UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

THEORIES AND USES OF CONTEXT IN KNOWLEDGE REPRESENTATION AND REASONING

Paolo Bouquet, Chiara Ghidini, Fausto Giunchiglia
and Enrico Blanzieri

October 2001

Technical Report # DIT-02-010

Also: published as invited paper in Journal of Pragmatics - Special Issue on Context, Elsevier Science.

Theories and uses of context
in knowledge representation and reasoning

Paolo Bouquet[1] Chiara Ghidini[2] Fausto Giunchiglia[1] Enrico Blanzieri[3]

[1]Department of Information and Communication Technology
University of Trento
Povo, I-38050 Trento, Italy

[2]Department of Computer Science, University of Liverpool,
Liverpool L69 7ZF, United Kingdom

[3]Department of Psychology, University of Torino
via Lagrange 3, I-10123 Torino, Italy

Abstract

This paper discusses the uses of context in knowledge representation and reasoning (KRR). We propose to partition the theories of context brought forward in KRR into two main classes, which we call *divide-and-conquer* and *compose-and-conquer*. We argue that this partition provides a possible explanation of why in KRR context is used to solve different types of problems, or to address the same problems from very different perspectives. The problems we use to illustrate this point are the problem of generality, the formalization of propositional attitudes, knowledge and data integration.

1 Introduction

The notion of context plays a crucial role in different disciplines, such as pragmatics, natural language semantics, linguistics, cognitive psychology, artificial intelligence (AI), and is a highly interdisciplinary field[1]. Even within AI, context is used in many different areas: in natural language

---

[1] The proceedings of the international and interdisciplinary conference on "Modeling and Using Context" [2, 10] offer a good illustration of this claim.

processing, context is used to assign an interpretation to assertions and resolve ambiguities; in information retrieval, context helps to refine the queries made by users; in distributed AI, context is used as a flexible formal tool for the design of systems of autonomous agents; in human-machine interaction, context is used to design context-sensitive applications and interfaces. We focus on the usage of context in an area of AI called *knowledge representation and reasoning* (KRR), whose aim is to devise languages for representing what (intelligent) programs or agents know about their environment, and the reasoning processes that allow them to derive new knowledge from what they already know.

In KRR, a notion very similar to context, called *LSpair* was firstly introduced by Weyhrauch in his *Prolegomena to a Theory of Mechanized Formal Reasoning* [52]. His goal was to implement the epistemological part of McCarthy's *Advice Taker*, a program which should possess abilities that in human beings would be called common sense [41]. A fundamental assumption of the Advice Taker's project was that formal logic was an appropriate tool for modeling and studying the properties of such a program. In particular, McCarthy held that the program's knowledge was to be represented as a logical theory, and that reasoning was to be modeled as inference in such a theory. Weyhrauch introduced *LSpairs* (that he later called "contexts") as a crucial device for the mechanization of these ideas: a context was thought of as a finite – are therefore suitable for being mechanized – presentation of a logical theory (theories, in logic, are infinite). Context was viewed as the building block of a theory of mechanized reasoning.

However, context became a widely discussed issue only in the late 80's / early 90's. Independently from each other, and with different motivations, Fausto Giunchiglia [29] and John McCarthy [43, 44] started to work on a formal theory of context, whose goal was to explain in a systematic way the properties of context and contextual reasoning.

Since then, context has been used in different types of applications in KRR: designing and building large common-sense knowledge-bases (see, for example, the project called CYC [39, 34], the largest knowledge base ever built); the formalization of theoretical issues concerning reasoning about beliefs [33, 28, 6, 21, 23]; the formalization of meta reasoning and propositional attitudes [32]; the formalization of reasoning with viewpoints [4]; reasoning about action [8]; modeling of different aspects of agents and multi-agent systems [7, 16]; modeling dialog, argumentation, and information integration in electronic commerce [47, 46, 26]; the integration of heterogeneous and autonomous knowledge and data bases [19, 45, 27, 50].

The goal of the paper is twofold: an analysis of the types of theory of context that have been proposed in KRR, and an application of such an analysis for describing the different uses of context in KRR. The paper goes as follows. In the first part, we suggest that two types of theory of context have been proposed in KRR: the first, which we call *divide-and-conquer*, sees context as a way of partitioning a global model of the world into smaller and simpler pieces; the second, which we call *compose-and-conquer*, sees context as a local theory of the world in a network of relations with other local theories; we also hint to a possible reduction of the first type to the second. In the second part of the paper, we show that each type of theory leads quite naturally to different uses of context, namely that some problems are more naturally addressed in a divide-and-conquer theory and others in a compose-and-conquer theory; this is illustrated by discussing examples like the problem of generality, the formalization of propositional attitudes, knowledge and data integration[2].

2 Two types of theory of context in KRR

The goal of KRR is to provide and study formal languages that can be used to represent what an agent of a certain kind can know about the world, and to show how this knowledge can be used to infer new knowledge from that already available via reasoning. With respect to this goal, many researchers seem to believe that a completely general representation of knowledge is impossible to achieve in practice, and – more interestingly – perhaps is not even desirable. Indeed, whatever language and facts we choose to represent knowledge, there is always a situation in which it is not adequate. Here's how McCarthy expressed this intuition in his well-known paper on generality in AI [43]:

> Whenever we write an axiom, a critic can say that the axiom is true only in a certain context. With a little ingenuity the critic can usually devise a more general context in which the precise form of the axiom doesn't hold. Looking at human reasoning as reflected in language emphasizes this point. Consider axiomatizing *on* so as to draw appropriate consequences from the informa-

---

[2] Our goal is not to provide an exhaustive survey of formal theories of context and of uses of context in AI. For those interested in formal theories, we refer to an excellent paper by V. Akman and M. Surav [3]; for those interested in applications, see for example a recent paper by P. Brezillon [11].

> tion expressed in the sentence, 'The book is on the table'. The critic may propose to haggle about the precise meaning of *on*, inventing difficulties about what can be between the book and the table, or about how much gravity there has to be in a spacecraft in order to use the word on and whether centrifugal force counts. Thus we encounter Socratic puzzles over what the concept mean in complete generality and encounter examples that never arise in life. There simply isn't a most general context.
>
> Conversely, if we axiomatize at a fairly high level of generality, the axioms are often longer than is convenient in special situations. Thus humans find it useful to say, 'The book is on the table', omitting reference to time and precise identification of what book and what table. [...]
>
> A possible way out involves formalizing the notion of context [...]

The quote makes clear that an important motivation for studying formal theories of context is the need for problem-tailored representations. Indeed, this should allow a reasoning system to avoid the use of unnecessarily complicated theories, and at the same time provide the ability to "jump" to a more general representation if the one in use proves to be inadequate.

The idea of using context as a tool to "localize" reasoning to a subset of the facts known by an agent (ideally, to the "right" set of facts in a given circumstance) is also part of the motivations brought forward by Giunchiglia in one of his first papers on context [29]:

> It is widely agreed on that most cognitive processes are *contextual* in the sense that they depend on a set of variables which constitute the environment (or context) inside which they are carried on [...] Our basic intuition is that reasoning is usually performed on a subset of the global knowledge base; we never consider *all we know* but only a very small subset of it [...]
>
> We take a *context c to be that* **subset of the complete state of an individual** *that is used for reasoning about a given goal.*

A third illustration of this widely shared view on context is Lenat's account of why context was introduced in CYC, the largest common sense knowledge base ever built (see Section 3.3.3) [38]:

> During the 1984-1989 time period, as the Cyc common sense knowledge base grew ever larger, it became increasingly difficult to shoehorn every fact and rule into the same flat world. Finally, in 1989, as Cyc exceeded 100,000 rules in size, we found it necessary to introduce an explicit context mechanism. That is, we divided the KB up into a lattice of hundreds of contexts, placing each Cyc assertion in whichever context(s) it belonged.

A further illustration of this idea is Dismore's book on partitioned representation [17]. He sees partitioned representations as the functional counterpart of the notion of "mental space" as defined by Gilles Fauconnier [20], and defines them as context–dependent representations of the world:

> Functionally, partitioned representations illustrate the principle of *divide and conquer* in mental representation. Rather than a large homogeneous representation or set of representations with unmanageable possibilities for synthesis in reasoning, partitioned representations make use of many isolated context–dependent representations with locally circumscribed [...] opportunities.

Interestingly enough, KRR seems to share this intuition with other related areas. Two examples will illustrate this "family resemblance". Sperber and Wilson, in their book on relevance [49], express a similar intuition from a psycho-linguistic perspective:

> The set of premises used in interpreting an utterance [...] constitutes what is generally known as the context. A context is a psychological construct, a subset of the hearer's assumptions about the world.

And Boicho Kokinov, in his paper on a dynamic approach to context modeling [36], writes that:

> [c]ontext is the set of *all* entities that influence human (or system's) cognitive behaviour on a particular occasion.

Despite the evidence of a shared intuition, we argue that there are at least two different types of theories of context that have been proposed in KRR:

- the first sees a context as a way of partitioning (and giving a more articulated internal structure to) a global theory of the world;

- the second sees a context as a local theory, namely a (partial, approximate) representation of the world, in a network of relations with other local theories.

According to the first view, which we call *divide-and-conquer*, there is something like a global theory of the world, but it has an internal structure, and this structure is articulated into a collection of contexts. According to the second view, which we call *compose-and-conquer*, there is not such a thing as a global theory of the world, but only many local theories; each local theory represents a viewpoint on the world; there may exist relations between local theories that allow a reasoner to (partially) compose them into a more comprehensive view. In the rest of this section we analyze these two views of context, and present in some details two formalizations of context as a practical illustration of the two types of theory: the Propositional Logic of Context (LoC) [13], which illustrates the divide-and-conquer type (in fact, we will review also Dinsmore's theory of partitioned representations, as it provides an interesting variation of a divide-and-conquer theory), and Local Models Semantics / Multi-context systems (LMS/MCS) [24, 32], which illustrates the compose-and-conquer type. Of course, there are several other (formal) theories of context in KRR, for example structured contexts with fibred semantics [22], and the type theoretic foundation for context [51]. However, we decided to focus on LoC and LMS/MCS for two reasons: first, they provide the clearest illustrations of the two types of theories we described above, and moreover are the theories that were more extensively applied to solve problems in KRR[3].

2.1 A "divide-and-conquer" theory of context: LoC

Partitioning a global theory of the world can mean two different things: (i) that the collection of facts globally available is partitioned into smaller subsets, each of which describes knowledge about some domain, or knowledge that is needed to solve a specific problem; or (ii) that the same set of facts can be given different descriptions, each at a different level of detail, depending on what is implicitly assumed (for example, the fact that at a time T_0 a block A is on a block B can be represented as $on(A, B, T_0)$, but also as $on(A, B)$ in a context in which it is implicitly assumed that the time is T_0). In general, a divide-and-conquer theory has the following form:

---

[3] Again we point the interested reader to [3] for a survey of formal theories of context.

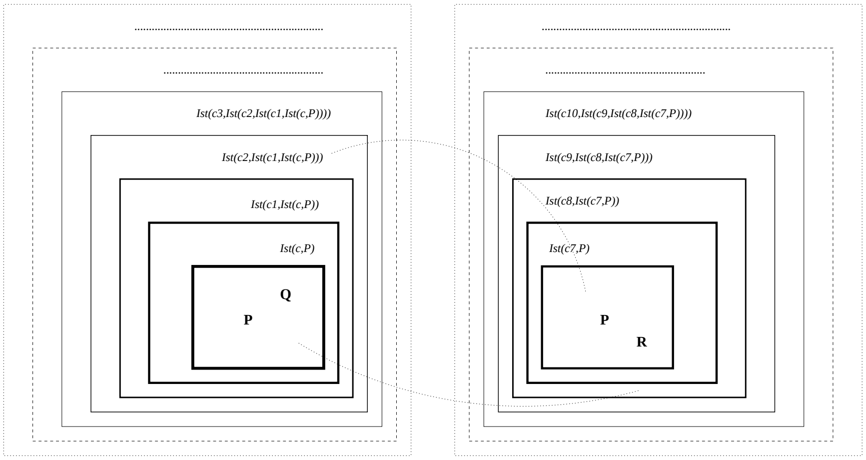

Figure 1: A divide-and-conquer theory of context: LoC

- given a (global) representation language L, the facts that are true in a given context c can be isolated ("localized") and treated as a distinct collection of facts with respect to the facts belonging to different contexts;

- there are hierarchical relations between contexts that allow reasoning to "climb" from a context to a more general context in which the dependence of a fact on a context is explicitly stated (and possibly reasoned about);

- finally, there are lateral (i.e. non hierarchical) relations between facts of different contexts (for example, one would like to be able to represent the relation between the fact $on(A, B, T_0)$ in the context *bw* of the blocks world and the formula $on(A, B)$ in a context *specializes(bw,*T_0*)* that specializes the context of the blocks world to time T_0).

LoC, as intuitively described by McCarthy and then formalized by Buvač and Mason [13], is a possible way of capturing the general intuitions of a divide-and-conquer theory. The building blocks of LoC can be described as follows (see Figure 1):

- first of all, any formula can only be asserted in some context (namely, there are no context independent formulae). The fact that a formula p is asserted in a context c is written as $c : p$ (formally, formulae are asserted in a context sequence $c_1 \dots c_n$, which allows to distinguish, say, the context of car racing in the context of the 50's from the context of car racing in the context of the 90's);

- contexts are reified as first-class objects, which means that they are objects about which we can make assertions in the language of the theory. This means that the representation language L is enriched with a set of context names $c_1, c_2, \dots$;

- the most important statements about a context are made through the formula $ist(c, p)$. Intuitively, it means that the formula $p \in L$ is true in the context c. ist is treated as a modality in [13, 14];

- the main hierarchical relation between facts belonging to different contexts is that between the fact p stated in a context c, and the formula $ist(c, p)$ stated in some outer context c'. The relation is the following:
  - if $ist(c, p)$ can be proved in c', then we can always "enter" the context c and assert $c'c : p$. If c is the context of car racing, this allows us to "enter" c and restrict reasoning only to the facts that are true in it;
  - if p can be proved in c, then we can always "leave" (or *transcend*) the context c and make the explicit assertion $ist(c, p)$ in the outer context c' (i.e. $c' : ist(c, p)$).

  These relations define the nested structure of Figure 1. Indeed, leaving (or transcending) a context is tantamount as moving from a box to the box immediately outside; whereas entering a context is the move from outside to inside;

- other (lateral) relations between facts belonging to different contexts are stated through the so-called *lifting axioms* (see [35] for this notion). The general form of a lifting axiom is

  $$ist(c, p) \iff ist(c', p')$$

  In Figure 1, lifting axioms are represented as dotted lines connecting boxes. One can use a lifting axioms to say that, for example, in the context of the Sherlock Holmes stories, it is true that Holmes lived near Victoria Station if and only if, in the context of the actual London, Victoria Station is near Baker Street;

$$\begin{array}{ll}
\text{(PL)} & \vdash_{\overline{c}} \phi \qquad \text{If } \phi \text{ is an instance of a classical tautology} \\
\text{(K)} & \vdash_{\overline{c}} ist(c, p \supset q) \supset ist(c,p) \supset ist(c,q) \\
(\Delta) & \vdash_{\overline{c}} ist(c_1, ist(c_2,p) \vee q) \supset ist(c_1, ist(c_2,p)) \vee ist(c_1,p) \\
\text{(MP)} & \frac{\vdash_{\overline{c}} p \qquad \vdash_{\overline{c}} p \supset q}{\vdash_{\overline{c}} p} \\
\text{(CS)} & \frac{\vdash_{\overline{c}c} p}{\vdash_{\overline{c}} ist(c,p)}
\end{array}$$

Figure 2: Axioms and inference rules for PLC

- finally, there is no such a thing as an outermost context. This reflects McCarthy's intuition that we can never resolve all contextual dependencies of a fact. More technically, this means that we can always transcend a context c and move to a more general context in which facts about c (including making explicit some of its implicit assumptions) can be asserted. In Figure 1, this is represented by the external dotted boxes.

These intuitions are formalized in Buvač and Mason's propositional logic of context [13] (followed by Buvač's first order formalization [14]). For the sake of simplicity, we only describe the propositional part, called LoC.

In LoC, we start with a propositional language L (which includes a collection of context names and the modality *ist*). Roughly speaking, a model $\mathfrak{M}$ for LoC associates a set of partial truth assignments to each sequence of contexts (possibly of length one) and satisfiability is defined with respect to a sequence of contexts. This means that any formula is satisfied with respect to a context. The idea is that partial truth assignments capture the fact that in different contexts there are different sets of meaningful formulae. Indeed, starting from a unique language L, a model $\mathfrak{M}$ defines a function, called vocabulary, that associates to each context c a subset of L which is the set of meaningful formulae in c. Obviously, satisfiability and validity of formulae are defined only for these models that provide enough vocabulary, namely the vocabulary which is necessary to evaluate a formula in a context.

Buvač and Mason propose an Hilbert style axiomatization of validity for the logic of context, which is presented in Figure 2. (PL) says that all propositional tautologies are valid in every context $\overline{c}$. Axiom (K) imposes that predicate *ist* satisfies properties analogous to those of the modality $\Box$ in a modal system K [15]; while axiom (Δ) forces the truth of *ist* formulae (i.e. formulae of the form $ist(c,p)$) to be independent of the assignments of the contexts in which they occur[4]. (MP) is the usual rule of Modus Ponens. The rule (CS) is very important. It formalizes the hierarchical relationships between contexts in LoC. This relationship, which is part of the logic itself, is the mechanism that allows transcending a context. As we already said, this corresponds to moving one step outward in Figure 1.

Very briefly, we present also Dinsmore's theory of partitioned representations (PR) [17], a logic strictly related to Fauconnier's work on *mental spaces* [20]. As we will show, PR can be viewed as another instance of a divide-and-conquer theory. Despite some terminological differences, LoC and PR share most of their structure. However, there is an important difference: while in LoC the process of transcending a context is open-ended (there is no outermost context), in PR we eventually reach a special space (called BASE), which cannot be further transcended, and therefore is a sort of outermost context.

In PR, a sentence is always asserted in a space. A space represents some logically coherent situation or potential reality, where various propositions are treated as true, objects are assumed to exist, and relations between objects are supposed to hold (e.g., belief spaces; hope and wish spaces; fictional, dream, and pretense spaces; spaces representing specific places, times and situations; spaces representing the scope of certain existential assumptions; spaces expressing generalizations; spaces representing the implications of certain propositional assumptions, either conditional or counterfactual). In Figure 3, spaces are represented as rectangles.

Each space has exactly one primary context. A *primary context* is defined as a function that maps the truth of a proposition in one space onto the satisfaction of a (more complex) proposition into another space. Suppose, for example, that the sentence "Mr. Bush is the President of the US" is asserted in a space named S_1 and that "Warren believes that [S_1]" is the primary context of S_1 (here we are using a notation which is slightly different from the original). Then this allows us to map the truth of "Mr. Bush is the President of the US" onto the truth of the (more complex) sentence "Warren believes that [Mr. Bush is the President of the US]", which in turn is asserted in some other space. Of course, the semantics of "Mr. Bush is the President of the US"would be very different in a space S_2 whose primary context was something like "In the Sherlock Holmes stories [S_2]". This process is called *context climbing*, and corresponds very closely to the idea of transcending (leaving) a context in LoC.

---

[4] For a short discussion of the axiom (Δ), see [9].

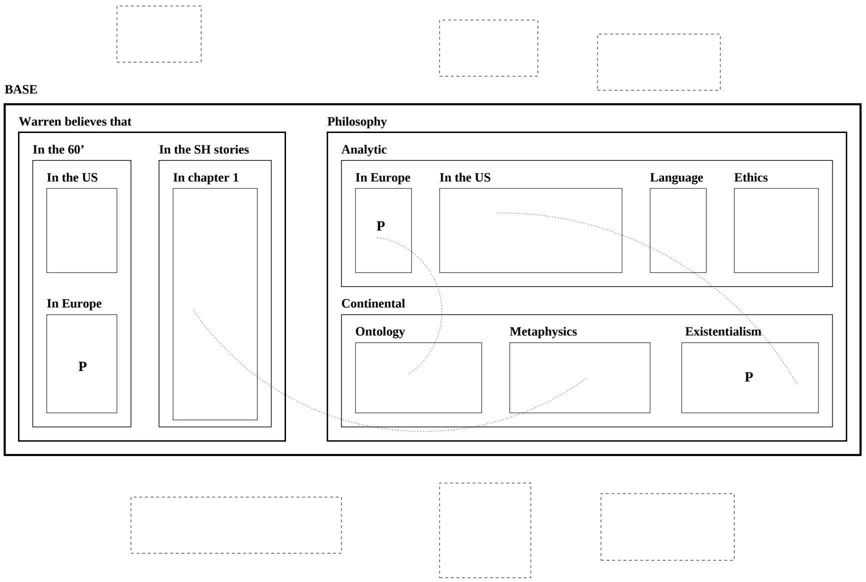

Figure 3: Dinsmore's partitioned representations

Via context climbing, PR allows us to reach a special space, called BASE, which functions as an outermost space. BASE is the only space that does not have a primary context. Even though Dinsmore would not accept this interpretation, formally BASE represents a "de-contextualized" representation of the world. Indeed, it is only in BASE that assertions are given an interpretation. As a consequence, if BASE is not reachable (via context climbing) from a space S, then the assertions of that space are left without a truth value.

Dinsmore introduces also a notion of *secondary context*, which allows for lateral mappings. Intuitively, a mapping is a consequence of the semantics of the primary contexts involved. In other words, a secondary context opens a channel of communication between two spaces. For instance, if S_1 models Warren's beliefs about Bush and S_2 Warren's beliefs in general (no matter about what), then we can imagine that the facts asserted in S_1 will be inherited by S_2 (inheritance is just one – perhaps the simplest – use of secondary context in PR).

It is easy to see that there are many similarities between LoC and PR (even though LoC is formally richer than PR). Moreover, they share the general structure of a compose-and-conquer theory of context: a way of localizing collections of facts; a hierarchical relation between contexts built-in in the logic; a way of defining non hierarchical (lateral) mappings between facts belonging to different contexts. We presented both because they offer different solutions to the problem of de-contextualization: open-ended in LoC, bounded in PR.

2.2 "Compose-and-conquer" theories of context

Compose-and-conquer theories start from the assumptions that local, domain specific, goal directed theories of the world are the building blocks of what an agent knows, and that the totality of his/her knowledge is given by composing such local theories through a collection of rules that connect them into a more comprehensive (but still partial) representation of the world.

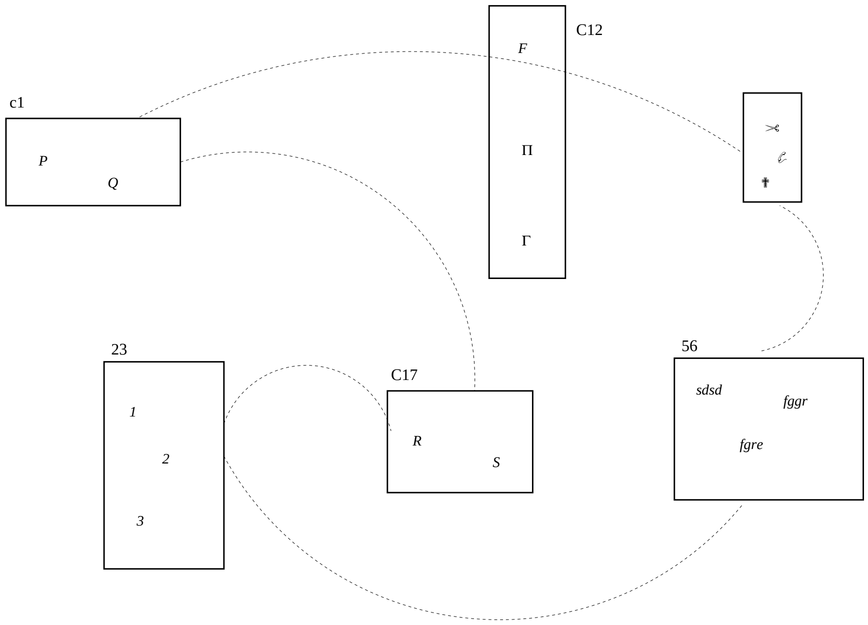

Figure 4: A compose-and-conquer theory of context: LMS.

The general structure of a compose-and-conquer theory can be described as in Figure 4. First of all, each box is a local theory. A local theory is not a partition of a bigger (global) theory, but a full-blown theory which represents knowledge about some portion of the world (partiality), at some level of detail (approximation), from a given perspective. Examples are:

domain theories (e.g. about air traveling, cars, sports, cooking), snapshots of a dynamic situation (e.g. the state of a chess game, the current situation during the execution of a plan), representation of a physically limited portion of the world (e.g. the location of physical objects in a room, the location of restaurants in new York); beliefs ascribed to another agent or group of agents (e.g. the beliefs that John ascribes to Mary, John's beliefs about the beliefs that Mary ascribes to him).

Second, there are no a priori relations between contexts. This is a major difference with divide-and-conquer theories. As we said, hierarchical relations are "hardwired" in systems like LoC and PR, and the reason is that contexts are chunks of a bigger (global) theory of the world; in a sense, the global model is a sort of road map that says how contexts are related to each other. In compose-and-conquer theories, there is no predefined road map, and contexts are autonomous theories, though partial and approximate. Of course, this does not mean that there are no relations between contexts, but that these relations are established on a peer-to-peer basis, as a collection of constraints on what can (or cannot) be true in a theory given that there is some relation with what holds in another context. A special case of a peer-to-peer relation is a hierarchical relation (e.g. transcendence), but it is interpreted as a constraint between two autonomous local theories. For example, there may be a constraint between the truth of a fact p in a context c and the truth of a fact $ist(c, p)$ in a context c', but this is interpreted as the fact that any local model of c that satisfies p is incompatible with any local model of c' that does not satisfies $ist(c, p)$, and vice versa. Under this respect, hierarchical relations in compose-and-conquer theories are assimilated with any other relation, and does not require to hardwire in the logic any order over contexts.

The neatest formalization of a compose-and-conquer theory of context is Ghidini and Giunchiglia's *Local Models Semantics* (LMS) [24], together with its proof-theoretical counterpart, namely Giunchiglia and Serafini's *Multi-Context Systems* (MCS) [32]. LMS is based on two very general principles, that for our purposes we restate as follows:

- *principle of locality*: reasoning always happen in a local theory (a context);
- *principle of compatibility*: there may be compatibility constraints between the reasoning processes that happen in different contexts.

In [24], these two principles are given both a model-theoretic (LMS) and a proof-theoretic (MCS) formalization (even though the first proof-theoretic version can be found in [32]). In LMS, one starts with a family of languages $L_1, \dots L_n, \dots$, where each L_i is the representation language of a context c_i. Each language L_i has its set of models M_i. Every subset M_{T_i} of M_i satisfies a set of formulae, each corresponding to a different choice of the theory (set of true facts) T_i associated to c_i[5]. Once the theory T_i associated with c_i is fixed, a model belonging to M_{T_i} is called a *local model* of c_i. Going back to Figure 4 this means that each context (box) c_i depicted in that figure is formalized by the set M_{T_i} of models of L_i which satisfy the axioms in c_i. Relations between two contexts (dotted lines between boxes in Figure 4) are represented by compatibility constraints, which state that the truth of a formula Φ in c_1 is related to the truth of the formula Ψ in c_2[6]. This is achieved by imposing that sets of local models $\mathbf{c}_1$ and $\mathbf{c}_2$ of the two contexts c_1 and c_2 are such that

$$\text{if } \mathbf{c}_1 \text{ satisfies } \Phi, \text{ then } \mathbf{c}_2 \text{ satisfies } \Psi \tag{1}$$

where the notion of satisfiability of a formula in a set of local models is the same as the notion of satisfiability of a formula in the theory associated to c_i. Pairs $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ satisfying Equation (1) are said to belong in a compatibility relation and define a model for the pair of contexts $\{c_1, c_2\}$.

The proof-theoretic counterpart of LMS is called *MultiContext Systems* (MCS). A MCS is a pair $MC = \langle \{c_i\}, BR \rangle$, where $\{c_i\}$ is a set of axiomatic formal theories (namely triples of the form $c_i = \langle L_i, \Omega_i, \Delta_i \rangle$), and BR is a set of *bridge rules*. Bridge rules are rules whose premisses and conclusion belong to different contexts. For instance, the bridge rule corresponding to the compatibility constraint described above would be the following:

$$\frac{c_1 : \Phi}{c_2 : \Psi}$$

where $c_1 : \Phi$ is the premiss of the rule and $c_2 : \Psi$ is the conclusion. Obviously, bridge rules are conceptually different from *local rules* (i.e., rules in Δ_i). The latter can be applied only to formulae of L_i, whereas the former have the premises and the conclusion that belongs to different contexts. Intuitively, bridge rules allow for the MCS version of compatible derivations.

---

[5] By abuse of notation, we will use the symbol c_i (possibly with different subscripts), to mean either the theory associated with context c_i or a context embedded in a structure of relationships with other contexts

[6] The case of multiple contexts and multiple compatibility constraints is a straightforward generalization

A deduction in a MCS is a tree of local deductions, obtained by applying only rules in Δ_i, concatenated with one or more applications of bridge rules (see [32] for a technical treatment).

It should be clear that, using this machinery of compatibility in LMS (or bridge rules in MCS), a wide range of relations between contexts can be formalized. Some examples are:

- suppose a context c represents John's beliefs at time t, and that it contains the sentence "I'm hungry". If c_1 represents John's beliefs at time $t+1$ (e.g. the day after), then there exists a relationship between the two contexts such that the sentence "Yesterday I was hungry" must be true in c';
- suppose c_2 represents Mary's beliefs at time t, and that John tells Mary "I'm hungry". Then the relationship between c and c_2 is such that the sentence "He is hungry" (or "John is hungry", if Mary knows that the speaker's name is John) must be true in c_2;
- suppose c_3 represents the positions of the object in a room (including John and Mary) at time t and that the sentence "John is near Mary" is true in it. If c_4 represents the location of the same objects at different times, then the sentence "At time t, John was near Mary" must be true in it.

All these relations have the following form: if a sentence p is true in a context c, and c is in a given relation R with c', then the sentence p' must be true in c'. Of course, a special case is when there is no relation between the two contexts, and therefore there is no constraint on what is true in the two contexts.

It is easy to see why LMS/MCS is a "compose-and-conquer" logical framework. Model-theoretically, the idea is that contexts can be composed through compatibility relations, which allow to exclude all local models that are not compatible with the known relationships between two local theories. Proof-theoretically, the idea is that bridge rules allow us to derive in a contexts more facts than would be derivable if the context was taken in isolation (the reason is that these further facts are derivable precisely because of the relations that contexts have with each others). However, both model- and proof-theoretically there is no assumption that there is a global model of the world.

To sum up, there are very significant conceptual and formal differences between divide-and-conquer and compose-and-conquer theories. Indeed, in a compose-and-conquer theory:

- there is not such a thing as a general representation language L. The representation language is context-dependent, as it reflects the "ontology" implicitly assumed in a local representation;
- denotation and truth are by definition contextual, as they are defined with respect to the language and models of each context;
- reasoning is local by definition, as it always happen in a context. Since there is no single logical space that contains (not even potentially) all what an agent knows, reasoning can only happen in the small logical spaces that correspond to a context. This means that different contexts may even correspond to different reasoning rules;
- even relationships between different contexts are not necessarily stated and used at a meta-level (as it is the case, for instance, in LoC). They can be viewed as constraints on what can be locally derived in a context (see below for technical details about this).

Despite the differences above, [9] proves that LMS/MCS can be used to "simulate" a "divide-and-conquer" approach. In particular, it is shown that LMS/MCS can subsume LoC. Intuitively, a "divide-and-conquer" approach uses context as a mechanism for partitioning a global representation into logical spaces that are smaller and simpler than the global knowledge space of a program (or, more in general, of an agent). This can be easily done in a logic where each context is described by its own local language and semantics. First of all, entering and leaving a context can be modeled as a specific compatibility relation imposing that p holds in the context $\overline{c}$ if (and only if) $ist(c,p)$ holds in $\overline{c}$. Analogously, lifting axioms can be modeled as compatibility constraints between local theories. This intuitively justifies the claim that the principles of locality and compatibility are the most general principles of contextual reasoning, and that a logic based on these principles is general enough to provide a suitable basis for context-based KRR applications.

3 Uses of context in KRR

In the following we focus on the usage of context in specific areas of KRR. Our aim is to illustrate the idea that some problems are more naturally addressed in a divide-and-conquer perspective, others in a compose-and-conquer perspective. The examples we consider are: the problem of generality, the formalization of propositional attitudes, knowledge and data integration.

3.1 The problem of generality

The so-called problem of generality (identified by McCarthy in his paper on *Generality in Artificial Intelligence* [43]) and its dual problem, the qualification problem[7] are a typical point of contact between context and KRR. In [31], the authors elaborate a specific version of the problem of generality, that is the problem of dealing with expected and unexpected obstacles in the Glasgow-London-Moscow (GLM) example, first proposed by McCarthy in the unpublished note *Overcoming an Unexpected Obstacle* (1991):

> You are planning a trip from Glasgow to Moscow via London. You would like to build the plan maybe without having to think of all the details, i.e. by working in a fairly approximated theory. For instance you are willing to consider the fact that you must have a ticket in order to get on a plane but not the fact that the flight could be cancelled. However you want to be able to revise your plan if an expected obstacle arises (e.g. you do not have the ticket because you have lost it) and more particularly if an unexpected obstacle arises (e.g. the flight is cancelled).

Here the problem is the tradeoff between needed generality and an excess of generality. The requirement of building a plan "without having to think of all the details" goes in the direction of finding the appropriate context containing only the information which is needed to solve the specific problem of planning a trip from Glasgow to Moscow via London. However, the fact that "you want to be able to revise your plan if an expected obstacle arises [...] and more particularly if an unexpected obstacle arises" requires the ability to change to a less general context, containing also the information

---

[7] The qualification problem was originally identified by McCarthy in [42]. However, its duality with the problem of generality was highlighted in [31].

which is needed to undertake the unexpected obstacle. Depending on the context, the "same" problem of planning a trip from Glasgow to Moscow via London can be given several different representations with a different degree of generality. If, on one side, a more general representation can be applied to a larger class of circumstances, on the other side too much generality is a problem from the standpoint of implementing a reasoning system. In many contexts, some information can be left implicit that in other contexts must be included. Another prototypical example of the problem of generality is McCarthy's 'above–theory' example [44]:

> Consider a context *above-theory*, which expresses a static theory of the blocks world predicates *on* and *above*. In reasoning about the predicates themselves it is convenient not to make them depend on situations or on a time parameter. However, we need to *lift* the results of *above-theory* to outer contexts that do involve situations or times.

As a consequence, the *above-theory* context contains very simple axioms on the blocks world of the form:

$$on(x, z) \supset above(x, z) \tag{2}$$

$$above(x, y) \wedge above(y, z) \supset above(x, z) \tag{3}$$

These axioms say that an object x is above an object y if either x is on y or x is above an object y which in turn is above z. Most times, these two axioms are sufficient for reasoning about the property of being above in the blocks world. However, there are cases where these two axioms are not general enough; for instance, (3) is true only if $above(x, y)$ and $above(y, z)$ are true at the same time. If we need to reason about the property of being above in a context c that do involve situations or times, McCarthy suggests the use of lifting rules to export axioms of *above-theory* to c and to add a parameter for the time to the predicates *on* and *above*. Axioms (2) and (3) are then exported in c as:

$$on(x, z, t) \supset above(x, z, t) \tag{4}$$

$$above(x, y, t) \wedge above(y, z, t) \supset above(x, z, t) \tag{5}$$

Summarizing, McCarthy suggests to solve the problem of generality with a formalization of context which allow us to use the "right" axioms in the "right" context, e.g., the less general axioms (2)–(3) if the context allows us to disregard the time, and the more general axioms (4)–(5) in a context where time is relevant.

The formalization of context proposed by McCarthy and his group, including Buvač and Mason's logic of context, is meant to deal particularly with the problem of generality. The reader interested in this approach may refer to [44, 35] for the general intuitions and many motivating examples, and to [13, 14] for the logical framework. On the contrary, the problem of generality has never been tackled by Giunchiglia and his group. This is not very surprising. From the description of the problem of generality given above, it is easy to see how the notion of context used to solve this problem is naturally addressed in a divide-and-conquer perspective. There is a global model of the world. However, too much knowledge is a problem from the standpoint of implementing a reasoning (planning, ...,) system. Contexts provide the solution as they are a device used to focus on (smaller, simpler) portions of such a model. Transcendence, and lifting are the mechanisms used to relate contexts at different levels of generality.

3.2 Modeling propositional attitudes

Contexts have been extensively used to formalize mental states. It is not very surprising to discover that this problem has been mainly addressed in a compose-and-conquer perspective. In fact, contexts have been used in this area because they provide a tool for the representation of a (partial, approximate) theory from an individual's perspective. As a consequence most of the work we describe here is formalized using Local Models Semantics and MultiContext systems.

3.2.1 VIEWGEN

Ballim, Wilks and colleagues proposed VIEWGENas a framework for modeling agents which have a representation of beliefs, intentions and goals of other agents involved in a dialogue [5, 37]. VIEWGEN assumes that each agent taking part in a dialogue has a belief environment which includes attitudes about what other agents believe, want, and intend. Such attitudes are represented in a nested structure. Each nesting contains propositions which may be grouped by a particular topic or stereotype. The particular topic is given on the top left corner of the environment while the holder of a belief is given at the bottom of the environment. Moreover, the attitude type (belief, intention, or goal) is given on the far right bottom of the box. Though different attitude types are separated by environments, they can be nested so that agents can have beliefs, goals, and intentions about these attitudes. Figure 5 shows how meta-attitudes are used in ViewGen to represent the fact that the System believes that John intends to buy a car, but wants to convince him otherwise by getting him to believe correctly that the car is a wreck [37].

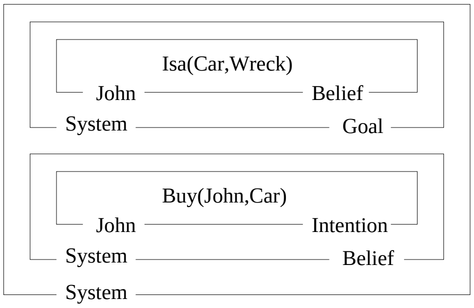

Figure 5: Meta-attitudes in ViewGen.

As we can see from Figure 5 ViewGen avoids using a concept of shared or mutual beliefs. Rather, ViewGen attributes beliefs, goals and intentions to other agents as required. This process is termed ascription. There are two methods of ascription: default ascription and stereotypical ascription. Default ascription applies to common attitudes which ViewGen assumes that any agent will hold and also ascribe to any other agent unless there is contrary evidence. Such a rule results in beliefs being pushed from outer belief environments to inner belief environments. For example, Figure 6 illustrates ViewGen ascribing beliefs to John about New Mexico. Stereotypical ascription is usually applied to "uncommon" attitudes which ViewGen assumes hold only for a particular class of agents.

In the ViewGen framework locality and compatibility play an important role. Environments are thought of as a means for modeling local belief, intentions and goals. The idea of locality of beliefs (intentions, goals) is so strong that ViewGen avoids using a concept of shared or mutual beliefs. Every environment describes a set of attitudes from a certain point of view. Attitudes cannot be shared by different agents, all what an agent can do – before and during a dialogue with another agent – is to ascribe to the other agent its beliefs unless there is contrary evidence. We can therefore consider ViewGen as an example of the compose-and-conquer approach. It is clear that the mechanism of ascription imposes particular relations be-

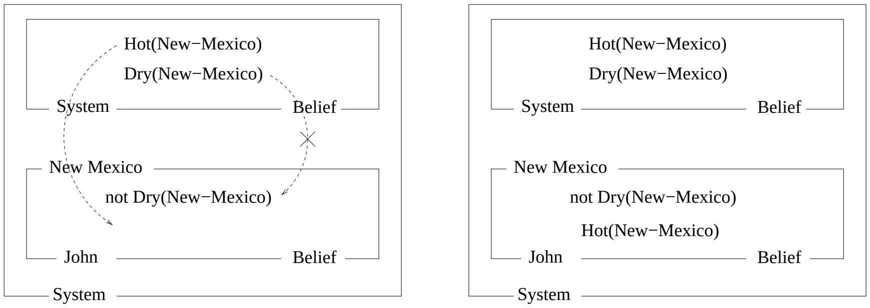

Figure 6: ViewGen: before ascription(left) and after ascription (right).

tween environments, and is possible only because the environments involved are supposed to be compatible. In the example of Figure 6, after the belief ascription process, the environment containing John's beliefs is compatible with the system's beliefs only if for every fact contained in the system's beliefs either this fact or its contrary is present among John's beliefs.

3.2.2 Representing belief

The notion of context has been applied to formalize different aspects of intentional context, and in particular belief context. The approach we describe here was first introduced in [32], and then used in several papers to formalize different aspects of reasoning about belief.[8] In the following we call this approach to the representation of belief *Hierarchical Belief* (HB).

HB focus on a scenario with an agent, ϵ, who is acting in a world, who has beliefs about this world and is also observing and representing beliefs about a set $Ag = \{1, \ldots, n\}$ of agents (possibly including itself), and it is able to reason about them. Any element i in Ag is called an *agent index*. In order to formalize this scenario, HB introduces the notion of *belief context* (also called *view*) [7].

> A view [belief context] is a representation of a collection of beliefs

---

[8] [16] uses the notion of belief context introduced in this section to solve a well-known puzzle involving reasoning about belief and ignorance, namely the Three-Wise-Men problem. In [33, 28] the representation of an ideal and real reasoner using belief context is thoroughly discussed. In [6] belief contexts are used to solve the problem of the opaque and transparent reading of belief reports. In [21, 23] the representation of resource-bounded deliberative agents is discussed.

> that a reasoner [...] ascribes to an agent (including itself) under a given perspective.

Examples of perspectives are: the beliefs that ϵ ascribes to itself; the beliefs that ϵ ascribes to another agent i; the beliefs that ϵ ascribes to an agent i about another agent j, and so on. As a convention, HB uses the Greek letter ϵ to indicate the belief context containing the beliefs that the reasoner ascribes to itself, and sequences of agent indexes (that is, sequences of elements of Ag) to label any other belief context. For instance, ϵi is the belief context containing the beliefs that the program ascribes to agent i from its perspective, and ϵij is the belief context containing the beliefs that ϵ ascribes to i from agent j's perspective. Iterating the nesting, the belief context ϵijk formalizes the beliefs of agent i about j's beliefs about k's beliefs. Since in the scenario presented above no confusion can arise, the prefix ϵ is often omitted. Figure 7 summarizes the idea underlying the HB approach, in a scenario in which the reasoner ϵ ascribes a collection of beliefs to three agents 1, 2, and 3.

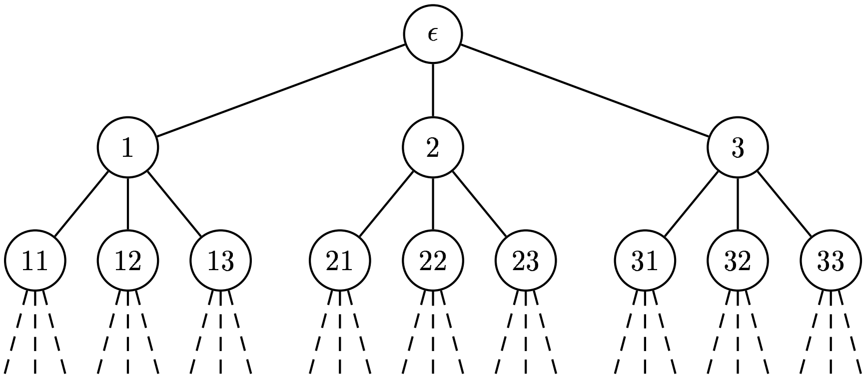

Figure 7: The context structure of beliefs in a scenario with three agents.

HB formalization is based on the idea that a distinct language is associated with each belief context, and the interpretation of such a language is local to the belief context it is associated with. The idea is that a formula A in the external observer context, also written $\langle A, \epsilon\rangle$ to stress the context dependence, expresses the fact that ϵ believes A. The same formula in context ijk, i.e. $\langle A, ijk\rangle$, expresses the (more complex) fact that i believes that j believes that k believes that A (from the point of view of ϵ). HB are formalized using Local Models Semantics and Multicontext systems. Each belief context is represented as a formal system $c_i = \langle L_i, \Omega_i, \Delta_i\rangle$ (see

Section 2.2). To express statements about the world, L_i contains a set P_i of propositional constants. To express belief, L_i contains well formed formulae of the form $B_k(\text{“}A\text{”})$, meaning that agent k believes the proposition expressed by A.

The other fundamental idea of the HB approach is that although distinct, the contents of different belief contexts are related. Relations between contexts, which in principle can be very different, express how the beliefs of an agent, say agent ϵ, and the beliefs that ϵ ascribes to (itself or to) another agent, say agent i, are connected. A taxonomy of possible relations involving belief about belief is introduced in [33] and then refined in [28]. In this work the authors show that, depending on the relations among different contexts, the agent ϵ has different reasoning capabilities. An example of an obvious, and well studied, relation between belief contexts is the following: if a sentence of the form A is in ϵi, then a sentence of the form "i believes that A" is in ϵ. In this case we say that ϵ is a *correct* observer (w.r.t. the sentence "i believes that A"). Another situation is when a sentence of the form A is in ϵi, only if a sentence of the form "i believes that A" is in ϵ. In this case we say that ϵ is a *complete* observer (w.r.t. the sentence "i believes that A"). Formally, these relations between belief contexts are represented by the following *bridge rules* [32]:

$$\frac{\langle B_i(\text{“}A\text{”}),\ \epsilon\rangle}{\langle A,\ \epsilon i\rangle}\ \mathcal{R}_{dn} \qquad\qquad \frac{\langle A,\ \epsilon i\rangle}{\langle B_i(\text{“}A\text{”}),\ \epsilon\rangle}\ \mathcal{R}_{up}$$

As we can easily see from the discussion above, HB are a clear example of compose-and-conquer use of context. Belief contexts are local theories, each of them expressing a partial and approximate representation of the world under a given perspective. Different relations, as the one shown above, may exist among belief contexts that allow the reasoner to (partially) compose them into a more comprehensive structure.

3.2.3 Recognizing mental states from communication

Representation of beliefs is naturally described by a compose-and-conquer approach, and the modeling of communication acts is a good example of this paradigm. The consequences of the speech acts are completely local and limited to the beliefs of an agent, but composition between different contexts is required in order to model the changes in the mental representation during communication. Moreover, it is not guaranteed that a unique, coherent theory exists, due to phenomena like deceit or misunderstanding.

The formal machinery of MultiContext systems was exploited by Dragoni et al. [18] in order to give an account of the belief revisions of communicating agents. The goal is modeling the consequences of an utterance on the mental states of the hearer. In particular they focused their attention on agents with Beliefs and Intentions expressed in a single language L_i that is to say that for each agent j, $B_j\phi$ means that the agent j believes ϕ and $I_j\phi$ means that agent j intends to bring about ϕ.

The authors adopt a plan-based vision of speech acts and they deal with the speech acts INFORM and REQUEST as presented in artificial agents communication languages and described in terms of preconditions and main effects in the set of believes. They assume the existence of a causal relationship between an agent's mental state and its possible uttering a sentence. As a consequence, abduction is used for updating the hearer's mental state.

The revision goes as follows. Let us suppose the agent s performs for instance the speech act INFORM(s,h,ϕ) (where h is another agent and ϕ is a formula) whose precondition is $\phi \in B_s^*$ and whose main effect is $\phi \in B_h B_s^*$ where B_s^* is the deductive closure of B_s in the MultiContext system. h may update its image of the speaker's mental states by imposing that the preconditions of INFORM(s,h,ϕ) hold that to say: $\phi \in B_h B_s^*$ (that, in this particular case, coincides with its main effect). Abduction is then needed in order to revise the current explanation from X to X' where X and X' are the sets of formulas that can be modified by receipt of the speech act.

The causal relationships the authors assume for the speech act INFORM(s,h,ϕ) are:

I1 s has the intention of bringing h in a mental state where ψ (in the general case $\psi \neq \phi$) is either believed or intended by h.

I2 s does not believe that h is already in that mental state.

I3 s believes that if it performs the INFORM act, then h will be in a mental state in which it believes or intends ψ.

The next phase is devoted to intention recognition [18].

> By intention recognition we mean the hearer's ability to recognize the intention that induced the speaker to perform the speech act. Condition I1 states that a motivation for s to perform an INFORM(s,h,ϕ) is its intention of changing h mental states so that h believes or intends some new formula. To discover this intention, h checks the differences between its mental state before

> and after s executes INFORM(s,h,ϕ) (X and X' respectively), and then it revises X' to include the fact that s has the intention of causing this differences.

Finally, the hearer updates its image of speaker's mental state in order to have that the speaker believes that its intentions have been satisfied. The authors presents also the more complex case of of the speech act REQUEST. The work is deeply related to work done in cognitive pragmatics as described in [1].

3.2.4 Agent's theory and applications

Contexts have been used for specifying architectures for negotiating agents in [47, 48]. In a nutshell, the idea is to use the notion of context for representing the different components of an agent architecture, and for specifying the interaction between the different components as appropriate rules between contexts. Here the focus is on the use of contexts for designing modular architectures, easy to maintain and to modify [47].

> We use different contexts to represent different components of an agent architecture, and specify the interactions between the components by means of [...] rules between context. [...] This approach enforces a modular structure with well-defined interfaces, and thus accords well with good software practice.
>
> This approach to system design affords all the traditional advantages of modularization in software engineering and enables complex artifacts to be designed out of simpler components.

More in detail, an agent architecture consists of four components

- *Units:* Structural entities representing the main components of the architecture.
- *Logics:* Declarative languages, each with a set of axioms and a number of rules of inference. Each unit has a single logic associated with it.
- *Theories:* Sets of formulae written in the logic associated with a unit.
- *Bridge rules:* Rules of inference which relate formulae in different units.

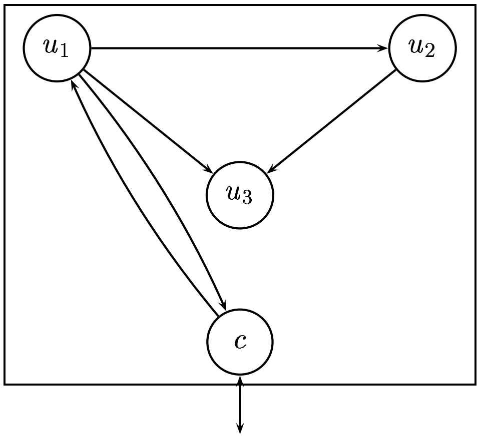

Figure 8: A context-based agent architecture.

Units define the set of modules (contexts) of an agent architecture. *Logics* assign to each module a logic used to formally describe the content of the unit. *Theories* assign to each module a set of facts true in that module. Finally, *Bridge rules* contains the set of rules that specify the interactions between modules. Notationally, bridge rules are written as follows

$$\frac{\langle \phi,\, u_1 \rangle \quad \langle \psi,\, u_2 \rangle}{\langle \theta,\, u_3 \rangle}$$

This particular bridge rule means that the formula θ may be inferred in unit u_3 because of the fact that ϕ and ψ are derivable in the units tagged with u_1 and u_2, respectively. Figure 8 shows an example of architecture, where the units are u_1, u_2, u_3, and c. u_1 is formalized as a propositional logic, while u_2, u_3, and c are formally described using a first order logic. Bridge rules are depicted as arcs connecting the units. As a more practical example, units u_1, u_2, u_3 can contain the Belief, Desires, and Intentions of the agent, respectively, while c is the communication unit which is responsible for enacting the agent's communication needs. By imposing the appropriate bridge rules between the Belief, Desires, and Intentions units, [47] shows how to model different agent's behaviors. For a more complete presentation of how BDI agent architectures can be modeled within this approach, the interested reader may refer to [47]. The formal framework underlying the agent architecture defined in [47] is that of MultiContext systems.

Is is disputable whether the notion of context presented in [47, 48] follows a divide-and conquer or a compose-and-conquer approach. The authors do not state explicitly either the fact that a context is an expression of a viewpoint or the existence of a unique model as a primitive concept.

3.3 Knowledge and data integration

Knowledge and data integration is an area where both the divide-and conquer and the compose-and-conquer approaches to contexts have been widely applied. The difference between the two different approaches is very relevant. The divide-and conquer use of context in knowledge and data integration is based on the fact that a unique global schema can always be reconstructed. As a consequence, the semantic heterogeneity between different information sources (represented as different contexts) can always be resolved. On the contrary, the compose-and-conquer use of context aims at providing formal systems for a federation of heterogeneous data or knowledge bases, possibly developed independently. Each knowledge base can be seen as a set of views of an ideal data base which is often impossible or very complex to reconstruct completely. In this respect the main goal of the compose-and-conquer approach is to (partially) relate semantically heterogeneous information sources and not to integrate such information in a unique and homogeneous schema.

In the following we present several usages of context in data integration. The main problems we consider are the integration of different information sources and the partition of very large knowledge bases.

3.3.1 Integration of different information sources

Context based frameworks have been used to provide formal models for the integration of information (or knowledge) coming from different sources, often developed independently. Their usage is based on the intuition that different information sources integrated in a unique system (or federation) can be thought as partial views (thought as contexts) on a common world. Indeed, very often the different information sources have very little in common. They are obviously distributed, that is, each information source is on a different system and contains a specific piece of knowledge. They are redundant, meaning that the same piece of knowledge may be represented, possibly from different perspectives, in more than one information source. They are partial, that is the information contained in an information source may be incomplete. Finally, they are autonomous, that is, each information source has a certain degree of autonomy regarding the design, the execution, and the communication with the other databases. As a consequence information sources may adopt different conceptual schemata (including domain, relations, naming conventions, ...), and certain operations are performed locally by the information source, without interactions with the others. It is therefore easy to consider the knowledge (data) contained in each information source as context dependent. Apart from this initial common assumption, different problems are addressed, and different solutions provided, by the compose-and-conquer and divide-and-conquer approaches.

The compose-and-conquer approach Ghidini and Serafini [25, 26] use Local Models Semantics as a formal framework for information integration. Let us discuss this work by mean of an example presented in [26]

> Let m be a mediator of an electronic market place for fruits composed of three fruits sellers: 1, 2, and 3. m collects information about fruit prices from 1, 2, and 3 and integrates it in a unique homogeneous database. Customers that need information about fruit prices, instead of connecting each seller, can submit a single query to the mediator. Figure 9 gives a graphical representation of the structure of this example. Circles represent databases and arrows represent information flow between databases.

[26] describes a formalization of the exchange of information in this example by means of the four contexts and information flows depicted in Figure 9. Circles represent contexts associated to the different databases and arrows represent information flow between contexts (databases).

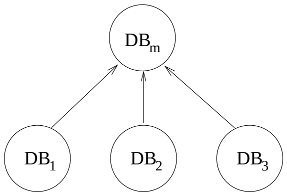

Figure 9: Contexts in the mediator example.

The representations of the different contexts in Figure 9 is done by using Local Models Semantics. It associates a different (set of) first order model(s) to each database. This enables the authors to formalize the fact that each database is associated to a specific domain, e.g. the sellers might provide different subsets of fruits, and therefore the domains of their databases are different. Moreover it enables to represent the fact that the domain of fruits can be represented at different level of details by different sellers. For instance, database 1 may contain prices for red apples and yellow apples, while database 2 and 3 abstract away the dependence on the color and do not make this distinction. Finally, [26] points out that associating different models to different databases enables to capture the fact that prices for different sellers might be not homogeneous, depending on their particular viewpoint. For instance, we assume that prices of fruits don't include taxes in database 1, while they do in databases 2, 3 and m.

In order to integrate meaningfully knowledge coming from the different databases, one has to carefully consider extra information that is left implicit in the representation of knowledge itself. In this example, this consists in

- (i) carefully represent the differences between the domains of the representation; and
- (ii) represent the different interpretation of the predicate $hasprice(x, y)$ (meaning that fruit x costs y GBP) in the different databases, depending on the fact that prices do include or don't include taxes.

In [25] the relations between the different domains of the representation are represented by introducing *domain relations*, i.e., relations between the interpretation domains of the different databases. A domain relation may, for instance, relate a "more abstract" object (e.g. apple) in the domain of a database to a set of "less abstract" objects (e.g. red−apple, green−apple) in the domain of another database. In the same paper, different perspectival views on related information (e.g., the different perspectival views on $hasprice(x, y)$ in our example) are represented by using *interpretation constraints.* An interpretation constraint is a relation between formulae contained in the languages of the different databases. For instance the different (but related) meaning of the predicate $hasprice(x, y)$ in database 1 and in the database m are represented by using the following expression:

$$1 : hasprice(x, y) \to m : \exists y' hasprice(x, y') \wedge y' = y + (0.07 * y)$$

Its meaning is that every time the models of database 1 satisfy the formula

$$hasprice(x, y)$$

then the models of the mediator database must satisfy the formula

$$\exists y' hasprice(x, y') \wedge y' = y + (0.07 * y)$$

which means that item x has price y' which is obtained by adding the correct amount of taxes to y.

As we can see from the example, the goal here is not to integrate the four databases in a unique schema, described using the same language, and interpreted over the same domain. On the contrary, Local Models Semantics aims at meaningfully relate knowledge coming from the different databases.

The divide-and-conquer approach Farquhar et al., [19] uses Buvač and Mason's logic of context to provide a framework for information source integration. The main idea is to represent the information contained in each information source (database) by mean of two contexts [19]:

> The **information source context** is a direct translation of a database schema into logic *without* resolving semantic conflicts [...]. The **semantic context** holds the translation with the semantic conflicts resolved.

In order to integrate the different information sources, an **integrating context**, containing axioms that lift from several semantic contexts is added, then providing a global schema of the integrated system. Figure 10 gives a graphical representation of the structure of contexts involved.

In this respect Farquhar et al. are a prototypical example of divide-and-conquer approach. Semantic contexts are used to solve semantic conflicts between databases so that the global model (schema) of the world can be reconstructed in the integrating context.

3.3.2 Partitioning knowledge bases

A context-based approach to the problem of specifying redundancy among different databases, maintaining an hight degree of autonomy has been proposed by Mylopoulos et al. [45]. Mylopoulos et al. describe a set of criteria for splitting a database in a set of (possibly overlapping) partitions. This work is particularly relevant as it provides mechanisms for the management of different overlapping partitions based on the notion of (possibly overlapping) *context*.

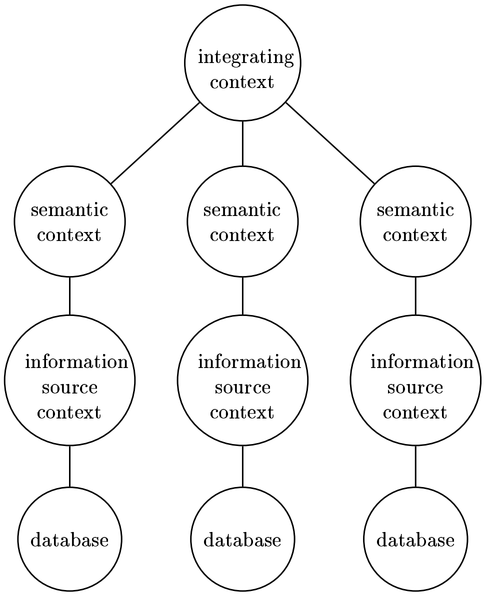

Figure 10: A divide-and-conquer approach to information integration.

In Mylopoulos et al., an information base is composed by units. A unit might represent an entity, object, attribute, relationship, rule, method, etc. A context is a special unit representing the decomposition of an information base in which they appear.

> The definition of each context includes three components which define respectively the *contents* of the context, the local names (*lexicon*) used for units in the context's contents and authorization rules for combinations of different users and transactions:
>
> $P(\texttt{Units}) \times (\texttt{Identifiers} \leftarrow \texttt{Units}) \times \texttt{Predicates} \supseteq \texttt{Contexts}$

That is, a context `c` is a triple where:

- the first element is a set of units (i.e., an element of $\mathrm{P}(\texttt{Units})$) which define the content of the context. Notationally, it is indicated by `contents(c)`. If a certain unit `o` belongs to `contents(c)`, then `o` is *visible* in `c`.
- the second element is a mapping from `Identifiers` to `Units` (i.e., an element of `Identifiers` ← `Units`) which defines the lexicon of the

context. Notationally, it is indicated by `lexicon(c)`. Mylopoulos et al., assume identifiers to be sequences of characters distinguished by enclosing quotes. Thus, `'john'` is an identifiers. The selector function `Identifiers` ← `Units` maps `'john'` into a specific unit, and this mapping depends on the context `c`.

- the third element is a set of formulae which determine whether a certain user, say `u`, is authorized to execute a certain transaction, say `t`, in the context. Notationally, it is indicated by `authorP(c)`. An example of authorization rules is

  $(\texttt{AssignSal(t)} \wedge \texttt{Employee(arg(t))}) \Longrightarrow (\texttt{Manager(u)}) \wedge \texttt{Manages(u, arg(t))}$

  which says that a transaction `t` of type `AssignSal` with argument an `Employee` can only be carried out if the user `u` is a `Manager` who whishes to assign a salary to one of his/her employees within context `c`.

In a nutshell, Mylopoulos et al., use the notion of context for representing the fact that a set of units can be partitioned into a set of different, partially overlapping modules, and that each of the modules can assign different Identifiers to the units and have different transaction rules. Each context has its own content, lexicon, and transaction rules. Relations between context play an important role in Mylopoulos et al. In fact, a (strong) relation existing between different contexts is given by the fact that, although named with different identifiers, the same unit can belong to two (or more) contexts. In particular Mylopoulos et al., is interested in establishing mechanisms for change propagation, that is mechanisms that establish whether the effects of a change operation performed on a certain unit in a certain context are (or are not) visible in other contexts [45].

> The effects of a change operation performed on a certain unit, say `o`, with respect to one context are not automatically visible in other contexts which also contain `o`. The visibility of a change in other contexts depend on whether that change is *propagated.*
>
> In particular, change propagation between two contexts is declared through two complementary operations
>
> $\texttt{propagationFrom}(\texttt{c}_1, \texttt{fromContext}, \texttt{authorP})$
>
> $\texttt{propagationTo}(\texttt{c}_2, \texttt{toContext}, \texttt{authorP})$
>
> which have to be executed with respect to the contexts $\texttt{c}_1, \texttt{c}_2$ which are respectively importing and exporting changes. For example, suppose we want to define context $\texttt{c}_2$ as a view which

> contains a subset of the contents of `c`$_1$, [. . .]. This can be accomplished by first creating `c`$_2$ and then setting up a propagation channel from `c`$_1$ to `c`$_2$:
>
> `c`$_2$:= `nexContext`([. . .])
> . . .
> `propagationFrom`(`c`$_1$, `c`$_2$, `true`)
> . . .
> `propagationTo`(`c`$_2$, `c`$_1$, `true`)
>
> [. . .] Once these operations have been executed, we will say that there exists a *propagation channel* from `c`$_1$ to `c`$_2$.

The work described in [45] has strong connections with the divide-and-conquer approach. The same problem of partitioning knowledge bases seems to impose the existence of a global model of the world (the initial knowledge base), and uses context as a device to focus on (smaller, simpler) portions of this model. Nonetheless, the mechanism of propagation channels can be seen as a mechanism for diversifying the different partitions so that they can express some partial viewpoint on this general model of the world. In fact, although different contexts may in principle overlap, only the existence of a propagation channel makes this overlapping count. If no propagation channel is established the overlapping is totally irrelevant as the changes made to units which are in a context can't be propagated to other contexts containing the same unit.

3.3.3 Very Large Knowledge Bases: CYC

CYC is an attempt to build a massive knowledge base so that one (person or machine) can apply it to some reasoning mechanism. The importance of building very large knowledge bases as one of the steps for developing programs with some sort of intelligence, is well motivated in [40]. In this paper the authors name the lack of very large knowledge bases, covering a wide area of (human) knowledge as one of the reasons why "expert systems" failed in their attempt to be regarded as intelligent programs [40].

> Suppose an expert system has the following four rules:
>
> IF frog(x), THEN amphibian(x)
> IF amphibian(x), THEN laysEggsInWater(x)
> IF laysEggsInWater(x), THEN livesNearLotsOf(x, Water)
> IF livesNearLotsOf(x, Water), THEN ¬livesInDesert(x)

> Given the assertion frog(Freda), those rules could be used to conclude that various facts are true about Freda: amphibian(Freda), laysEggsInWater(Freda), ¬livesInDesert(Freda), etc. Yet the program would not "know" how to answer questions like: Does Freda lays eggs? Is Freda sometimes in water?
>
> Humans can draw not only those direct conclusions from laysEggsInWater(Freda), but can also answer slightly more complex queries which require a modicum of "outside" knowledge: Does Freda live on the sun? Was Freda born live or from an egg? Is Freda a person? [...]
>
> Carefully selecting just the fragment of relevant knowledge leads to adequate but "brittle" performance: when confronted by some unanticipated situation, the program is likely to reach the wrong conclusion.

Unfortunately, the solution of building a knowledge base composed of a million rules has several drawbacks [39]:

> During the 1984-1989 time period, as the CYC common sense knowledge base grew ever larger, it became increasingly difficult to shoehorn every fact and rule into the same flat world. Finally, in 1989, as CYC exceeded 100,000 rules in size, we found it necessary to introduce an explicit context mechanism. That is, we divided the KB up into a lattice of hundreds of contexts, placing each CYC assertion in whichever context(s) it belonged.

Contexts in CYC have a fine internal structure. Lenat identifies a dozen mostly-independent dimensions along which contexts vary: Absolute Time, Type of Time, Absolute Place, Type of Place, Culture, Sophistication/Security, Granularity, Epistemology, Argument-Preference, Topic, Justification, and Anthropacity. In other worlds, each region of this 12-dimensional space implicitly defines a context [39]. The capability of importing an assertion from one context into another is provided by lifting assertions similar to the ones described in Section 3.1.

Differently from Mylopoulos et al., CYC follows entirely the divide-and-conquer approach. Contexts are used precisely for partitioning the CYC knowledge base, which provides the global model of the world, into (smaller, simpler) portions.

4 Conclusions

In the paper, we suggested that there are two types of theory of context in KRR: the first, which we called *divide-and-conquer*, sees context as a way of partitioning into smaller and simpler pieces a global model of the world; the second, which we called *compose-and-conquer*, sees context as a local theory of the world in a network of relations with other local theories. We discussed ins and outs of the two types, and showed that each type leads quite naturally to use context for addressing different issues in KRR, or to provide conceptually different solutions to the same issues.

As a conclusion, we'd like to raise the issue of the relationship between the work on context in KRR and the work on context in other disciplines, like philosophy of language and pragmatics. On the one hand, one might argue that the notion of context as the collection of a speaker's (or a hearer's) assumptions in a communication process is tightly related to the notion of context as a theory. In this respect, the formal machinery introduced in KRR might help in the analysis of some practical communication mechanisms. In particular, modeling the relationship between a speaker's intended meaning and a hearer's possible interpretations of a linguistic token might benefit from the notion of compatibility constraints between local models.

However, it is worth noting that the notion of context as the circumstance in which a token is uttered (also called an *utterance context*) is not straightforwardly be mapped onto the notions of context used in KRR. An utterance context has to do with a collection of features that define the external situation in which a linguistic expression was used by an agent. A typical example is Kaplan's definition of context as a quadruple of features that define the location of utterance: speaker, position, time, and possible world (the latter is used to give a semantics to adverbs like "actually" that indexically refer to the world in which an utterance takes place). This notion of context as a location does not seem to fit into the idea of a context as a collection of facts, either in a divide-and-conquer or in a compose-and-conquer theory. However, this contrast can be overcome if we realize that the notion of utterance context is traditionally defined from the point of view of an external observer, whose purpose is to determine the metaphysical content of what an agent says (the *content* à la Kaplan). The metaphysical content is not necessarily the same as the speaker's intended meaning; on the contrary, very often the two contents diverge. Thus, if one is interested in building a more cognitively oriented semantics of indexical and demonstrative expressions (i.e. a theory from the speaker's - or the hearer's – perspective), then the gap between the two notions of context is significantly reduced, as the utterance context can be interpreted as an agent's representation of the circumstance in which an utterance takes place. This representation (the agent's perspective on an utterance context) may have all the features of context in KRR: partiality (not every possible fact about the world would be taken into account on a given occasion), approximation (the current circumstance would be represented at some level of granularity), and perspective (the representation would be given from an individual's perspective). Our guess is that this approach to a semantics of indexical and demonstrative languages would allow people to solve many problems in the current semantic models. Some examples are: two agents may assign different content to the same indexical expression in the same utterance context (e.g. "today", if the two agents have different beliefs about the current date); two agents may assign different content to the same indexical expression because of different implicit assumptions (if one says "It's hot here", the extensions to be included in the interpretation of "here" may vary a lot, from the point of utterance to an entire country); and so on[9].

References

[1] G. Airenti, B. G. Bara, and M. Colombetti. Conversation and behaviour games in the pragmatics of dialogue. *Cognitive Science*, 17:197–253, 1993.

[2] V. Akman, P. Bouquet, R. Thomason, and R.A. Young, editors. *Modeling and Using Context*, volume 2116 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2001. Proceedings of CONTEXT'2001 – Third International and Interdisciplinary Conference on Modeling and Using Context (27–30 July 2001, Dundee, Scotland).

[3] V. Akman and M. Surav. Steps toward formalizing context. *AI Magazine*, 17(3):55–72, 1996.

[4] G. Attardi and M. Simi. A formalisation of viewpoints. *Fundamenta Informaticae*, 23(2–4):149–174, 1995.

[5] A. Ballim and Y. Wilks. *Artificial believers: The Ascription of Belief.* Lawrence Erlbaum, Hillsdale, NJ, 1991.

---

[9] Under this respect, the notion of utterance context can be partially assimilated to that of context as a set of an individual's beliefs. A tentative presentation of this argument can be found in [30].

[6] M. Benerecetti, P. Bouquet, and C. Ghidini. Formalizing belief report – the approach and a case study. In F. Giunchiglia, editor, *Artificial Intelligence: Methodology, Systems, and Applications (AIMSA'98)*, volume 1480 of *Lecture Notes in Artificial Intelligence*, pages 62–75. Springer, 1998.

[7] M. Benerecetti, F. Giunchiglia, and L. Serafini. Model Checking Multiagent Systems. *Journal of Logic and Computation, Special Issue on Computational & Logical Aspects of Multi-Agent Systems*, 8(3):401–423, 1998. Also IRST-Technical Report 9708-07, IRST, Trento, Italy.

[8] P. Bouquet and F. Giunchiglia. Reasoning about theory adequacy: A new solution to the qualification problem. *Fundamenta Informaticae*, 23(2–4):247–262, June,July,August 1995. Also IRST-Technical Report 9406-13, IRST, Trento, Italy.

[9] P. Bouquet and L. Serafini. Two formalizations of context: a comparison. Technical Report 0006-03, ITC-IRST, Trento, Italy, 2000.

[10] P. Bouquet, L. Serafini, P. Brezillon, M. Benerecetti, and F. Castellani, editors. *Modelling and Using Context*, volume 1688 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 1999. Proceedings of CONTEXT'99 – Second International and Interdisciplinary Conference on Modelling and Using Context (9-11 September 1999, Trento, Italy).

[11] P. Brézillon. Context in problem solving: A survey. *The Knowledge Engineering Review*, 14(1):1–34, 1999.

[12] S. Buvač, V. Buvač, and I.A. Mason. Metamathematics of Contexts. *Fundamentae Informaticae*, 23(3), 1995.

[13] S. Buvač and Ian A. Mason. Propositional logic of context. In R. Fikes and W. Lehnert, editors, *Proc. of the 11th National Conference on Artificial Intelligence*, pages 412–419, Menlo Park, California, 1993. American Association for Artificial Intelligence, AAAI Press.

[14] Saša Buvač. Quantificational logic of context. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996.

[15] B. F. Chellas. *Modal Logic – an Introduction.* Cambridge University Press, 1980.

[16] A. Cimatti and L. Serafini. Multi-Agent Reasoning with Belief Contexts: the Approach and a Case Study. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents: Proceedings of 1994 Workshop on Agent Theories, Architectures, and Languages*, number 890 in Lecture Notes in Computer Science, pages 71–85. Springer Verlag, 1995. Also IRST-Technical Report 9312-01, IRST, Trento, Italy.

[17] J. Dinsmore. *Partitioned Representations.* Kluwer Academic Publishers, 1991.

[18] A. F. Dragoni, P. Giorgini, and L. Serafini. Mental State Recognition from Communication. Technical Report 0007-03, ITC-IRST, Trento, Italy, 2000. Revised version submitted to the Journal of Logic and Computation.

[19] A. Farquhar, A. Dappert, R. Fikes, and W. Pratt. Integrating Information Sources Using Context Logic. In *Proceedings of AAAI Spring Symposium on Information Gathering from Distributed Heterogeneous Environments*, 1995.

[20] G. Fauconnier. *Mental Spaces: aspects of meaning construction in natural language.* MIT Press, 1985.

[21] M. Fisher and C. Ghidini. Programming Resource-Bounded Deliberative Agents. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 200–206. Morgan Kaufmann Publ., Inc, 1999.

[22] D. M. Gabbay and R. T. Nossum. Structured Contexts with Fibred Semantics. In *Proceedings of the 1st International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*, pages 46–55, Rio de Janeiro, Brazil, 1997.

[23] C. Ghidini. Modelling (Un)Bounded Beliefs. In P. Bouquet, L. Serafini, P. Brezillon, M. Benerecetti, and F. Castellani, editors, *Modelling and Using Context – Proceedings of the 2nd International and Interdisciplinary Conference, Context'99*, volume 1688 of *Lecture Notes in Artificial Intelligence*, pages 145–158. Springer Verlag - Heidelberg, 1999.

[24] C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127(2):221–259, April 2001.

[25] C. Ghidini and L. Serafini. Distributed First Order Logics. In D. Gabbay and M. de Rijke, editors, *Frontiers Of Combining Systems 2*, Studies in Logic and Computation, pages 121–141. Research Studies Press, 1998.

[26] C. Ghidini and L. Serafini. Information Integration for Electronic Commerce. In C. Sierra and P. Noriega, editors, *Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98)*, volume 1571 of *LNAI*, pages 189–206, Minneapolis, USA, May, 10 1998. Springer Verlag. Also accepted for presentation at UKMAS'98. Manchester, U.K. December 14–15, 1998.

[27] C. Ghidini and L. Serafini. Model Theoretic Semantics for Information Integration. In F. Giunchiglia, editor, *Proceedings AIMSA'98, 8th International Conference on Artificial Intelligence, Methodology, Systems, and Applications*, volume 1480 of *LNAI*, pages 267–280, Sozopol, Bulgaria, September, 21–23 1998. Springer Verlag. Also IRST-Technical Report 9807-03, IRST, Trento, Italy.

[28] E. Giunchiglia and F. Giunchiglia. Ideal and Real Belief about Belief. In *Practical Reasoning, International Conference on Formal and Applied Practical Reasoning, FAPR'96*, number 1085 in Lecture Notes in Artificial Intelligence, pages 261–275. Springer Verlag, 1996.

[29] F. Giunchiglia. Contextual reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, XVI:345–364, 1993. Short version in Proceedings IJCAI'93 Workshop on Using Knowledge in its Context, Chambery, France, 1993, pp. 39–49. Also IRST-Technical Report 9211-20, IRST, Trento, Italy.

[30] F. Giunchiglia and P. Bouquet. Introduction to contextual reasoning. An Artificial Intelligence perspective. In B. Kokinov, editor, *Perspectives on Cognitive Science*, volume 3, pages 138–159. NBU Press, Sofia, 1997. Lecture Notes of a course on "Contextual Reasoning" of the European Summer School on Cognitive Science, Sofia, 1996.

[31] F. Giunchiglia, E. Giunchiglia, T. Costello, and P. Bouquet. Dealing with Expected and Unexpected Obstacles. *Journal of Experimental and Theoretical Artificial Intelligence*, 8, 1996. Also IRST-Technical Report 9211-06, IRST, Trento, Italy.

[32] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics or: how we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994. Also IRST-Technical Report 9110-07, IRST, Trento, Italy.

[33] F. Giunchiglia, L. Serafini, E. Giunchiglia, and M. Frixione. Non-Omniscient Belief as Context-Based Reasoning. In *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pages 548–554, Chambery, France, 1993. Also IRST-Technical Report 9206-03, IRST, Trento, Italy.

[34] R.V. Guha. *Contexts: a Formalization and some Applications.* PhD thesis, Department of Computer Science, Stanford University, 1991. Also report No. STAN-CS-91-1399-Thesis.

[35] R.V. Guha. Contexts: a Formalization and some Applications. Technical Report ACT-CYC-423-91, MCC, Austin, Texas, 1991.

[36] B. Kokinov. A Dynamic Approach to Context Modelling. In P. Brezillon and S. Abu-Hakima, editors, *Working Notes of the IJCAI-95 Workshop on "Modelling Context in Knowledge Representation and Reasoning"*, Montreal (Canada), 1995.

[37] M. Lee and Y. Wilks. An ascription-based approach to speech acts. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, Denmark, 1996.

[38] D. Lenat. The Dimensions of Context Space. Technical report, CYCorp, 1999. `http://www.cyc.com/context-space.rtf,doc,txt`.

[39] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[40] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, and M. Shepherd. Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49, August 1990.

[41] J. McCarthy. Programs with Common Sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91. London: Her Majesty's Stationery Office, 1959. Also in V. Lifschitz (ed.), *Formalizing common sense: papers by John McCarthy*, Ablex Publ., 1990, pp. 9–20.

[42] J. McCarthy. Epistemological Problems of Artificial Intelligence. In *Proc. of the 5th International Joint Conference on Artificial Intelligence*, pages 1038–1044, 1977. Also in V. Lifschitz (ed.), *Formalizing common sense: papers by John McCarthy*, Ablex Publ., 1990, pp. 77–92.

[43] J. McCarthy. Generality in Artificial Intelligence. *Communications of ACM*, 30(12):1030–1035, 1987. Also in V. Lifschitz (ed.), *Formalizing common sense: papers by John McCarthy*, Ablex Publ., 1990, pp. 226–236.

[44] J. McCarthy. Notes on Formalizing Context. In *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pages 555–560, Chambery, France, 1993.

[45] J. Mylopoulos and R. Motschnig-Pitrik. Partitioning Information Bases with Contexts. In *Third International Conference on Cooperative Information Systems*, Vienna, 1995.

[46] P. Noriega and C. Sierra. Towards Layered Dialogical Agents. In J.P. Müller, M.J. Wooldridge, and N.R. Jennings, editors, *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), ECAI-96*, volume 1193 of *LNAI*, pages 173–189, Budapest, Hungary, 1996.

[47] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[48] J. Sabater, C. Sierra, S. Parsons, and N.R. Jennings. Engineering executable agents using multi-context systems. *Journal of Logic and Computation*, 2001. To appear.

[49] Dan Sperber and Deirdre Wilson. *Relevance: Communication and Cognition.* Basil Blackwell, 1986.

[50] 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, August 1998. IEEE Computer Society.

[51] R.H. Thomason. Type theoretic foundation for context, part 1: Contexts as complex type-theoretic objects. In P. Bouquet, L. Serafini,

P. Brezillon, M. Benerecetti, and F. Castellani, editors, *Modeling and Using Context – Proceedings of the 2nd International and Interdisciplinary Conference (9-11 September 1999, Trento, Italy)*, volume 1688 of *Lecture Notes in Artificial Intelligence*, pages 351–360. Springer Verlag - Heidelberg, 1999.

[52] R.W. Weyhrauch. Prolegomena to a Theory of Mechanized Formal Reasoning. *Artificial Intelligence*, 13(1):133–176, 1980.