Chapter 24
Multi-context Logics—A General Introduction

Chiara Ghidini and Luciano Serafini

Abstract Multi-context logics (MCLs) constitute a family of formalisms that allow one to integrate multiple logical theories (contexts) into an articulated structure, where different theories can affect one another via so-called bridge-rules. In the past 20 years multi-context logics have been developed for contexts in propositional logics, first order logics, description logics and temporal logic. Each of these logics has been developed, in an independent manner, for representing and reasoning about contextual knowledge in a specific application domain instead of originating from a single general formal framework. The absence of such a general formal framework for Multi-Context Systems (MCS), from which to extract tailored versions for the different application domain, has led to the development of a rather heterogeneous family of formal systems, whose comparison is sometimes very difficult. Being able to represent all these systems as specifications of a general class would be very useful as, for instance, one could reuse results proven in one MCS in another one. In this chapter, the authors provide an a-posteriori, systematic, and homogeneous description of the various MCSs introduced in the past. The authors do this firstly by providing a general definition of the MCS framework with its main components, which is general enough to capture the various versions of MCSs. Then, an account of the main logical specialisations of the MCS framework is provided, with an explanation of the domain of application they have been developed for.

24.1 Introduction

Multi-context logics (MCLs) are a family of formalisms for the integration of multiple logical theories (contexts) in an articulated composite structure. They are based upon two key principles of contextual knowledge, named principle of *locality* and principle of *compatibility* (Ghidini and Giunchiglia 2001). The principle of locality states that a context represents (or, in a more technical fashion, axiomatizes in a

C. Ghidini (✉) · L. Serafini
Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy
e-mail: ghidini@fbk.eu

L. Serafini
e-mail: serafini@fbk.eu

P. Brézillon, A. J. Gonzalez (eds.), *Context in Computing*,
DOI 10.1007/978-1-4939-1887-4_24

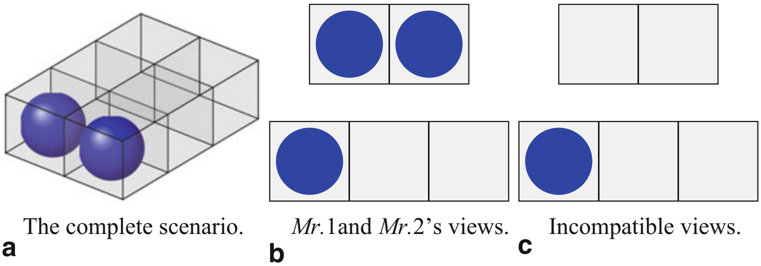


Fig. 24.1 The magic box. **a** The complete scenario. **b** *Mr.* 1 and *Mr.* 2's views. **c** Incompatible views

logical theory) a portion of the world, and that every statement entailed by such a representation is intended to hold within that portion of the world. The principle of compatibility instead states that different contexts that describe overlapping portions of world are represented by compatible logical theories, which are constrained to describe compatible situations. To illustrate these principles, and the idea of context underlined by MCLs we recall here the magic box example originally introduced in Ghidini and Giunchiglia (2001).

Example 1 Consider the scenario of Fig. 24.1a. Two observers, *Mr.* 1 and *Mr.* 2, are looking at a rectangular glass box from two different perspectives, one from the front, and one from the side. The box consists of six sectors, each sector possibly containing a ball. The box is "magic" and observers cannot distinguish the depth inside it. Figure 24.1b shows what *Mr.* 1 and *Mr.* 2 can see in the scenario depicted in Fig. 24.1a.

In this example we have two contexts, each context describing what an observer sees (its viewpoint) and the consequences that it is able to draw from it. The content of the two contexts is graphically represented in Fig. 24.1b. Notice that here the term context does not refer to a particular circumstance, or state of affair, but it refers to the point of view of each agent. Indeed we use the expression "the context of agent 1 (or 2)" to indicate his/her point of view.

Concerning *locality*, both *Mr.* 1 and *Mr.* 2 have the notions of a ball being on the right or on the left. However there may be situations in which there is a ball which is on the right for *Mr.* 1 and not on the right for *Mr.* 2. Furthermore $Mr.2$ has the notion of "a ball being in the center of the box" which is meaningless for $Mr.1$.

Concerning *compatibility*, the partial representations of *Mr.* 1 and *Mr.* 2's contexts are obviously related. The relation is a consequence of the fact that *Mr.* 1 and *Mr.* 2 see the same box. Figure 24.1b shows a pair of two compatible representations (contexts), while Fig. 24.1c shows a pair of incompatible representations (contexts). In this simple example we can synthetically describe all the compatible representations using a narrative like: "if *Mr.* 1 sees at least a ball then *Mr.* 2 sees at least a ball".

The MCL representing the magic box scenario is composed of:

- Two logical theories T_1 and T_2, each of them containing the logical representation of the context that describes one of the observer's viewpoints over the box. Each logical theory T_i will be described using an appropriate, and possibly different, logic L_i, interpreted in its own set of local models.
- A description of how to constrain the individual logical theories (and similarly the underlying logical models) in pairs that represent compatible viewpoints.

By generalising from the above example, the basic framework of MCLs is constituted by a number of logical theories T_i, each of them used to represent a context by means of an appropriate logic L_i, plus a description of how to combine/constrain the individual logical theories in compatible sequences that represent the entire multi-context structure. In the past 20 years this basic framework of MCL has been developed to model contexts described by means of different types of logic. In this chapter we provide an account of the main logical specialisation of MCLs, namely propositional logic and first order logic Multi-context Logics with an explanation of the types of applications they have been used for.

The chapter is structured as follows. In Sect. 24.2 we introduce the general definition of Multi Context Logic with its basic components, namely: syntax, semantics, logical consequence and deductive system (in literature called Multi Context System). MCLs can be categorized in two main families, namely: propositional MCL and quantificational MCL. In Sect. 24.3 we introduce the general definition of propositional MCL and some of the MCL that have been proposed in the past. In Sect. 24.4 we introduce quantificational MCLs and its two main important instances: Distributed First Order Logics and Distributed Description Logics.

24.2 Multi-context Logics

It its more general form, a *multi-context logic* (MCL) is defined on a family of logics $\{L_i\}_{i\in I}$ where each L_i is a logic used to formalize the i-th context. We assume that each logic L_i is equipped with a formal language, a class of structures in which this language is interpreted, a satisfiability relation (denoted by $\models_i$) which defines when a formula is true or false in an interpretation structure, and a logical consequence relation (also denoted by $\models_i$), that states when a formula is a logical consequence of a set of formulas of the language of L_i.

Languages of MCL

We distinguish two main categories of MCL: propositional and quantificational MCL, depending on the fact the languages associated to each context, are only propositional or quantificationals. By propositional languages we refer to logical languages that contain only expressions that express that a certain state of affair has a certain truth value (independently from the specific truth value). Quantificational languages extend propositional languages with the possibility of specifying objects of the domain, by means of special expressions usually called terms. In the case of quantificational multi-context logic we extend the language of L_i (with $i \in I$) with a set of terms called *arrow variables*, denoted as $x^{\rightarrow j}$ and $x^{j\rightarrow}$, (with x variable and $j \in I$). Arrow variables are used to point to objects to other domains. The formal semantic will be clarified later.

Multi-context Structure

The set I of context indexes (aka context names) can be either a simple set, or a set equipped with an algebraic structure such as, for instance a partial order, a lattice, a linear order, and possibly a set of operations on context indices. For instance, a partial order structure $\langle I, \prec\rangle$ can be used to represent a set of contexts which are organized according to a general-specific relation. For instance if FOOTBALL is the context (theory) that formalizes the domain of football, while SPORT is the context (theory) that formalizes the more general domain of sport, the fact that the football domain is more specific than the sport domain can be captured by imposing FOOTBALL$\prec$SPORT in I. A discrete linear order can be used to represent the evolution of the knowledge of one or a group of agents, where each context formalizes the agents' knowledge state at a given stage. For instance if ICD10 is the context that describes the international classification diseases - version 10, and ICD11 is the context that describes the the next version of the same classification, then ICD10 $\prec$ ICD11 states that ICD11 is the subsequent version of ICD10. Finally a lattice structure can be adopted to represent knowledge which holds in convex time intervals (represented by pairs of time points $\langle start, end\rangle$. The containment relation between intervals, represents the fact that the temporal span of a context covers the temporal span of another context.

Multi-context Model

A *model* for a multi-context logic $\{L_i\}_{i\in I}$ is a class of functions C where each function $c \in C$ assigns to each element $i \in I$ a set of interpretations c_i for the logic L_i. Each element of c_i is called a *local model* of L_i, and every $c \in C$ is called a *chain*[1]. Figure 24.2 provides a graphical illustration of a chain for a set of four contexts. A multi-context model is also called *compatibility relation* to emphasize the fact that it describes a class of compatible combinations of local models that mimic the type of relation that is assumed to hold between the original contexts they represent.

Some additional definitions are necessary to define the semantics for quantificational MCL. Quantificational logics extend propositional logics with the capability of predicating properties of objects of a universe, by introducing a class of expressions that denote objects of a domain. These expressions are usually called terms. As happens for propositional formulas, terms in different contexts can have different

---

[1] The term "chain" is slightly misleading, as it suggests that the set of contexts are structured in a total order (i.e., a chain) which might not be the case. Historically total ordered context structure was the first form of multi-context logic that has been studied. This made it natural to use the term "chain" for $c \in C$. For the sake of notation this terminology was maintained also in more complex MCLs with different context structures as the one depicted in Fig. 24.2.

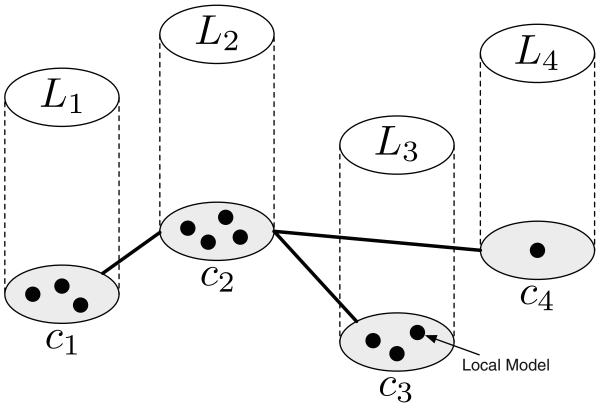


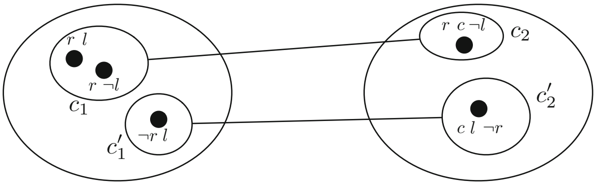
Fig. 24.2 A multi-context model. Each language is interpreted in a set of models. Each combination of sets of local models defines a chain and belongs to a compatibility relation. Graphically (and formally) this can be thought as an iper-arc on the graph of a set of sets of local models

meaning. Classical examples of terms with context dependent meaning taken from the area of formal linguistics are indexicals (like "here", "now", "me"); other examples can be found in the area of heterogeneous information integration, where a term can be used differently in different information resources (e.g., "Trento" in a database can be used to denote the province of Trento, while in another one is used to denote the city of Trento). Conversely, meaning of terms in different contexts can be related. To capture the relations between elements of different domains of interpretation we introduce the notion of *domain relation*. More precisely, let Δ^m be the domain of a local model m, and $\Delta_i = \bigcup_{m \in c_i} \Delta^m$, $\Delta_j = \bigcup_{m \in c_j} \Delta^m$ be the domains of interpretation for the models in c_i and c_j respectively, then a *domain relation* r_{ij} from i to j is any subset of $\Delta_i \times \Delta_j$.[2]

For instance suppose that A is the context corresponding to the database of books available on a web catalogue (say Amazon), and B is the database of the physical copies of books available in a library (say Biblioteca di Trento). A chain for the MCL composed of A and B is a pair $\langle c_A, c_B \rangle$. To represent the correspondence between the books titles available on Amazon and the book copies available in the Library of Trento, we can use the r_{AB}, that contains each pairs $\langle a, b_1 \rangle, \dots, \langle a, b_k \rangle$ where $b_1, \dots, b_k$ are the $k \geq 1$ copies of the book a sold by Amazon, which are available in the library of Trento.

[2] The domain relation is used to represent the overlapping between the domains of two contexts. Usually, in databases, or in ontology integration scenarios, the overlapping between two domains Δ_i and Δ_j is represented by imposing that $\Delta_i \cap \Delta_j$ contains a set of elements which are supposed to exists both in the domain of the ith context and of the jth context. The usage of a domain relation turns out to be more flexible than assuming domain intersection since it allows to integrate knowledge defined over overlapping but heterogeneous domains of interpretation. The typical case is the one of two databases that adopt a different level of abstraction to represent a specific domain. For instance, time at the level of day, and time at the level of hours.

Fig. 24.3 A MCL model



Multi-context Satisfiability

Satisfiability is a relation that spans between a model and formulas belonging to different logical languages, which are not necessarily disjoint. This introduces the necessity to distinguish between formulas that occur in different contexts. A *labelled formula* is an expression of the form $i : \phi$ where ϕ is a well formed formula of L_i. The intuitive meaning of $i : \phi$ is that ϕ holds in the i-th context.

Local satisfiability, that is, the satisfiability of a formula ϕ in a context i, is defined w.r.t. the local models and, possibly, the assignments to the free variables occurring in ϕ in case of quantificational contexts. Intuitively, a labelled formula $i : \phi$ is satisfied by a model C if all the local models $m \in c_i$ for all the chains $c \in C$ satisfy ϕ.

To make a simple example, consider the MCL model C for the magic box scenario depicted in Fig. 24.3. As explained in the introduction, this scenario can be formalized with two contexts 1 and 2 that formalise the points of view of *Mr.* 1 and *Mr.* 2, respectively. The two contexts are associated with two propositional logics L_1 and L_2 respectively, defined over the sets of propositional atoms $\{l, r\}$ and $\{l, c, r\}$[3]. Intuitively, we aim at introducing a definition that says that C satisfies the formula $1 : \neg r \vee l$ as the two elements c_1 and c'_1 belonging to the (only) two chains c and c' in the model C satisfy the formula $\neg r \vee l$ (where, in turn, the fact that c_1 and c'_1 satisfy $\neg r \vee l$ means that all the local models they contain satisfy that formula according to the notion of satisfiability in the appropriate logic, propositional in this case).

The above definition is sufficient for propositional contexts and also for quantificational contexts, if ϕ does not contain free variables. However, the general definition should also take into account the case in which ϕ contains *free and arrow variables, which need to be assigned* to the elements of the domains of the models in c_i. Notice however the models of c_i could have different domains of interpretation, so variables need to be assigned so that they are meaningful in all the models $m \in c_i$. i.e., to the intersections of the domains of the models in c_i. So if a is an assignment for the variable x, since we want to maintain the definition that $c \models i : \phi(x)[a_i]$ if $m \models \phi(x)[a_i]$ for all $m \in c_i$, then x should necessarily be assigned by a_i to some element which is in the intersection of the domains of each $m \in c_i$. More formally, a *local assignment* a_i should map every (arrow) variable x in an element of $\bigcap_{m \in c_i} \Delta^m$, where Δ^m is the domain of interpretation associated to the model m. We make the additional assumption that such an intersection is non-empty.

[3] Where l, c and r stand for *l*eft, *c*enter and *r*ight, respectively.

Formally, for any formula $i : \phi$, for every multi-context model C, for every chain $c \in C$ and for every assignment a_i that assigns the free variables occurring in $i : \phi$ to the intersection of the domains of interpretation of the locals models in c_i, we say that C *satisfies* $i : \phi$ w.r.t. the assignment a_i if for all $c \in C$, $c \models \phi[a_i]$, where $c \models \phi[a_i]$ means that $m \models_i \phi[a_i]$ for all the local models m of the i-th element $c_i \in c$, and $\models_i$ is the satisfiability relation defined in the logic L_i. We indicate that C models $i : \phi$ with the symbol $C \models i : \phi[a_i]$. When we have to evaluate a set of labelled formulas Γ that span over multiple contexts free variables, as all the other symbols, are locally interpreted, and therefore we need to have an assignment a_i for each context $i \in I$. This is called *local assignment*. An *MC-assignment* (or simply an assignment) is a family of assignments $a = \{a_i\}_{i \in I}$ such that for each $i \neq j \in I$, a_i assigns every variables of L_i which is not an arrow variable, and if L_i and L_j are quantificational logics, there is a domain relation r_{ij} such that: if $a_j(x^{i\rightarrow})$ is defined then $\langle a_i(x), a_j(x^{i\rightarrow})\rangle \in r_{ij}$ and if $a_i(x^{\rightarrow j})$ is defined then $\langle a_i(x^{\rightarrow j}), a_j(x)\rangle \in r_{ij}$.

Multi-context Logical Consequence

In MCL the notion of logical consequence is defined over labelled formulas. In particular, if Γ is a set of labelled formulas and $i : \phi$ a labelled formula, then $i : \phi$ is a logical consequence of Γ if and only if,

1. there is a model C, a chain $c \in C$ and a family of assignments $a = \{a_i\}_{i \in I}$ to the free variables of $\Gamma \cup \{\phi\}$ such that $c \models \Gamma \cup \{i : \phi\}[a]$, and
2. for all models C, for all $c \in C$ and for all family of assignments $a = \{a_i\}_{i \in I}$ to the free variables of the formulas in Γ if $c \models \Gamma$ then there is an extension a' of the assignment a, to the free variables of ϕ such that $c \models i : \phi[a'_i]$[4].

Information Flow Across Contexts via Bridge Rules

In a MCL every context is interpreted in a set of local models, possibly arranged into chains. Local interpretation is the way to relate the truth and the falsity of the formulas to each context. However, only certain combinations of local interpretations are possible. Those are the ones admitted by the class of compatibility relations associated to a MCL. At the level of formulas, this means that there is a dependency between the truth of a (set of) formulas in a context and the truth of different formulas another context. To go back to our magic box scenario, this means that if a formula l (there is a ball in the left sector) is true in the context of Mr.1, then the formula $l \vee c \vee r$ (there is at least one ball in the box) must be true in the context of Mr.2.

From this perspective we can say that (classes of) compatibility relation(s) determine an information flow across contexts: the truth of a certain formula in a context

[4] In the definition of multi-context logical consequence there is an implicit existential quantification of the free variables in ϕ which are not free in Γ. This is similar to what happens for the semantics of rules in logic programming, where variables that appear in the head of a rule (the consequence) which are not contained in the body are usually interpreted existentially.

affects (imposes) the truth of another formula in a different context. *Bridge rules* are expressions over the languages of different contexts that enable the formalisation of this information flow. They are of the form:

$$i_1 : \phi_1, \ldots, i_n : \phi_n, \text{not } i_{n+1} : \phi_{n+1}, \ldots, \text{not } i_m : \phi_m \rightarrow i : \phi \tag{24.1}$$

with $0 \leq n \leq m$, $i_k \in I$ and ϕ_k a formula in the language of L_{i_k}. The intuitive reading of (24.1) is: "if $\phi_1, \ldots, \phi_n$ hold in $i_1, \ldots, i_n$ respectively and $\phi_{n+1}, \ldots, \phi_m$ do not hold in $i_{n+1}, \ldots, i_m$ respectively, then ϕ holds in i." Thus, a simple bridge rule that represents the propagation flow in the magic box example discussed above is

$$1 : l \rightarrow 2 : l \vee c \vee r. \tag{24.2}$$

Multi-context System

We are now ready to define an axiomatic system for multiple contexts. A *multi-context system MCS* in a multi-context logic $L_I = \{L_i\}_{i \in I}$ is a pair $\langle \mathbb{T}, \mathbb{BR} \rangle$ where $\mathbb{T}$ is a family of theories $\{T_i\}_{i \in I}$, with T_i a set of closed formulas in the logic L_i, and $\mathbb{BR}$ is a set of *bridge rules*. Intuitively, each T_i axiomatizes what is true in the logic L_i, while the bridge rules $\mathbb{BR}$ axiomatize the constraints imposed by the compatibility relations and act like cross-logic axioms.

Reasoning in Multi-context Systems

There are multiple reasoning systems for MCL. Depending on the local logics, different reasoning systems have been developed in the past. Often, reasoning methods for specific MCL are the result of the combination via bridge rules of local reasoning methods. The work in Giunchiglia and Serafini (1994), Ghidini and Serafini (1998) propose an extension of Natural Deduction for reasoning in propositional and first order MCLs; in Serafiniand Roelofsen (2005) the SAT decision procedure for propositional logic is extended to a context SAT (or C-SAT) procedure to check for satisfiability in propositional multi-context systems; in Ghidini (1999), Borgida and Serafini (2003) tableaux methods for reasoning in modal and description logics have been extended for MCLs based on modal/description logics; Brewka et al. (2007) extends answer set programming to deal with propositional MCLs with non-monotonic bridge rules; finally, Bozzato and Serafini (2013) shows how SROIQ2-RL rule based forward reasoning can be extended to deal with multi-context logics in which each context is associated to a semantic web language OWL2RL. In the remaining of the chapter we will briefly recall and describe the most important reasoning methods that have been developed for MCLs along with an explanation of their main usages.

Local and Global Inconsistency

The fact that in MCL knowledge is split in multiple theories, makes MCL a flexible framework for modelling various types of inconsistencies. A first form of inconsistency arise when a proposition is assumed to hold in a context and the negation of the same proposition is assumed to hold in another context. This is easily represented in MCL with the two formulas $i : \phi$ and $j : \neg\phi$, which, in general, can be managed without generating any form of inconsistency. This is similar to what happens in multi modal logic where the two propositions $\Box_i \phi$ and $\Box_j \phi$ do not interfere, unless there are specific axioms that connect the two modalities $\Box_i$ and $\Box_j$. In addition to this, in MCL we can define two forms of inconsistency. One is called *local inconsistency* and refers to the fact that in a particular context it is possible to derive contradictory statements, i.e., *for some* $i \in I$, $i : \phi$ and $i : \neg\phi$ are both derivable; the second is called *global inconsistency*, which refers to the fact that a contradiction is derivable in all the contexts, i.e., *for all* $i \in I$, $i : \phi$ and $i : \neg\phi$ are both derivable. In general local inconsistency does not entail global inconsistency. So it is possible that one context is locally inconsistent, while others are consistent. From the semantic perspective, local inconsistency in a context i corresponds to the fact that there are chains in the compatibility relation of an MCL where i is interpreted in the empty set of local model, while other contexts are associated with a non empty set of local models.

24.3 Propositional Multi-context Logic

The fisrt, and simplest, family of MCLs that was developed is based on an unstructured set I of contexts, where each context is described by means of a propositional logical language. Following the general definition, a *propositional multi-context logic* (PMCL) is defined starting from a family $\{P_i\}_{i \in I}$, where each P_i is a set of propositional variables. Each logic L_i is therefore described using a propositional language defined on P_i. A model (compatibility relation) C for PMCL is composed by a set of chains $c \in C$ where each c_i is a set of truth assignments to the propositional variables in P_i (that is, each c_i is a set of propositional models defined over P_i). Depending on the constraints one imposes on C it is possible to define various types of PMCS. In the following we provide three important examples of PMCS present in literature.

Partitioning Propositional Theories

One of the simplest ways of looking at multi-context logics is in terms of a partition of a (propositional) theory into a set of interacting *microtheories*. In this case the entire MCS is the (propositional) theory, the different contexts are the microtheories, and the compatibility relations (or analogously bridge rules) express the way microtheories are connected one to another. As explained in Amir and Mcilraith (2000), one of the main reason for partitioning a large (propositional) theory into a set of smaller interacting microtheories is efficiency of reasoning.

Partitioned propositional theories correspond to a *specific class* of compatibility relations for PMCL, which we indicate with C_{part}, that contain chains c defined as follows:

$$\text{for all } i \in I, \quad |c_i| = 1 \tag{24.3}$$

$$\text{for all } p \in P_i \cap P_j, \quad c_i(p) = c_j(p) \tag{24.4}$$

Condition (24.3) states that all the elements c_i of a chain contain exactly one local model and intuitively represents the fact that each chain can be considered as composed of different contexts (the different c_i) that have a *complete* representation of a scenario (from their point of view). For instance, the chain $c' = \langle c'_1, c'_2 \rangle$ in Fig. 24.3 satisfies this requirement and correspond to the scenario in which *Mr.* 1 sees a ball in the left sector, and no ball in the right sector, and *Mr.* 2 sees a ball in the center and left sectors and no ball in the right sector. Condition (24.4) states that the different elements c_i contained in a chain agree on the interpretation of the propositional variables that are common to the two elements. Intuitively this means that the two contexts described by, say, $c_i$andc_j agree on the truth value of the knowledge they have in common.

If we denote with $\models_{part}$ the logical consequence defined w.r.t. C_{part}, then we can state the following correspondence between a partitioned PMCL and propositional logic.

Theorem 1 *Let $T = T_1 \uplus \cdots \uplus T_n$ be a propositional theory on the set of propositions P, which is partitioned in n theories $T_i \subseteq T$ (for $1 \leq i \leq n$) defined on the set of propositional variables P_i, then: for every formula ϕ that contains only propositions in P_i, we have that*

$$T \models \phi \textit{ if and only if } 1 : T_1, \ldots, n : T_n \models_{part} i : \phi.$$

Partial Views

Relaxing conditions (24.3) and (24.4) enables to obtain a more general (that is, weaker) class of MCLs where each context can be considered as describing a partial view on the world. For example, in Fig. 24.3 element c_1 corresponds to a partial view of the two sector's box where *Mr.* 1 can state that there is a ball in the right sector but is uncommitted on whether there is a ball in the left hand side sector (e.g., because the sector is behind a wall as in an example shown in Ghidini and Giunchiglia (2001).

As shown in Roelofsen and Serafini (2004), this general formulation of PMCL is embeddable in the propositional multi-modal logic S5, with one modal operator $\Box_i$ for each context label $i \in I$. Local formulas of the form $i : \phi$ are translated in $\Box_i \phi$, and bridge rules of the form (24.1) are translated in the implication

$$\Box_{i_1} \phi_1 \wedge \cdots \wedge \Box_{i_n} \phi_n \rightarrow \Box_i \phi \tag{24.5}$$

The correspondence between this PMCL and multi-modal S5 is not an equivalence since modal logics has a *global language* allowing formulas that express relations between local models which are more complex that the one representable in terms of propagation rules. For instance the modal formula $\Box_i \phi \vee \Box_j \phi$ does not correspond

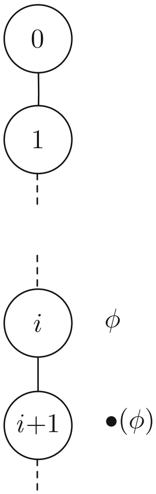


Fig. 24.4 A hierarchical meta structure

to any bridge rule between i and j as it cannot be represented as a propagation pattern. Another example is negated modal formulas like e.g., $\neg\Box_i\phi$, which is not expressible in MCL, as it states that a proposition does not hold in a context. This limitations in the expressivity reflects the fundamental assumption of MCL, i.e., that every formula should be stated in a context. This expressivity limitation turns out to be of great help in the definition of modular reasoning systems, since they prevent to express global inconsistency, since there is no global formula. The assumption of not permitting global logical operators, has been relaxed in the formulaization of non-monotonic MCL (Brewka et al. 2007), where the negation (as failure) operator is applicable to a labelled formula, obtaining $\mathbf{not}(i : \phi)$.

Hierarchical Meta Logics

In the work on propositional multi-context logics, a special effort was devoted to investigate the usage of these formalisms to formalize the "object and meta relation" between contexts, that is, the situation in which for each context one can define a meta context that predicates on what holds in the object context.

In this case I is the set of natural numbers with the usual total linear order, and each language L_i is a propositional language, such that for every formula ϕ in the language of L_i there is a propositional variable $\bullet(\phi)$ in the language of L_{i+1}, as depicted in Fig. 24.4.

The compatibility relation C_{OM} for a hierarchical meta logic satisfies the following constraints:

1. Closure w.r.t. union: If $c, c' \in C$, then $c \cup c' \in C$ (where $c \cup c' = \{c_i \cup c'_i\}_{i \in I}$).
2. Interpretation of meta-formulas: For all $c \in C$, $i \in I$ and ϕ in L_i, $c_i \models \phi$ if and only if $c_{i+1} \models \bullet(\phi)$.

The work in Giunchiglia and Serafini (1994) shows that this logic is equivalent to the modal logic K, when the "•" operator is translated in the modal operator $\Box$, while further works (see, e.g., Ghidini 1999) prove that by further restricting C it is possible to obtain the other normal modal logics, such as B, K4, K45, S4 and S5, and have applied these equivalence results to model propositional attitudes and multi-agent systems by means of a context-based approach (see, e.g., Cimatti et al. 1994; Benerecetti et al. 1998a; Benerecetti et al. 1998b; Fisherand Ghidini 2010).

24.4 Quantified Multi-context Logics

Quantified multi-context logics extend propositional MCL with the possibility of predicating object properties in different contexts and relations between objects. The two principles of MCL of locality and compatibility are extended to the contextual interpretation of terms. In details: according to the locality principle, each context is associated with a local domain. According to the compatibility principle, only certain combinations of local domains are admitted. For instance, if A and B are the contexts associated to two databases DB_A and DB_B, respectively, then the universe of A (i.e., the set of constants that appear in the relations of DB_A) can be completely distinct from the universe of DB_B. For instance, the two databases might use different identifiers, and different ways to denote attributes, and so on. On the other hand, if the intended domains of both DB_A and DB_B overlap, i.e., they contain information about a common subset of objects, say books, then the identifiers of books used in the two databases should be somehow related. As explained in the introductory section, the relation between local domains is modelled via the, so-called, domain relation.

Specific instances of quantified MCL have been developed with the scope of formalizing heterogeneous database integration, ontology integration, and ontology matching. They are all monotonic logics, and the local logics are either first order logic, or description logics. In the following subsections we introduce the two main quantified MCLs: Distributed First Order Logics (Ghidini and Serafini 1998) and Distributed Description Logics (Borgida and Serafini 2003).

24.4.1 Distributed First Order Logics

DFOL is a family of MCL that has been defined with the objective of formalizing contextual knowledge expressed in first order languages. One of the main motivation for DFOL is the formalization of heterogeneous relational database integration (Serafini and Ghidini 2004) and to provide a formal semantics for heterogeneous schema and ontology mapping (Serafini et al. 2007).

A DFOL is defined on a family of first order logics $\{L_i\}_{i \in I}$. A DFOL model is any compatibility relation $\{c\}$ composed of a single chain c where c_i, for all $i \in I$ is a (possibly empty) set of interpretations of L_i on the same domain $\Delta_i \neq \emptyset$. With respect of the original formalization described in Ghidini and Serafini (1998) we also admit the arrow variable $x^{i\to}$ and $x^{\to i}$ in the language L_i and the domain relation r_{ii}. This results in a more uniform treatment.

Consequently, this simplify the general definition on assignment, as this implies that the intersection of the domains of all the local models associated to the context i, is the same as the domain of each local model. The representational hypothesis which derives by assuming shared domain for all local models in c_i is the fact that at each context there is *complete knowledge on the size of the local domain*. Formally this corresponds to the fact that every formula ϕ that does not contain, constant symbols, functional symbols, and predicate symbols with the exception of the equality symbol, is such that $C \models i : \phi$ or $C \models i : \neg\phi$. i.e., all the ith local models agree on the evaluation of ϕ[5]. Examples of such formulas, are the those that allow to state bounds on the dimension of the domain. as, $i : \forall x_0, \ldots, x_m \bigvee_{0\leq i<j\leq m} x_i = x_j$, which states that ith domain contains at most m elements, and $i : \exists x_1, \ldots, x_n \bigwedge_{0\leq i<j\leq n} x_i \neq x_j$ that states that ith domain contains at least n elements. The assumptions, of constant local domains does not imply full constant domains, i.e., the fact that every domain in every context has the same dimension. Indeed, for instance the set of labelled formulas $\{1 : \forall xy.x = y,\ 2 : \exists xy.x \neq y\}$ is satisfiable, and they state that the domain of context 1 contain one element and the domain of context 2 contains at least two elements.

DFOL is the first example of MCL described in this chapter where logical consequence relation involves the assignment to variables. Under the assumption of constant local domains, we can simplify the definition of logical consequence as follows:

$$\Gamma \models \phi \text{ if for every chain } c \text{ and every assignment } a \text{ for all the variables in } \Gamma, \text{if } c \models \Gamma[a] \text{ then there is an extension of } a \text{ to } a' \text{such that } c \models i : \phi[a']. \tag{24.6}$$

24.4.1.1 Representing Cross Domain Constraints in DFOL

In the general case, i.e., when no constraints are imposed on the compatibility relation, the logical consequences across contexts is extremely week, and it is such that $\Gamma_{\neq i} \models i : \phi$ iff $\models i : \phi$ (where $\Gamma_{\neq i}$ is a set of labelled formulas with index different from i). As in all the other MCL's, also in DFOL it is possible to impose restrictions on the compatibility relation and on the domain relation by means of bridge rules. In the following we present some of the properties involving quantificational contextual information that can be formalized by means of DFOL bridge rules:

Absolute names. In general, in different contexts a constant (or a term) can have different meanings, however, it is also possible that the meaning of a term in a context is related to the meaning of another term in another context. An extreme

[5] Notice that, if $|c_i| > 1$, i.e., there is more than one local model, it is possible that $C \not\models i : \phi$ and $C \not\models i : \neg\phi$.

situation is when a constant is an *absolute/global name*. I.e., a constant that have the same meaning in all the context. Absolute names can be modelled by imposing the following bridge rule to be valid for every $i, j \in I$.

$$i : \mathrm{a} = x^{i\rightarrow} \qquad i : \mathrm{a} = x \rightarrow j : x^{i\rightarrow} = \mathrm{a} \qquad i : \mathrm{a} = x^{\rightarrow i} \rightarrow j : x = \mathrm{a} \quad (24.7)$$

The first of the above bridge rules imposes that the constant a is interpreted in a unique element by all local model in c_i. Indeed if c_i contains two models m and m' that interpret a in two different objects, then it is not possible that $m \models \mathsf{a} = x^{i\rightarrow}[a]$ and $m' \models \mathsf{a} = x^{i\rightarrow}[a]$, and therefore $c \not\models i : \mathsf{a} = x^{i\rightarrow}$. The other two bridge rules in (24.7) do not impose that a is interpreted in the same object in i and j, since Δ_i and Δ_j can be different (possibly disjoint) domains, but they state that the interpretation of a in contexts i and j corresponds via the translation defined by the domain relation among the domains of the two contexts. If the bridge rules in (24.7) are imposed for two individual constants a and b in the intersection of the languages of L_i and L_j, then we have that the following logical consequences hold.

$$i : \mathrm{a} \neq \mathrm{b} \models j : \mathrm{a} \neq \mathrm{b} \qquad i : \mathrm{a} = \mathrm{b} \models j : \mathrm{a} = \mathrm{b}$$

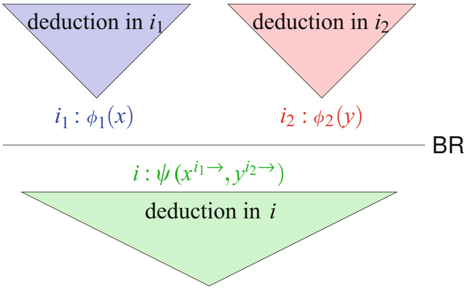
Imposing bridge rules (24.7) on the set of constants contained in the intersection of the universes of two databases DB_A and DB_B, corresponds to assume that the intersection of the universes of the two DBs are isomorphic, and therefore this allow to safely join informations about the intersected domain available in both DBs.

Constraints on the domain relation. Bridge rules can be used to formalize relations between domains in different contexts. For instance, in some situation it is useful to assume that the domains of two contexts (say i and j) are isomorphic. This can be forced by the bridge rules

$$\rightarrow j : \exists y.y = x^{i\rightarrow} \qquad \rightarrow i : \exists y.y = x^{\rightarrow j} \qquad i : x^{\rightarrow j} = y^{\rightarrow j} \rightarrow j : x = y$$

The first two of the above bridge rules imposes that for every element x of the domain of context i, there is a corresponding element of the domain of context j and, vice-versa for every element of the domain of context j, there is a corresponding element of the domain of context i. The third one states that the domain relation between i and j must be a function. In Ghidiniand Serafini (1998) we describe how many other properties can be formalised by bridge rules containing just the equality symbol and arrow variables.

Join among heterogeneous domains. Bridge rules can be used to express the fact that a certain knowledge in a database DB_C can be obtained by joining the information available in two heterogeneous databases DB_A and DB_B. As an example, suppose that we want to represent that the ternary relation $R(x, y, z)$ in the database DB_C is obtained by a join between the relations $P(x, y)$ in DB_A, and $Q(y, z)$ in DB_B over the argument y. But we know that the three databases have three heterogeneous representation of the values in the attributes, and therefore before doing the join it is necessary to perform a translation. There are three possible ways to proceed,



(1)	$1 : P(a)$	(1)	assumption
(2)	$1 : x = a$	(2)	assumption
(3)	$1 : P(x)$	$(1,2)$	from (1) and (2)
(4)	$2 : Q(x^{1\rightarrow})$	$(1,2)$	from (1) by BR
(5)	$2 : b = x^{1\rightarrow}$	(2)	from (2) by BR
(6)	$2 : Q(b)$	$(1,2)$	from (4) and (5)

Fig. 24.5 A DFOL proof

- Translate the tuples of P from DB_A into the DB_B, do the join in B and translate the result into C. This is represented by the bridge rules

$$A : P(x, y),\ B : Q(y, z) \wedge y^{A\rightarrow} = y \wedge x^{A\rightarrow} = x \rightarrow C : R(x^{B\rightarrow}, y^{B\rightarrow}, z^{B\rightarrow})$$

- Do the same but starting form DB_B, joining in DB_A and translating in DB_C, which is represented by the bridge rule:

$$B : R(y, z),\ A : P(x, y) \wedge y^{B\rightarrow} = y \wedge z^{B\rightarrow} = z \rightarrow C : R(x^{A\rightarrow}, y^{A\rightarrow}, z^{A\rightarrow})$$

- or transfer the tuples of P and Q into DB_C and do the join there. This way of reasoning is represented by the bridge rule:

$$A{:}P(x, y),\ B{:}Q(y, z),\ C{:}x{=}x^{A\rightarrow} \wedge y{=}y^{A\rightarrow} \wedge y{=}y^{B\rightarrow} \wedge z{=}z^{B\rightarrow} \rightarrow C{:}R(x, y, z)$$

24.4.1.2 Reasoning in DFOL

Being DFOL an extension of first order logic, reasoning in DFOL is an undecidable task but it is finitely axiomatizable. In Ghidini and Serafini (1998) we proposed a sound and complete Natural Deduction Calculus for DFOL logical consequence parametrized on a set of bridge rules BR. This calculus is sound and complete with respect to the class of DFOL models and the domain relations that satisfies the set of bridge rules BR. Natural Deduction systems for FOL is a set of inference rules, with an arbitrary (finite) number of premises and a single conclusion. A deduction of ϕ form a set of hypothesis $\phi_1, \ldots, \phi_n$ is a tree rooted at ϕ, with leaves $\phi_1, \ldots, \phi_n$, such that the father node is derived by applying an inference rule to it's children. The extension of ND to DFOL with bridge rules BR, is obtained by composing local deductions via bridge rules. Informally, the bridge rule $i_1 : \phi_1(x),\ \ i_2 : \phi_2(y) \rightarrow i : \psi(x^{i_1\rightarrow}, y^{i_2\rightarrow})$ allows to "plug in" a deduction in context i_1 the two deductions performed in context i_1 and i_2 that infers the premises of the bridge rule. In Fig. 24.5 we provide a graphical representation of local inference composition and an example of a simple proof in DFOL.

24.4.2 Distributed Description Logics

DDL has been introduced in Borgida and Serafini (2003) as a variation of multi-context logic with the motivation of modeling ontology matching and integration by means of a formal logic. In DDL local logics L_i's are description logics. The starting point of ontology mapping is constituted by two (a set of) ontologies, usually called source and target ontology. Ontology matching algorithms provides a set of semantic matches that partially maps the elements of the source ontology into the "corresponding" elements of the target ontology[6]. Once a source ontology is semantically matched with a target ontology, and every heterogeneity in the representation of knowledge by the two ontologies has been resolved, the knowledge contained in the two ontologies can be integrated and combined in a unique (sometimes modular) knowledge base. In many cases ontology matches act as information channels that propagate knowledge from the source ontology to the target ontology that is extended with the additional information coming from the source ontology. This perspective of ontology matching/integration can be naturally represented in multi-context logic, using context based on description logic languages. In DDL each context represents an ontology and semantic matches between a source ontology i and a target ontology j are represented via bridge rules with premises in i and consequences in j. A context i can contain concepts and role subsumptions, namely formulas of the form $i : C \sqsubseteq D$, $i : R \sqsubseteq S$, and assertions, namely statements of the form $i : C(a)$ and $i : R(a, b)$ where C and D are concept expressions, R and S are role expressions and a and b are individuals. A model for DDL is the same as a DFOL model on the FOL translation of the description logic language (where concepts, are unary predicate, relation binary predicates, and individual constants stays the same) with the restriction that for every chain c, and every $i \in I$, $|c_i| \leq 1$. DDL bride rules are used to represent ontology matches, and they can be defined among concepts, roles and individuals. Heterogeneous bridge rules has also been introduced, which maps concepts to roles and viceversa (e.g., "wedding" to "is-married-to"), but for simplicity we only report homogeneous bridge rules here: Bridge rules between concepts and roles are of two forms

$$i : C \xrightarrow{\sqsubseteq} j : D \qquad i : C \xrightarrow{\sqsupseteq} j : D \tag{24.8}$$

where C and D are concept expressions in L_i and L_j respectively. The above bridge rules are satisfied by the DDL model c if there is domain relation r_{ij} such that

$$r_{ij}(C^{c_i}) \subseteq D^{c_j} \qquad r_{ij}(C^{c_i}) \supseteq D^{c_j} \tag{24.9}$$

where X^{c_i} is the extension of the concept C in the unique model $m \in c_i$ or it is the empty set if $c_i = \emptyset$.

[6] The most general setting semantic matches are associated with weights (confidence value) but when mappings are crisp (i.e., confidence value is equal to 1) then they can be fruitfully formalized in two valued logics.

The intuitive meaning of DDL bridge rules can be easily induced from the satisfiability conditions (24.9). In particular the $\xrightarrow{\sqsubseteq}$ bridge rule, states that the concept C in i matches with some subconcept of D in j. While the $\xrightarrow{\sqsupseteq}$ bridge rule states that the concept C in i is mapped into some superconcept of D in j. Analogous bridge rules can be defined among roles. Bridge rules among individuals are expressions of the form

$$i : a \xrightarrow{=} j : b_1. \tag{24.10}$$

The bridge rule (24.10) is satisfied when $\langle a^{m_i}, b^{m_j}\rangle \in r_{ij}$ (where m_i and m_j are local models in c_i and c_j respectively). The intuitive meaning of the bridge rule (24.10) is that b is one of the possible translations in j of a in i.

Similarly to what happens for DL, which is a fragment of FOL, DDL is a fragment of DFOL. Indeed a DDL can be rewritteng into a DFOL by applying the standard translation of DL into FOL for each of the formulas in L_i, and by translating the bridge rule (24.8) into the following DFOL bridge rules:

$$i : C(x^{\rightarrow j}) \rightarrow j : D(x) \qquad j : D(x) \rightarrow i : C(x^{\rightarrow j}) \tag{24.11}$$

and translating the individual bridge rule (24.10) into

$$i : x = a \rightarrow j : x^{i\rightarrow} = b. \tag{24.12}$$

The semantics of DDL bridge rules entails a form of information propagation between mapped ontologies. The papers Serafini et al. (2004), Serafiniand Tamilin (2005), Ghidini et al. (2007) investigate on the knowledge propagation patterns between a source and a target ontology mapped with a set of DDL bridge rules. A simple example of such a propagation pattern from i to j induced by a pair of mappings from i to j is described by the following sound inference:

$$\frac{i : A \sqsubseteq B \qquad i : A \xrightarrow{\sqsupseteq} j : G \qquad i : B \xrightarrow{\sqsubseteq} j : H}{j : G \sqsubseteq H}.$$

The above propagation pattern in true for any unrestricted domain relation. However in many cases it's interesting to investigate on DDL models where the domain relation satisfies natural restriction, such as functionality or injectivity or compositionality. The more restricted the domain relation the more information is passed by the bridge rules. Detailed investigation of different propagation patterns depending on the restriction imposed on the domain relation are studied in Homola and Serafini (2010).

24.5 Conclusions

Research and implementation activities around multi context system has been carried out for the last 25 years with a number of significant results that include propositional MCLs, first order MCLs, description logics-based MCLs, and MCL for semantic web languages like RDF and OWL. The research activities have focused to the development of theoretical frameworks as well as a set of prototype implementations, among which DRAGO, a Distributed Reasoning Architecture for a set of ontology linked via ontology mappings (Serafini and Tamilin 2005), and the Contextualized Knowledge Repository (Bozzato et al. 2013), a system that extends standard RDF triple stores with the capability of reasoning with multiple RDF graphs (Contexts) linked via bridge rules.

A number of studies that compare multi context system with other logical formalism that support distributed knowledge representation have also been developed, and mappings between the different formalisms have been proposed. In particular Serafini and Bouquet (2000) presents a formal comparison between propositional MCL and the propositional logic of contexts based on modal logics proposed in Buvac and Mason (1993), while Brockmans et al. (2009) exploits DFOL for encoding and comparing several formalisms for ontology mappings.

In this chapter we have provided an overview of the main families of multi context logics (MCLs), a logical formalism that allow to integrate multiple logical theories (contexts) in a structure of inter-related contexts, a description of their main logical properties and an illustration of the types of applications they have been used for.

References

Amir, E., Mcilraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artif. Intell. **162**, 49–88 (2000)

Benerecetti, M., Bouquet, P., Ghidini, C.: Formalizing belief reports – The approach and a case study. In: Giunchiglia, F. (ed.) Proceedings AIMSA'98, 8th International Conference on Artificial Intelligence, Methodology, Systems, and Applications, *LNAI*, vol. 1480, pp. 62–75. Springer, Sozopol (1998a)

Benerecetti, M., Giunchiglia, F., Serafini, L.: Model checking multiagent systems. J. Logic Comput. **8**(3), 401–423 (1998b)

Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. J. Data Semantics **1**, 153–184 (2003)

Bozzato, L., Serafini, L.: Materialization calculus for contexts in the semantic web. In: Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, 23–26 July 2013, pp. 552–572 (2013)

Bozzato, L., Ghidini, C., Serafini, L.: Comparing contextual and flat representations of knowledge: a concrete case about football data. In: Proceedings of the 7th International Conference on Knowledge Capture, K-CAP 2013, Banff, Canada, 23–26 June 2013, pp. 9–16. ACM (2013)

Brewka, G., Roelofsen, F., Serafini, L.: Contextual default reasoning. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6-12 January 2007, pp. 268–273 (2007)

Brockmans, S., Haase, P., Serafini, L., Stuckenschmidt, H.: Formal and conceptual comparison of ontology mapping languages. In: Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.) Modular Ontologies, Lecture Notes in Computer Science, vol. 5445, pp. 267–291. Springer, Heidelberg (2009)

Buvac, S., Mason, I.A.: Propositional logic of context. In: Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, 11–15 July 1993, pp. 412–419. AAAI Press/MIT Press (1993)

Cimatti, A., Serafini, L., Cimatti, R., Serafini, L.: Multi-agent reasoning with belief contexts: the approach and a case study. In: Wooldridge, M., Müller, J. P., Tambe, M. (eds.) Intelligent Agents: Proceedings of 1994 Workshop on Agent Theories, Architectures, and Languages, Lecture Notes in Computer Science, number 890, pp. 71–85. Springer, Heidelberg (1994)

Fisher, M., Ghidini, C.: Executable specifications of resource-bounded agents. Autonom. Agents Multi-Agent Syst. **21**(3), 368–396 (2010)

Ghidini, C.: Modelling (un)bounded beliefs. In: Bouquet, P., Serafini, L., Brézillon, P., Benerecetti, M., Castellani, F. (eds.) Modelling and Using Context – Proceedings of the 2nd International and Interdisciplinary Conference, Context'99. Lecture Notes in Artificial Intelligence, vol. 1688, pp. 145–158. Springer, Heidelberg (1999)

Ghidini, C., Giunchiglia, F.: Local models semantics, or contextual reasoning = locality + compatibility. Artif. Intell. **127**(2), 221–259 (2001)

Ghidini, C., Serafini, L.: Distributed first order logics. In: Gabbay, D., de Rijke, M. (eds.) Frontiers of Combining Systems 2 (Papers presented at FroCoS'98), Studies in Logic and Computation, pp. 121–140. Research Studies Press, Baldock (1998)

Ghidini, C., Serafini, L., Tessaris, S.: On relating heterogeneous elements from different ontologies. In: Kokinov, B., Richardson, D., Roth-Berghofer, T., Vieu, L. (eds.) Proceedings of the Sixth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), Lecture Notes in Artificial Intelligence, Roskilde University, Denmark, vol. 4635, pp. 234–247. Springer (2007)

Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. Artif. Intell. **65**(1), 29–70 (1994)

Homola, M., Serafini, L.: Augmenting subsumption propagation in distributed description logics. Appl. Artif. Intell. **24**(1–2), 39–76 (2010)

Roelofsen, F., Serafini, L.: Complexity of contextual reasoning. In: McGuinness, D.L., Ferguson, G. (eds.) Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, San Jose, California, USA, 25–29 July 2004, pp. 118–123. AAAI Press/MIT Press (2004)

Serafini, L., Bouquet, P.: Comparing formal theories of context in ai. Artif. Intell. **155**(1–2), 41–67 (2004)

Serafini, L., Ghidini, C.: Context based semantics for information integration. In: Bonzon, P., Cavalcanti, M., Nossum, R. (eds.) Formal Aspects of Context, Applied Logic Series, vol. 20, pp. 175–192. Springer Netherlands (2000)

Serafini, L., Roelofsen, F.: Satisfiability for propositional contexts. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, 2–5 June 2004, pp. 369–376 (2004)

Serafini, L., Tamilin, A.: Drago: Distributed reasoning architecture for the semantic web. In: Gómez-Pérez, A., Euzenat. J. (eds.) The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, 29 May -1 June 2005, Proceedings. Lecture Notes in Computer Science, vol. 3532, pp. 361–376. Springer, Heidelberg (2005)

Serafini, L., Tamilin, A.: Instance migration in heterogeneous ontology environments. In: The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, 11–15 November 2007. Lecture Notes in Computer Science, vol. 4825, pp. 452–465. Springer (2007)

Serafini, L., Borgida, A., Tamilin, A.: Aspects of distributed and modular ontology reasoning. In: IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, 30 July–5 August 2005, pp. 570–575 (2005a)

Serafini, L., Stuckenschmidt, H., Wache, H.: A formal investigation of mapping language for terminological knowledge. In: 19th Joint Conference on Artificial Intelligence (IJCAI-05), pp. 576–581 (2005b)