In *Advance Papers of the IJCAI'75, Int. Joint Conf. on AI*, pages 115–121, 1975.

17. G. Hendrix. Encoding Knowledge in Partitioned Networks. In N. Findler, editor, *Associative Networks*. New York: Academic Press, 1979.
18. V. Kashyap and A. Sheth. Semantic and Schematic Similarities between Database Objects: A Context-Based Approach. *VLDB Journal*, 5(4):276–304, Dec. 1996.
19. R. Katz. Towards a Unified Framework for Version Modeling in Engineering Databases. *ACM Comput. Surv.*, 22(4):375–408, Dec. 1990.
20. J. McCarthy. Notes on Formalizing Context. In *Proc. IJCAI-93*, pages 555–560, Chambery, France, 1993.
21. J. McCarthy and S. Buvač. Formalizing Context (Expanded Notes), 1994. Technical Note STAN-CS-TN-94-13, Stanford University, Available from http://sail.stanford.edu.
22. J. McCarthy and P. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, 1969.
23. R. Michalski. How to Learn Impressive Concepts: A Method Employing a Two-Tiered Knowledge Representation for Learning. In *Proceedings of the 4th International Workshope in Machine Learning*, pages 50–58, Irvine, CA, 1987.
24. J. Mylopoulos and R. Motschnig-Pitrik. Partitioning Information Bases with Contexts. In *Proceedings of CoopIS'95*, pages 44–55, Vienna, Austria, 1995.
25. J. Richardson and P. Schwarz. Aspects: Extending Objects to Support Multiple, Independent Roles. In *Proc. ACM-SIGMOD Conference*, pages 298–307, Denver, Colorado, May 1991.
26. D. Roberts. *The Existential Graphs of Charles S. Peirce*. The Hague, Mouton, 1973.
27. J. Seligman. The Structure of Regularity, 1993. Draft Lecture Notes, Indiana University Logic Group, Bloomington, IN.
28. J. Sowa. *Conceptual Structures*. Reading Mas: Addison Wesley, 1984.
29. J. Sowa. Syntax, Semantics, and Pragmatics of Context. In *Proc. 3rd Int. Conf. on Conceptual Structures, ICCS'95*, Lecture Notes in AI 954, pages 1–15, Santa Cruz, CA, USA, Aug. 1995. Springer.
30. M. Surav and V. Akman. Modeling Context with Situations. In P. Brezillon and S. Adu-Hakima, editors, *Proc. of the IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning*, pages 145–156, Paris, France, 1995.
31. T. Teorey, G. Wei, D. Bolton, and J. Koenig. ER Model Clustering as an Aid for User Communication and Documentation in Database Design. *Commun. ACM*, 32(8):975–987, Aug. 1989.
32. M. Theodorakis. *Contextualization: An Abstraction Mechanism for Information Modeling*. PhD thesis, Department of Computer Science, University of Crete, June 2001.
33. M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. A Theory of Contexts in Information Bases. Technical Report 216, Institute of Computer Science Foundation for Research and Technology - Hellas (ISC-FORTH), Mar. 1998.
34. M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. Context in Information Bases. In *Proceedings of the 3rd International Conference on Cooperative Information Systems (CoopIS'98)*, pages 260–270, New York City, NY, USA, Aug. 1998. IEEE Computer Society.
35. M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. Contextualization as an Abstraction Mechanism for Conceptual Modeling. In *Proceedings of the 18th International Conference on Conceptual Modeling (ER'99)*, pages 475–489, Paris, France, Nov. 1999.
36. M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. Contextualization as an Abstraction Mechanism in Conceptual Modeling. Technical Report 255, Institute of Computer Science Foundation for Research and Technology - Hellas (ISC-FORTH), Mar. 1999.
37. M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. Querying Contextualized Information Bases. In *Proceedings of the 24th International Workshop of Information and Communication Technologies and Programming (ICTP'99)*, Plovdiv, Bulgaria, June 1999.

38. M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. A Theory of Context in Information Bases. To be published in Information Systems Journal, 2002.
39. M. Theodorakis and P. Constantopoulos. Context-Based Naming in Information Bases. *International Journal of Cooperative Information Systems*, 6(3 & 4):269–292, 1997.
40. H. Weber. Modularity in Data Base System Design: A Software Engineering View of Data Base Systems. *VLDB Surveys*, pages 65–91, 1978.