NeOn-project.org

NeOn: Lifecycle Support for Networked Ontologies
Integrated Project (IST-2005-027595)
Priority: IST-2004-2.4.7 – “Semantic-based knowledge and content systems”

D3.1.1 Context Languages - State of the Art

Deliverable Co-ordinator: Peter Haase, Pascal Hitzler, Sebastian Rudolph, Guilin Qi
Deliverable Co-ordinating Institution: Institute AIFB, University of Karlsruhe

Other Authors: Marko Grobelnik, Igor Mozetič, Damjan Bojadžiev (JSI), Jerome Euzenat (INRIA), Mathieu d’Aquin (OU), Aldo Gangemi and Carola Catenacci (CNR)

Abstract.
In this deliverable we present a state-of-the-art survey for context representation languages. We organize the survey along two dimensions: (1) Usage scenarios of context information and (2) approaches for representing and reasoning with context. We compare the different formalisms based on an abstract and generic mathematical definition.
Keyword list: ontology management, networked ontologies, context

Document Identifier:	NEON/2006/D3.1.1V1.0	Date due:	August 31, 2006
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	August 31, 2006
Project start date:	March 1, 2006	Version:	V1.0
Project duration:	4 years	State:	final
		Distribution:	public

NeOn Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2005-027595.

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor E-mail address: m.dzbor@open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB D-76128 Karlsruhe Germany Contact person: Rudi Studer E-mail: studer@aifb.uni-karlsruhe.de
Universidad Politécnica di Madrid (UPM) **Campus de Montegancedo sn** 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.ump.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Piazza Domenicani 3 39100 Bolzano Italy Contact person: Richard Benjamins E-mail: rbenjamins@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 665 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: jerome.euzenat@inrialpes.fr	**University of Sheffield** Department of Computer Science Regent Court, 211 Portobello St. Sheffield S1 4DP UK Tel: +44 114 222 1891, Fax: +44 114 222 1810 Contact person: Hamish Cunningham E-mail: hamish@dcs.shef.ac.uk
Universität Kolenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Marino della Battaglia 44 - 00185 Roma-Lazio Italy Contact person: Aldo Gangemi E-mail: aldo.gangemi@istc.cnr.it
Ontoprise GmbH Amalienbadstr. 36 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Asociación Española de Comercio Electrónico (AECE)** Avenida Diagonal, 437 08036 Barcelona, Spain Contact person: Gloria Tort E-mail address: gtort@fecemd.org
Food and Agriculture Organization of the UN (FAO) Viale delle Terme di Caracalla 00100 Rome, Italy Contact person: Johannes Keizer E-mail address: johannes.keizer@fao.org	**Atos Origin S.A (ATOS)** Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente E-mail address: tomas.parientelobo@atosorigin.com

Changes

Version	Date	Author	Changes
0.1	10.05.06	Peter Haase	creation
0.2	22.05.06	Sebastian Rudolph	major changes: filled in: 3.1, 4.2, 4.3, 4.5 several minor changes
0.3	02.07.06	Peter Haase, Pascal Hitzler	description of use cases, approaches
0.4	02.07.06	Sebastian Rudolph	C-OWL approach, approximate reasoning approach, revisited formal definition
0.5	24.07.06	Sebastian Rudolph	compiling Mathieu's input, bibliography, minor changes
0.6	10.08.06	Igor Mozetic, Marko Grobelnik, Damjan Bojadziev	JSI contributions, bibliography, minor changes
0.7	11.08.06	Peter Haase, Guilin Qi	completion of use cases, approaches; conclusion, exec summary
0.8	17.08.06	Peter Haase	integration of comments from partners
1.0	07.09.06	Peter Haase	integration of comments from internal review, final version

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

- University of Karlsruhe, AIFB (UKARL)
- Institut 'Jozef Stefan' (JSI)
- Open University (OU)
- Consiglio Nazionale delle Ricerche (CNR)

Executive Summary

The notion of context has a very long history within several research communities, leading to vast studies about how to define a context, how to take into account information coming from the context, how to contextualize or de-contextualize knowledge and information, etc. Despite all these studies, a clear and unifying definition of the notion of context is still missing. The goal of this deliverable is to provide a state-of-the-art overview of context representation formalisms in order to clarify the required features of a context model for networked ontologies. We first provide an abstract and generic mathematical definition of context. We then identify several possible usages of context for ontologies. These different usages are important for the evaluation of the approaches to representing and reasoning with context. Finally, we give an overview of some present approaches for representing and reasoning with context which may be relevant for NeOn. We compare the different approaches along several dimensions which are directly derived from our generic context definition.

In terms of the usages of context, we find that *supporting viewpoints and perspectives* and *dealing with inconsistent, uncertain and vague information*, will play a paramount role in NeOn. To be able to address these usage scenarios for context, we believe that the following approaches for contexts are relevant for NeOn: The *networked ontology model* developed in WP1 will provide the most obvious form of context: Ontologies will be embedded in a network of ontologies, which forms the context for its interpretation. *Reasoning with inconsistent ontologies* exploiting context information will be important when different information sources with contradicting information will be integrated. *Context-based selection functions* appear promising for addressing a number of different problems. Finally, a combination of *possibilistic* and *probabilistic logics* seems to be required to deal with the various forms of vagueness and uncertainty in a contextualized way. These findings will be considered in the forthcoming deliverables D3.1.2, where we will define the NeOn formalism for context representation, and in the accompanying deliverable D3.2.1, where we will develop a prototype for reasoning with contexts based on that formalism.

Contents

List of Figures

Chapter 1

Introduction

1.1 The NeOn Big Picture

Real life ontologies and corresponding data are produced by individuals or groups in certain settings for specific purposes. Because of this, they can almost never be considered as something absolute in their semantics and are often inconsistent with ontologies created by other parties under other circumstances. In order to fully utilize networked ontologies, those disagreements must be identified prior to using them for reasoning. Each ontology can be viewed as valid (or appropriate) in a certain context. The context can be seen as a set of all circumstances, properties and facts within which the ontology has the desired semantics. From the theoretical side, we could say that whenever the contextual information is necessary, the target ontology cannot have fully defined static semantics because it depends on some external information which we call *context*. We could call such ontologies *parametric ontologies* because their semantics depends on the value of contextual *parameters*. In this deliverable we provide a state-of-the-art-overview of context languages relevant for NeOn. This report is part of the work performed in WP3 on dealing with context. As shown in Figure 1.1, this work belongs to the central part of the research and development WPs in NeOn. One of the key points of this workpackage is to model and provide a formalization of the context in which a so called parametric ontology is valid. This model will support both a proper representation of the information particular to the context and its formalization that allows reasoning with the modeled context. The context representation formalism to be developed as part of WP3 has close relations with various activities performed in other workpackages, as we will detail in Section 1.3.

1.2 Ontologies and Context

The notion of context has a very long history within several research communities, leading to vast studies about how to define a context, how to take into account information

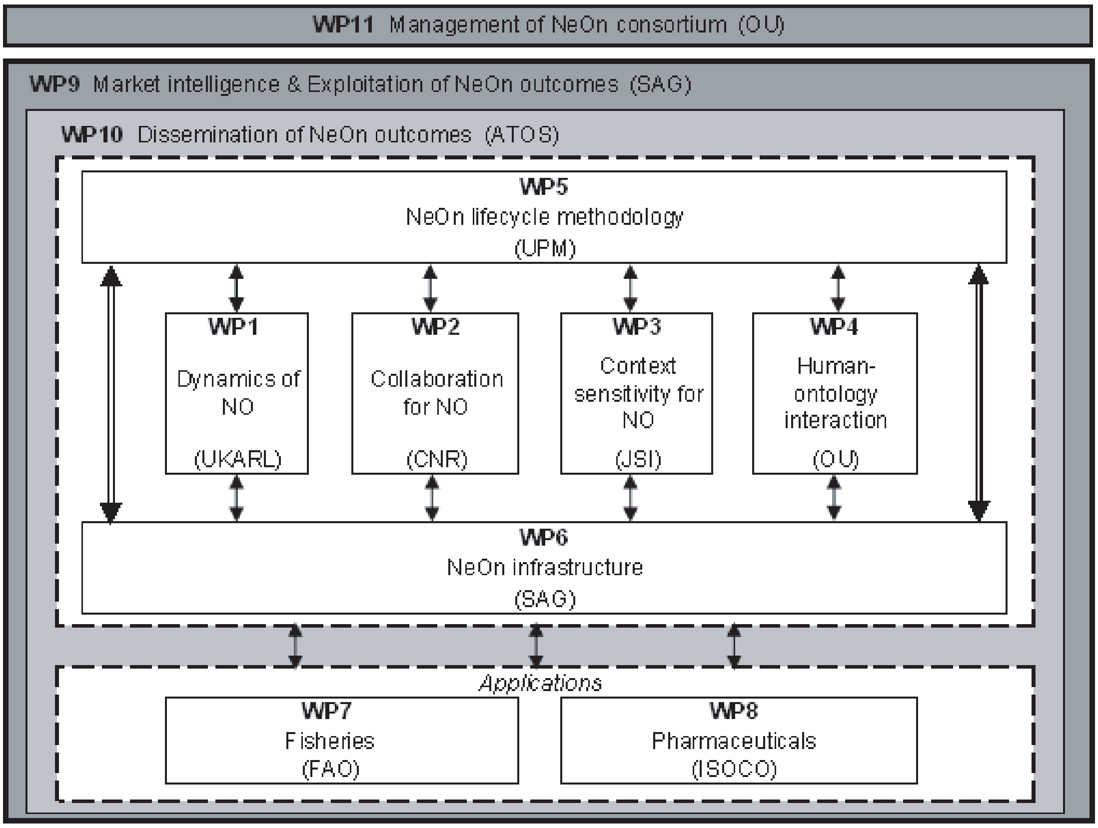


Figure 1.1: The NeOn Big Picture

coming from the context, how to contextualize or de-contextualize knowledge and information, etc. Despite all these studies, a clear and unifying definition of the notion of context is still missing (the page dedicated to the word *context* in Wikipedia[1] contains about ten different definitions, depending on the considered discipline).

Probably the most simple and concrete definition of context comes from linguistics. Indeed, WordNet defines a linguistic context as the "discourse that surrounds a language unit and helps to determine its interpretation".[2] In a more general perspective, a context can be considered as the information, facts, or assumptions, without which a particular situation, piece of information or word cannot be correctly understood. For example, [Guh95] considers that a sentence in a knowledge base is context dependent if its meaningfulness and its truth rely on some assumptions. Making explicit the context dependencies, i.e. reifying the context, corresponds to making explicit these assumptions. Taking as an example the sentence "it is crucial to take into account variables from the environment", it seems obvious that it should be *placed into its context* to be correctly interpreted. It would have very different meanings in the context of software development

---

[1] http://en.wikipedia.org/wiki/Context

[2] http://wordnet.princeton.edu/perl/webwn?s=context

than in the context of agriculture planning. Similarly, different contexts can be situations in which a word can have different definitions. For example, the word "lecturer" is interpreted differently in different countries[3].

Context-aware computing [Dey01b] defines: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." Judging by the many definitions of context in different disciplines, the notion of context is itself context-sensitive, and it is hard to point out the specific characteristic that distinguishes context from background, background knowledge and/or the multiplicity of implicit facts and assumptions that is simply taken for granted, unnoticed, left out, or suppressed as too obvious to mention. This is reflected in the reluctance in some important papers on context to actually define it, such as McCarthy's [McC86], and in his insistence that "there is no universal context".

The goal of this deliverable is not to establish a universal definition or to discuss this variety of definitions[4] but rather to focus on the question of how we can deal with *context for ontologies*. Even in connection with ontologies, there is also some context-dependence in the definition of context: ontologies themselves supply context, e.g. for browsing (which again indicates that context can be practically anything), but mappings between ontologies supply context too, as in C-OWL [BGH$^+$03].

To survey the field of context of ontologies, we proceed along two dimensions:

1. What are *usage scenarios* of context for ontologies?
2. What *approaches* for the representation of and reasoning with context exist?

Finally, we establish the relationship between the two dimensions, i.e. we investigate which approaches are applicable for which forms of usage. Usages of context are manifold and include for example

- supporting different viewpoints, where the viewpoints may represent mutually inconsistent perspectives, but each of them may be valid given a particular context,
- personalization of ontologies, where the context is provided by a user profile,
- dealing with imperfect information, e.g. inconsistent, uncertain and vague information, where context can be exploited to allow useful interpretations despite the presence of imperfection.

Similarly, there is a wide range of approaches to representing and reasoning with contexts:

- In some approaches such as *context logics* we find explicit support for talking about contexts as first class citizens of the language.

---

[3] http://en.wikipedia.org/wiki/Lecturer

[4] there exist several surveys addressing these issues [BBG01, AS96, BGGB03]

- In other approaches, context is implicitly provided e.g. via the relationship of an ontology with other elements of a domain.
- Yet other approaches rely on extending ontology languages with the ability to represent contextual information about ontologies and their elements in the form of numerical values indicating probabilities or possibilities of particular statements.

To study the existing concrete representations, we start from an abstract and generic definition of context for ontologies: We consider a context as a *modifier of semantics*, i.e., we define context via its function. If context for some information is provided, this will alter the information's meaning, i.e. its semantics. We then instantiate this generic definition with the concrete representations of contexts in order to better understand how different usages of context lead to different specializations of this definition, and so, which kind(s) of concrete definition may be central for NeOn.

1.3 Relationship With Other Workpackages

The context representation formalism to be developed as part of WP3 has close relations with various activities performed in other workpackages. We here discuss the relationships with the three other workpackages, that – together with WP3 – form the four basic dimensions of the NeOn project.

WP1 - Dynamics of Networked Ontologies One of the main goals of WP1 is the definition of the networked ontology model. The networked ontology model is closely related to the notion of context developed as part of WP3. Firstly, the networked ontology model needs to be able to represent context as defined in WP3. Secondly, the network of ontologies that a particular ontology is embedded in can itself be a special form of context: In the networked scenarios of NeOn, ontologies are not treated as isolated entities, but are related to other ontologies in various networked ways, including versioning and mapping information etc. These other ontologies together with these links can be understood as a context for the ontology, as they will (in some cases) alter the knowledge which can be inferred from the ontology. The exact semantics of the networked ontology model will be provided by WP1.

WP2 - Collaboration From the collaborative perspective of WP2, the main objective is to characterize and to apply design rationales for networked ontologies within knowledge-creating communities. Here, we may want to take into account also the social environment in which a knowledge object (database or ontology) is created or used, i.e. the social context of the ontology. Knowledge objects and their elements can be seen as elements of larger entities, together with the agents that have made (or use) them, the organizations or institutions to which these agents belong, the roles established within these organizations and the activities the agents perform. These larger entities can be modeled in terms of *knowledge collectives* (or communities) and of the projects in which the collectives are involved. Typically, knowledge collectives share *views* or *conceptualizations* of the domain they deal with. These views are *contexts* that specify the sense in which a concept is used by a given knowledge collective, and they are internal to the knowledge production process. In the DOLCE-DnS ontology, which will be used in WP2 to deal with the collaborative aspects of networked ontologies, concept definitions, called descriptions, are introduced, which correspond to this sense of ’context’ [MVB$^+$04]. In DOLCE-DnS, all social and knowledge entities are put in the domain of discourse via *reification*: social concepts and roles, as well as their descriptions. This allows to formally characterize in a first-order theory the relationships among all these entities, and to define properties ranging over them.

WP4 - Human Ontology Interaction In a networked scenario of ontologies with potentially thousands of classes and relations, a capability to not only visualize but intelligently select and filter the right level of detail to be shown and interacted with is greatly important. The level of detail that is most appropriate for a given situation depends on (i) userŠs profile and group-wide preferences, (ii) constraints of the user interface; e.g. a display size, device mobility, bandwidth restrictions, or (iii) access/re-use rights a specific user has for an ontology s/he wishes to link, etc. From an HCI point of view what is needed is the ability to customize user interfaces for working with multiple ontologies simultaneously, and allow users to perform actions that are relevant to a particular context. Here, WP4 will rely on the notion of context developed in WP3 to improve the interaction with networked ontologies.

1.4 Overview of the Deliverable

This deliverable is structured as follows. In Chapter 2, we provide a generic and abstract formalization of context for ontologies to cover a range of different notions of context. The purpose of this definition is to provide a basis for a characterization and comparison of different context representation formalisms within a common framework by instantiating the generic definition. The usage of context information for ontologies may be manifold. In Chapter 3, we discuss different usages of context that may be relevant in the scope of NeOn. In Chapter 4, we analyze existing approaches to the representation of and reasoning with context and compare these approaches by describing them in terms of our generic definition. Further, we describe how these approaches can be applied to realize the different usages of context. We conclude with a summary and a roadmap for future work in Chapter 5.

Chapter 2

A Generic Definition for Context

In this chapter we provide a formal definition of *context of an ontology*[1]. This definition does not attempt to give a notion of "context" in a more general sense. It is also not intended to be operationalized in a reasoner. The purpose of this definition rather is to provide a basis for a characterization and comparison of different context representation formalisms within a common framework by instantiating the generic definition.

Contexts as modifiers of semantics. We are interested in knowledge expressed as a set of assertions and rules. Examples are knowledge bases (or ontologies), and relations between such knowledge bases.

If such a set of assertions is put into a context, then this means that the context alters some of the meaning of the set of assertions. In other words, the context acts as a *modifier* for the semantics of a knowledge base.

2.1 Formal abstract definition of context for an ontology

Let K be a knowledge base, which comes with an associated semantics $S(K)$. Thus, S is a function which associates a semantics to any knowledge base K.

Now, given a context C and a knowledge base K, we denote by $S'(K,C)$ the semantics of K *in the context* C. Thus, S' is a function which associates to any knowledge base K and context C a semantics, e.g. expressed by the set of all logical consequences of K in the context C.

If we have empty context (denoted by $\emptyset$), then often we require $S'(K,\emptyset) = S(K)$.

Note that there is a convenient way to describe the function S' in many cases. Given a knowledge base K and context C, it will often be possible to create a knowledge base K' such that $S'(K,C) = S(K')$. In these cases, reasoning within a context can be

---

[1] Please note that within this deliverable we do not distinguish between the use of the notion of "ontology" and that of "knowledge base". However, for the logical characterization, we tend to prefer the term "knowledge base".

reduced to changes of the knowledge base K (converting it into K'), and by reusing existing reasoners.

Formal Definition of Context We will now go into further detail. Taking some language L, a *knowledge base on* L is a (possibly infinite) set of statements over L. Let $\mathcal{KB}(L)$ denote the set of all knowledge bases expressible in L.

Now, we consider a language L_k called *knowledge language*. A semantics for L_k can be formalized as a function $S : \mathcal{KB}(L_\mathrm{k}) \to \mathcal{KB}(L_\mathrm{k})$ assigning to a knowledge base K a knowledge base $S(K)$ containing all logical consequences of K expressible in the knowledge language.

Let furthermore be L_c a language called *context language* for expressing contextual knowledge. An L_c-context semantics for L_k is then a function $S : \mathcal{KB}(L_\mathrm{k}) \times \mathcal{KB}(L_\mathrm{c}) \to \mathcal{KB}(L_\mathrm{k})$. (The overloading of the symbol S is by purpose.)

In practice, one will mostly impose further restrictions on the knowledge base one works with. E.g., a knowledge base could be required to contain only certain kinds of expressions from L_k – as an easy example, take a database containing only tuples of entities (or, similarly, a logic program containing only ground facts) while the entailed knowledge (respectively the expressible queries) could have a much more complex structure. Another common constraint to knowledge bases is that they have to be finite (or at least finitely representable in some sense). The set of finite knowledge bases over some language L_k will be denoted by $\mathcal{KB}_\mathrm{fin}(L_\mathrm{k})$.

In many cases, additional constraints will be reasonable. In particular, we will call a context semantics

- *conservative*, if $S(L_\mathrm{k}, \emptyset) = S(L_\mathrm{k})$ for all $K \in \mathcal{KB}(L_\mathrm{k})$. This means that, if an empty context (i.e. no contextual information) is provided, the semantics coincides with the “pure” semantics of the knowledge language.

- *extensive*, if $K \subseteq S(K, C)$ for all $K \in \mathcal{KB}(L_\mathrm{k})$, i.e., all statements of the knowledge base are as well logical consequences of it. In other words, any information stated in the knowledge base can be deduced to be valid (and cannot be spoilt by whatever context provided).

- *knowledge-monotone*, if $K_1 \subseteq K_2$ implies $S(K_1, C) \subseteq S(K_2, C)$ for all $K_1, K_2 \in \mathcal{KB}(L_\mathrm{k})$ and $C \in \mathcal{KB}(L_\mathrm{c})$, i.e., all logical consequences remain valid if the knowledge base is augmented and the context does not change. Note, that this is not always the case (cf. non-monotonic semantics by closed world assumption).

- *context-monotone*, if $C_1 \subseteq C_2$ implies $S(K, C_1) \subseteq S(K, C_2)$ for all $C_1, C_2 \in \mathcal{KB}(L_\mathrm{c})$ and $K \in \mathcal{KB}(L_\mathrm{k})$, i.e., if the information given by the context increases, the derivable information does so as well. In particular, no previously valid consequence can be invalidated by adding more contextual knowledge.

- *idempotent*, if $S(S(K, C), C) = S(K, C)$ for all $C \in \mathcal{KB}(L_\mathrm{c})$ and $K \in \mathcal{KB}(L_\mathrm{k})$, i.e., taking all consequences of a knowledge base under a certain context and then taking again all consequences under the same context will yield nothing new.
- *dependently reducible*, if there is a function $\sigma : \mathcal{KB}(L_\mathrm{k}) \times \mathcal{KB}(L_\mathrm{c}) \to \mathcal{KB}(L_\mathrm{k})$, such that $S(K, C) = S(\sigma(K, C), \emptyset)$, i.e., knowing a knowledge base K and a context C, one can determine a new finite knowledge base with the same set of consequences as K with context C. I.e. for every contextualized knowledge base we can determine a logically equivalent knowledge base without context.
- *independently reducible*, if there is a function $\tau : \mathcal{KB}(L_\mathrm{c}) \to \mathcal{KB}(L_\mathrm{k})$ such that $S(K, C) = S(K \cup \tau(C), \emptyset)$ for all $K \in \mathcal{KB}(L_\mathrm{k})$ and $C \in \mathcal{KB}(L_\mathrm{c})$, i.e., any context can be "translated" into L_k (independently from K) and simply added to the knowledge base. In this case, contextual reasoning could be reduced to pure reasoning over L_k, such that existing methods could easily be employed for this.

The above definition is very abstract. This is done on purpose to accommodate the many practically important ways of context usage. In Chapter 4, we give some examples of concrete instances of the abstract definition. Many more notions of context fit our general definition. In the project, we will have to determine which concrete instances will be used and supported by the NeOn system. These instances will have to be dealt with on an individual basis when realizing the NeOn system.

Chapter 3

Usages of Context

The usage scenarios of context information for ontologies are manifold. In this Chapter we provide an overview of possible usages that may be relevant in the scope of the NeOn project.

3.1 Supporting different Viewpoints and Perspectives

In practice the use of ontologies for different tasks and purposes requires to consider the particular task as context for the ontology. The reason is that ontologies are often not really designed independent of the task at hand [Stu06]. In general, the context of use has an impact on the way concepts are interpreted to support certain functionalities. As some aspects of a domain are important in one context but do not matter in another one, an un-contextualized ontology does not necessarily represent the features needed for a particular use. In order to solve this problem, we have to find ways to enable the representation of different viewpoints that better reflect the actual needs of the application at hand.

When talking about viewpoints, we can distinguish two basic use cases: In the first case, the aim is providing means for maintaining and integrating different existing viewpoints. In the second use case, one may want to extract a certain viewpoint from an existing model that best fits the requirements of an application.

In many application domains (such as medicine [SvHB$^+$04]) it is acknowledged that the creation of a single universal ontology is neither possible nor beneficial, because different tasks and viewpoints require different, often incompatible conceptual choices. As a result, we need to support situations where different parties commit to different viewpoints that cannot be integrated by imposing a global ontology. This situation demands for a weak notion of integration, in order to be able to exchange information between the viewpoints. [Sto06] describes one such example from oncology: Oncology is a complex domain where several specialties, e.g. chemotherapy, surgery, and radiotherapy are involved in a sequence of treatment phases, each representing a particular viewpoint. A decision taken in a local viewpoint, i.e. for a particular oncology specialty, may have an influence on the decision to be taken in another local viewpoint.

An approach to extracting different viewpoints from a corpus of documents is presented in [FGM06]. The authors introduce a word weighting schema as an alternative to the standard TFIDF weighting (Term Frequency / Inverse Document Frequency). The weighting schema is automatically learned from the background knowledge provided by the user which corresponds to users' different views on some documents. Then, different taxonomic ontologies are constructed, each corresponding to a different viewpoint on the same corpus of data.

3.2 Dealing with Temporal Information

One aspect of implicit, contextual information is its temporal component. The ability to identify, represent and reason about time-dependent information is important for various applications, such as databases, planning, scheduling, natural language processing, news streams analysis, and others.

Some natural language applications where temporal information is relevant are information extraction, question answering, and multi-document summarization. In [MCH05], the performance of question answering is considerably improved by detecting temporally related events in text and converting them into an enriched logical representation.

Another example where temporal reasoning can be used is in news analysis to distinguish related news items from unrelated ones. In this case, explicating the temporal component of context in a formal model makes it possible to disambiguate some context-dependent events and discover connections between them [MB06].

A central theme of the Semantic Web is the aggregation of data from different sources. With respect to temporal information one is often faced with the problem of implicit time [GMF04]. Sites often publish a piece of data that is true at the time of publication, with the temporal qualification left implicit. Equally often, this data does not get updated when it no longer holds (e.g., some sites still list Bill Clinton as the President and Yugoslavia as a country). Even worse, such implicitly temporally qualified data is often mixed with data that is not temporally qualified. When aggregating data from these sites, one has to either make the time explicit or only selectively import those facts that are not likely to have changed.

The semantic heterogeneity problem on the Web is further complicated when the semantics of data not only differs across sources, but also changes over time. [ZMS04] introduce the notion of temporal context as a formalization of the problem. They represent temporal context as a multi-valued method in F-Logic, and treat temporal relations as constraints in an abductive constraint logic programming framework.

3.3 Dealing with Inconsistent Information

In the distributed NeOn scenario, inconsistency occurs naturally even if each source of information does not contain contradictory information. Inconsistency can occur due to several reasons, such as modeling errors, migration or merging ontologies, and ontology evolution. For example, information from different sources may need to be merged into a single knowledge base. As each of the source may have been created independently, it is reasonable to expect that the merged knowledge base may contain contradictory information. Current DL reasoners can detect logical inconsistency. However, they only provide lists of unsatisfiable classes. The process of *resolving* inconsistency is left to the user or ontology engineers. The need to improve DL reasoners to reason with inconsistency is becoming urgent to make them more applicable [HvHtT05].

Contextual information can be used to resolve such conflicts. It can be used to select relevant consistent parts of the knowledge base which suffice for the task at hand. Contextual information provides guidance for this selection process, as usually different possibilities exist for resolving an inconsistency.

Consider e.g. two knowledge bases, one dealing with livestock data, and one dealing with animal trade laws. As both knowledge bases may be in constant use, they will be changing rapidly and dynamically, which makes a manual alignment unfeasible. Automated tools for aligning respectively merging the two knowledge bases are prone to certain kinds of mistakes. In this case, a wrong alignment may e.g. be the result of different usages or meanings of the word (i.e. class name) *breed* in the two knowledge bases. Consequently, the merged knowledge base may contain inconsistencies.

Now, an application concerning mainly zoological aspects could use provenance or other information and successively remove parts of the merged knowledge bases which stem from the animal trade laws knowledge base, as in the case of conflicting information the livestock data is more likely to contain the information needed. The context, in this case, is given by the usage of the data, and the merged knowledge base is modified accordingly.

3.4 Dealing with Uncertain or Vague Information

Information is often pervaded with uncertainty. There are many different types of uncertainty [AM97]: knowledge is partial, beliefs are not fully reliable, the representation language is inherently imprecise and information from multiple sources is conflicting, etc. Often the uncertainty of information is related to the fact that it is only valid under certain assumptions, circumstances – or in other words, in particular contexts. If that information is preserved, it can later be used in reasoning.

In addition to uncertainty, information often comes with a certain level of vagueness. Something is said to be vague if its definition, its boundaries, cannot be clearly established. Classical examples are concepts like tall, big or adult, for which it is not clear where they begin and where they stop. Here, contextual information about the information can be used for a more precise interpretation.

3.5 Personalization / User Profiling

User profiling is a research and application area accompanying many other fields of research which provide their results through some kind of user interface. The function of user profiles is to optimize one or several tasks a user performs with the system.

Good examples of such task optimization are advanced e-shops which use information from pre-calculated user profiles showing to the user products which are more likely to be bought. The function being optimized in such a scenario is the profit of the e-shop. Another example would be personalized news delivery or personalized search where by using user profiles we increase the quality of user's experience when browsing the retrieved information.

In both of the above examples we deal with functions which are to be optimized (expressed implicitly or explicitly) and with a model (always expressed explicitly) which determines ranking of retrieved information items (e.g. search results, new items or products). We call such a model a user profile. A common scenario is that such a model is constructed and updated explicitly by manual intervention or implicitly by a background process which observes and summarizes user's behavior. In both cases the result is a model which allows to analytically determine the user's point of view and captures his interests. To be broader, we could also say that different users have not only different point of views but also attach different semantics to the same terms used to describe information needs. A good example would be the word 'Madonna' which has different default semantics for different people.

Having in mind the above description we can say that a user profile is a model which compensates differences in understanding of the same information - in other words, if we define the context as a semantic modifier function, we can say a user profile models a contextual view of a particular user to the common domain. This context can be static or dynamic (perceived semantics of terms can change through time) and can be also dependent on the 'upper level context' or 'situation' in which a user appears - e.g. a user can have different understanding of the word 'Madonna' when he talks to a local priest or when he talks to his buddies in the local pub.

An important question is how to represent the user profile model. The most common representation of the profile is analytical, as a set of weighted application terms/keywords - such a vector is provided by the user or is calculated from the previous user's activities. More complex models could be in the form of taxonomies of term vectors (or other representations) which capture more detailed contextual view of the data. An example of taxonomic user profiles is SEKTbar Internet-Explorer plug-in ([GMG04], see Figure 3.1) which monitors user's activities, builds taxonomy of user interests on the fly and through such a model implements the function of the smart history of user browsing and potentially helps in ranking search results as a meta level above some search engine.

3.6 Situation Awareness in Pervasive Computing

Pervasive computing in concerned with the availability of many connected computing devices in our environment. One of its goal is to take advantage of these devices in order to help users in the more appropriate way. For that purpose, the notion of context in which users are acting is very important: the more devices understand the context in which the user is evolving, the more helpful they can be. We talk about context-awareness or situation-awareness [Dey01a].

Although, several domains have considered the notion of context, the standpoints from which this notion is considered are different: in pervasive computing, the context of an application in terms of its physical parameters has been especially considered; in human-computer communication, the context is most often the user task and the history of its dialogue with the computer [Dou01]. So the notion of context can be a physical situation (as the spatio-temporal location of some person) or functional (as the current task of the person).

In pervasive computing, the physical context is of the outmost importance. In general, it is acquired through sensor data (for instance, the temperature acquired by a thermometer). These data are further elaborated into context characterization adapted to their use (for instance *high temperature* for some clime controller). With regard to the sensor data (a temperature), the information has been weakened (i.e., made less precise) but is more adapted.

So, information such as pressure, temperature, sunlight, humidity and date are context elements. If one wants to implement a temperature service the context will be restricted to temperature, but for a clime monitoring station, information of temperature, sunlight and date will be aggregated for regulating the heat sources.

A pervasive computing application must be able to get in and out of contexts: when the user enters a building, the context related to the city she is in, is over-ruled by that of the building (which will be over ruled by that of the room she is in, etc.). This context must be forgotten upon exit of the building.

The various definitions of context in pervasive computing are very often related to an application or a particular domain. The drawback of this characterization is its reliance on the task : *high temperature* is not an absolute characterization. It depends on the use of the room (a sauna or a sleeping room).

More than context, pervasive computing tends to manipulate a characterization of the context in the perspective of an application. In consequence, it is difficult to dynamically implement non expected applications with the characterization of context made for another one.

However, multi-application context modeling is now understood in pervasive computing. [CCDG05] raises the issue of considering context independently from applications.

Figure 3.2 shows the layered way to progressively elaborate context information from sensors to applications.

For this application-independent context modelling, the use of ontologies is natural because various context information have to be assembled [CCDG05]. In [ERP06], is even proposed a context management system in which ontology matching is used for enabling interoperability of contexts. This helps the perception and situation layers so that they can support the dynamic evolution of the environment (new sensors and applications).

3.7 Scalability

Typical application scenarios addressed by NeOn involve large knowledge bases, and it is of vital importance that corresponding tools scale well. Contextual information can be used to address the scalability issue in different ways. Most importantly, context helps to narrow the domain of interest, and can thus be employed to select relevant parts of the knowledge base while ignoring other parts, thus reducing the size of the knowledge base which has to be taken into consideration. Contextual information can also give usage information, which in turn allows to select visualisation or reasoning methods which are specifically tuned for the task at hand.

As an example, consider a knowledge base containing ontologies about geographical and touristic data in different versions, together with versioning information. When planning the next holidays, contextual information would suggest to ignore both the geographical data and older versions, thus reducing the size of the knowledge base to be taken into consideration. Another usage situation is given when querying the knowledge base while on a holiday trip, e.g. using a mobile phone to retrieve data about a certain point of interest. In this case the geographical data may be relevant, so that it should not be ignored. However, the situation demands that the system responds quickly, even at the expense of preciseness or correctness of the answer. The contextual information would thus invoke heuristic or approximate reasoning algorithms which trade correctness for speed.

3.8 Ontology Adaptation and Views on Ontologies

Originally ontologies have been conceived as a task-neutral description of a certain domain of interest that can be reused for different purposes, such that ontology-based descriptions of information make it possible to use the information for different purposes and in different contexts.

In scenarios of ontology reuse it is however often the case that the particular context of use requires to adapt an existing ontology for the given purpose of use. Thus, depending on the task to be accomplished – i.e. the context in this case – different views on the ontology might be useful. The knowledge of the possible tasks could then itself be represented explicitly in an ontology or it could be implicitly present by principles describing how the domain knowledge can be restricted to that relevant for the considered task.

This notion of exploiting context has also been referred to as "knowledge lens": a selection, filtering and viewing paradigm parameterized in several dimensions in order to provide a conceptual view onto an ontology. The parameterization can be done with regard to:

- Focus setting the stage for what facts are relevant. For instance, in the field of cardiology, one may decide to consider all concepts related to the *heart* to be in the focus of interest.
- Conceptual range defining what set of facts in the conceptual neighborhood of the focus is of interest. For instance, one may parameterize this dimension in order to see only what is conceptually very close to *heart*.
- Resolution determining at which level of abstraction one wants to consider the selected parts.
- Perspective projecting from the multidimensional space of facts onto a simpler subset; for instance one may only be interested in some aspect of the heart such as its functioning.

There has been some work on specifying views on RDF data, mainly inspired by the concept of views in database systems. The idea is to define rules for extracting and possibly restructuring parts of a basic RDF model to better reflect the needs of a certain user or application. Different techniques have been proposed including the use of named queries, the definition of a view in terms of graph traversal operations and the use of integrity constraints for ensuring the consistency of a view. One way of further subdividing related cases could be in analogy to operations of relational algebra (known as a basic DB theory paradigm):

- *Projection.* In many cases, it may be beneficiary to hide (or forget) certain data. As an example, imagine a large ontology maintained by several people. Every entry is endowed with the information which person entered it at which time. For "normal" retrieval tasks, this kind of information would be irrelevant and, moreover, providing public access to it could not be appropriate. Thus it would be reasonable to have a general and a public view, where the latter is a projection of the former. The context here would be the role of the user as a, say, ontology administrator vs. querying customer.
- *Selection.* This would be an alternative way to restrict present knowledge to the parts relevant for a specific task or interest. Suppose, there is an ontology about (extinct and alive) living beings – possibly maintained by the biology department of some university. This ontology is going to be used as a source of information about zoological gardens. Naturally, a potential web interface can be restricted to

alive animals, i.e. we have a constraint on the ontology entry's attributes. Here, the purpose of the intended usage of the data would provide the context for selection.

- *Join.* Here – contrary to the cases before – an integrated view has to be created by combination of separate sources. Obviously, knowledge from different ontologies has to be combined to be capable of providing comprehensive and adequate support for users in many cases. E.g. it would be practical to combine the above mentioned biological ontology with an ontology about geography and climate in order to investigate, whether a certain species can be kept in a certain zoo in an outdoor installation (this could be realized by a "join" of the climate of the animals natural habitat with the zoo's clime).

Obviously, the last kind of view generation necessitates the most sophisticated technologies, since it has to be ensured, that the used ontologies semantically agree on the shared terminology. Thus, elaborated methods for associating ontologies are required. Ideas in this direction have been described with the ontology-composition algebra by Wiederhold [MW04].

3.9 Using contexts for matching ontologies

Matching two (or more) ontologies consists of finding the correspondences (e.g., equivalence, subsumption) between the elements of these ontologies (e.g., concepts, properties, formulas).

This a very important task because it helps restoring interoperability but a difficult one because it is very hard to find these correspondences in the general case: independently built ontologies can vary a lot on the terminology they use and the way they model the same entities.

The problem is that domain ontologies are focussing on their particular domain and use terms in a sense that is relevant to this domain (e.g., *Ontology* in computer science) and which is not related to similar concepts in other domains. In one word, these ontologies are designed in a context which is not explicit.

One way to help this process consists of using a third ontology as the context of the two ontologies to be matched. This can typically be an upper-level ontology (Cyc [LG90], Suggested Upper Merged Ontology (SUMO) [NP01] or Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [GGMO03]) that is used as external source of common knowledge.

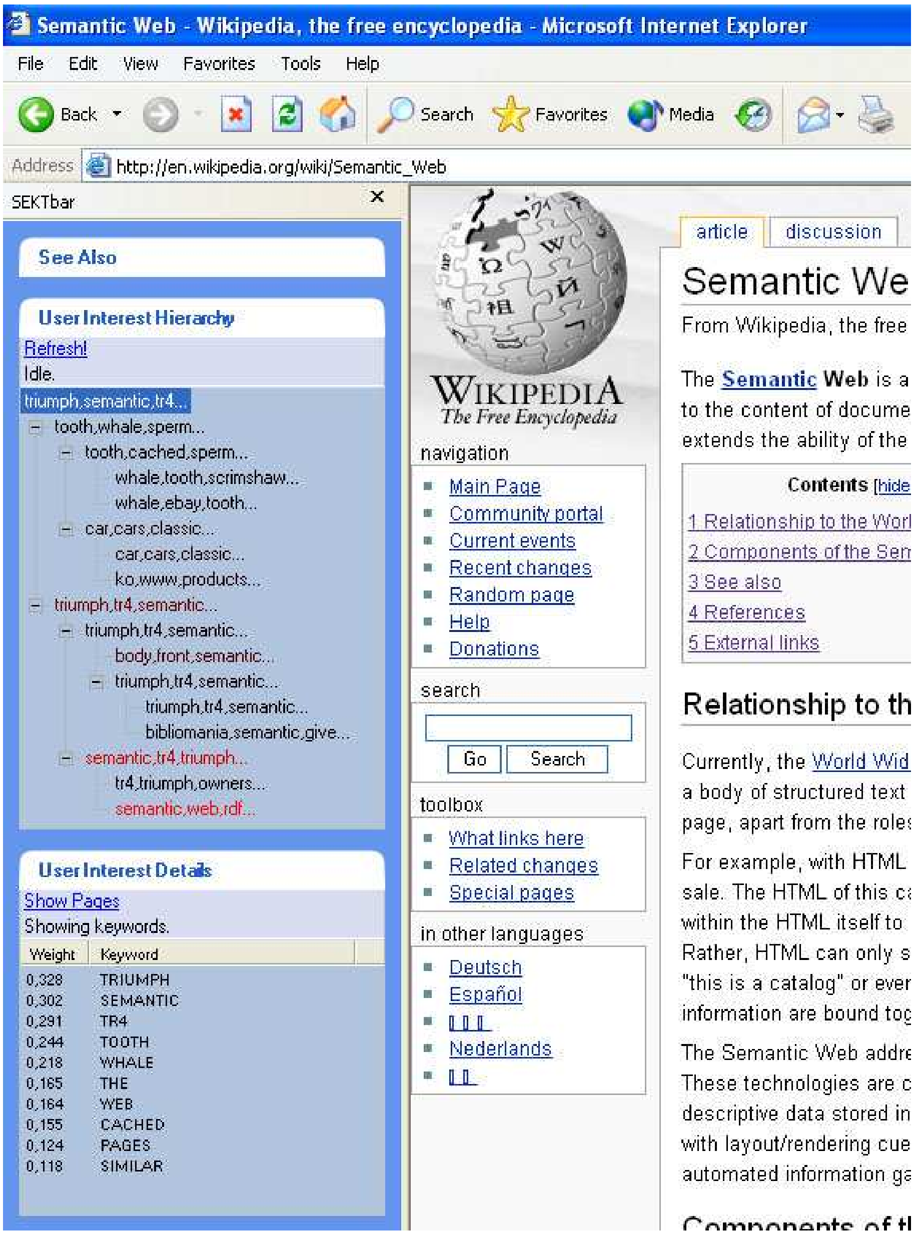
But this can also be made by reference to a specialized ontology that is known to cover the ontologies to match. This particular ontology can be used for providing the missing structure to poorly structured resources [AKtKvH06].

Matching each of the ontologies to a common third ontology put these two ontologies in the same context. This should be easier than the direct matching because: (i) upper level ontologies having a broad coverage are more prone to provide matching super concepts of these ontologies, (ii) there are more chances that these alignments already exists due to the availability of these ontologies.

Once the alignments between the two ontologies and the upper-level one are obtained, it is easier to partition the ontology matching task in smaller matching problems because the matching will identify the concepts having common super-concepts and can take advantage of the exclusion assertions typically found in the upper-level ontologies.

It is also possible to compose the correspondences with the such as if a Novel in one ontology is identified as less general than a Book in the context ontology which itself is a sub-concept of Product which is identified with Good in the second ontology. Then, one can conclude that Novel is less general than Good.

This involves reasoning about the ontologies.



Idle.

triumph,semantic,tr4...
- tooth,whale,sperm...
  - tooth,cached,sperm...
    - whale,tooth,scrimshaw...
    - whale,ebay,tooth...
  - car,cars,classic...
    - car,cars,classic...
    - ko,www,products...
- triumph,tr4,semantic...
  - triumph,tr4,semantic...
    - body,front,semantic...
    - triumph,tr4,semantic...
      - triumph,tr4,semantic...
      - bibliomania,semantic,give...
  - semantic,tr4,triumph...
    - tr4,triumph,owners...
    - semantic,web,rdf...

Showing keywords.

Weight	Keyword
0,328	TRIUMPH
0,302	SEMANTIC
0,291	TR4
0,244	TOOTH
0,218	WHALE
0,165	THE
0,164	WEB
0,155	CACHED
0,124	PAGES
0,118	SIMILAR

Semantic We

From Wikipedia, the free

The **Semantic Web** is a
to the content of docume
extends the ability of the

Relationship to th

Currently, the World Wid
a body of structured text
page, apart from the roles

For example, with HTML
sale. The HTML of this ca
within the HTML itself to
Rather, HTML can only s
"this is a catalog" or ever
information are bound tog

The Semantic Web addre
These technologies are c
descriptive data stored in
with layout/rendering cue
automated information ga

Figure 3.1: The screenshot shows the topic ontology of the user's interests and the most characteristic keywords from the root cluster. The user's most recent interest is highlighted with red color (the brighter the more relevant).

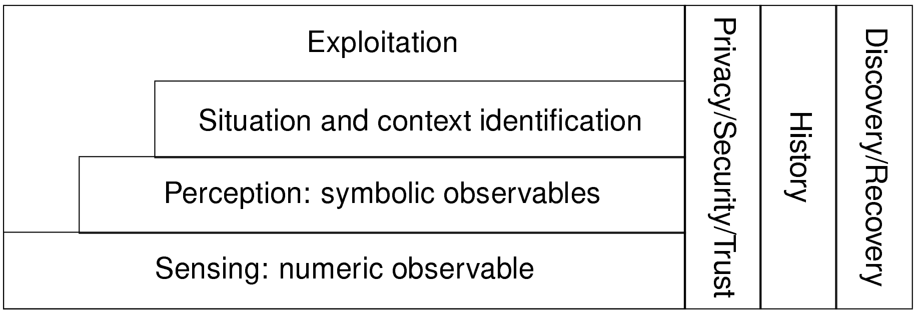


Figure 3.2: Model for context in pervasive computing. Data coming from sensors are aggregated and elaborated into the context used by applications (from [CCDG05]).

Chapter 4

Approaches to Representing and Reasoning with Context

4.1 General Dimensions

The following list provides dimensions that should be used to characterize and compare the approaches. It is directly derived from our generic context definition.

- **Knowledge Language**: What is the language that is used to encode the knowledge for which the context is provided? In other words, what are the structures that are put into context?
- **Knowledge Semantics**: How is the semantics formalized for the knowledge?
- **Context Language**: What is the language that is used to encode the context? What are the structures that define the context?
- **Context Semantics**: How is the semantics used to modify the semantics of the knowledge?

In addition, the following characteristics can be discussed.

- *Scalability*: An important aspect is the scalability of dealing with context. Here we are faced with the classical tradeoff between expressivity vs. complexity. If known, we provide complexity results for the relevant reasoning tasks.
- *Acquisition*: How can the context be acquired? Is it modeled manually, or do there exist semi-automated or automated methods for the acquisition of context?

4.2 Networked Ontology Models

4.2.1 Overview

In the networked scenarios of NeOn, ontologies are not treated as isolated entities, but are related to other ontologies in various networked ways, including versioning and mapping information etc. These other ontologies together with these links can be understood as a context for the ontology, as they will (in some cases) alter the knowledge which can be inferred from the ontology. The exact semantics of the networked ontology model will be provided by WP1. We here exemplarily describe one particular relation in a network of ontologies, namely $\mathcal{E}$-connections to establish links between individual ontology modules.

$\mathcal{E}$-connections (first described in [KLWZ03] and thoroughly treated in [KLWZ04]) evolved out of the motivation to combine different description logic formalisms – thereby exploiting their respective benefits – in a way that preserves decidability. Here, we give just a very brief overview over the basic notions and results.

By introducing *abstract description languages*, a common framework for concrete description logics is provided. Interpretations are defined in the canonical way known from description logics. Then, for two given description logic systems $\mathcal{S}_1$ and $\mathcal{S}_2$ and a non empty set $\mathcal{E} = \{E_j \mid j \in J\}$ of binary relation symbols, the resulting $\mathcal{E}$-connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ is a new description logic the terms of which (distinguished into 1-terms and 2-terms) are inductively defined as follows (let $i \in \{1, 2\}$):

- all atomic concepts of S_i are i-terms,
- the set of i-terms is closed under the constructors of the description logic system $\mathcal{S}_i$, and
- for every 1-term t, the expression $\langle E_j \rangle^2 t$ is a 2-term and for every 2-term t, the expression $\langle E_j \rangle^1 t$ is a 1-term.

Consequently, an interpretation for $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$ consists of an interpretation for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, as well as a binary relation between the corresponding domains for every E_j. An i-term is interpreted extensionally in the $\mathcal{S}_i$-domain. In particular, $\langle E_j \rangle^1 t$ is interpreted by those entities of the $\mathcal{S}_1$-domain having an E_j-neighbor in the $\mathcal{S}_2$-domain that belongs to the concept t. All other concept constructors are defined canonically.

Intuitively, $\mathcal{E}$-connections allow to "switch" between separate DL-interpretations along established links when building concept descriptions.

The main result about $\mathcal{E}$-connections is the following: if TBox reasoning for both $\mathcal{S}_1$ and $\mathcal{S}_2$ is decidable then so it is for $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$. This allows to construct so-called *fusions* of description logics for special purposes. Note that fusing a, say, decidable DL for describing static ontological knowledge with a temporal description logic could be a way to incorporate dynamic context into an ontology (cf. also Section 4.5).

In fact, $\mathcal{E}$-connections capture a lot of expressivity from distributed description logics (see [SSW05] for a comparison), which will be described in Section 4.3.

4.2.2 Characterization Along Our General Dimensions

If, for example, $\mathcal{E}$-connections are used for establishing relations between two ontologies, then a context C for an ontology (or knowledge base) K could consist of a set of other ontologies which are linked to K by $\mathcal{E}$-connections. The resulting semantics $S'(K, C)$ of K is described by means of the $\mathcal{E}$-connections semantics.

4.2.3 Usage Scenarios

Supporting Viewpoints and Perspectives Networked ontologies are suitable for supporting viewpoints and perspectives. Here, a single node (or a set of nodes) in the network of ontologies is used for the representation of a particular viewpoint. This viewpoint can then be related with other viewpoints via mapping or bridging relations, as described above for the case of $\mathcal{E}$-connections.

Dealing with Temporal Information In a similar way as for the representation of viewpoints, networked ontologies can also be used to represent different versions of a model. The network of nodes then represents a version space, which can be considered as a temporal context. It is then possible to rely on techniques from temporal logics to reason over such version spaces, as for example described in [HS05]. In this approach it is possible to ask queries such as: Does a statement hold in all versions? Does it hold in a particular prior version? Since when does a statement hold?

Dealing with Inconsistent Information Representing knowledge in networked ontologies – instead of single isolated ontologies – opens new approaches to dealing with contradicting information. Consider a scenario with multiple information sources that do not contain contradictory information if considered isolated. Yet, taking other information sources as contextual information into account may lead to inconsistencies. A global interpretation may not allow to answer queries in a meaningful way. However, representing each information source as a module in a networked ontology with local model semantics can help to overcome this problem. Further, the knowledge about the relationships between the individual nodes can be used in resolving inconsistencies.

4.3 C-OWL and Distributed Description Logics

4.3.1 Overview

[BGH$^+$03] defines context as "a model of some domain which is supposed to encode the view of a party". C-OWL (Context OWL) has been designed as an extension of OWL in order to contextualize ontologies, which means localizing their contents and thus enable to encapsulate information. Consequently, in the C-OWL approach, multiple ontologies are treated as separate entities, between which information interchange is realized only by explicit mappings (so-called *bridge rules*) [BGH$^+$03]. These bridge rules can be used to specify subsumption, equivalence, and disjointness of concepts, roles and individuals from different ontologies.

Formally[1], we consider an *OWL space* being a family of ontologies $\{\langle i, O_i\rangle\}_{i\in I}$ for an index set I. The ontologies are supposed to be ABox and TBox in the description logic $\mathcal{SHOIQ}(D+)$.

A *bridge rule from* i *to* j is a statement of the form $i : x \xrightarrow{\sqsubseteq} j : y$, $i : x \xrightarrow{\sqsupseteq} j : y$, $i : x \xrightarrow{\equiv} j : y$, $i : x \xrightarrow{\perp} j : y$, or $i : x \xrightarrow{*} j : y$ for x and y being concepts, or individuals, or roles from O_i and O_j, respectively. A set M_{ij} of bridge rules from i to j will be called *mapping*.

A *context space* is a pair consisting of an OWL space $\{\langle i, O_i\rangle\}_{i\in I}$ and a family $\{M_{ij}\}_{i,j\in\mathcal{I}}$ of mappings.

Following this distributed approach, the semantics of C-OWL is formalized as a *Local Model Semantics (LMS)*, i.e., a separate model is assigned to every ontology. Correspondences between entities of different models are formalized via *domain relations* which are in turn used to interpret the bridge rules. From this point of view, a context can be seen as a partial and approximate theory of the world from an individual's perspective. Hence it is not necessarily part of the structure of the world, rather a way of structuring an individual's representation. Reasoning will then be carried out locally with respect to a single context and is shared only via explicitly specified connections, following the principle of locality and compatibility.

Given an ontology O_i, a *local interpretation* $\mathcal{I}_i = \langle \Delta^{\mathcal{I}_i}, (.)^{\mathcal{I}_i}\rangle$ of this ontology is either a $\mathcal{SHOIQ}(D+)$ interpretation for O_i or a so-called *hole*, which by definition satisfies all axioms and facts.

An *interpretation for a context space* $\langle\{\langle i, O_i\rangle\}_{i\in I}, \{M_{ij}\}_{i,j\in\mathcal{I}}\rangle$ consists of

- local interpretations of all ontologies O_i and
- a family $\{r_{ij}\}_{i,j\in I}$ of *domain relations* with $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$.

Such an interpretation $\mathcal{I}$ will be called model of the underlying context space if it satisfies the bridge rules, i.e.:

---

[1] We refer to [BGH$^+$03] for the terminology.

1. $\mathcal{I} \models i : x \xrightarrow{\sqsubseteq} j : y$ iff $r_{ij}(x^{\mathcal{I}_i}) \subseteq y^{\mathcal{I}_j}$,
2. $\mathcal{I} \models i : x \xrightarrow{\sqsupseteq} j : y$ iff $r_{ij}(x^{\mathcal{I}_i}) \supseteq y^{\mathcal{I}_j}$,
3. $\mathcal{I} \models i : x \xrightarrow{\equiv} j : y$ iff $r_{ij}(x^{\mathcal{I}_i}) = y^{\mathcal{I}_j}$,
4. $\mathcal{I} \models i : x \xrightarrow{\perp} j : y$ iff $r_{ij}(x^{\mathcal{I}_i}) \cap y^{\mathcal{I}_j} = \emptyset$, and
5. $\mathcal{I} \models i : x \xrightarrow{*} j : y$ iff $r_{ij}(x^{\mathcal{I}_i}) \cap y^{\mathcal{I}_j} \neq \emptyset$.

4.3.2 Characterization Along Our General Dimensions

After introducing C-OWL, we will now show how the C-OWL approach fits into our general definition of context in Section 2.1.

We consider a C-OWL context space with (local) ontologies O_i where $i \in I$. Choosing therefrom one local ontology O_k to focus on, let L_k be the local language of O_k, i.e. the set of statements only referring to concepts, individuals, and roles from O_k. Furthermore, let L be the language built up from all concepts, individuals, and roles from all ontologies of the context space. So obviously $L_k \subseteq L$. We use L_k as knowledge language and L as context language. The semantic of O_k is then the set $S(O_k) \subseteq L$ of the consequences we can derive when simply discarding all "non-local" information, i.e. that about the other O_j with $k \neq j$ and all bridge rules $\{M_{ij}\}_{i,j \in \mathcal{I}}$. All other ontologies together with the bridge rules constitute the context C for O_k. We are aware, that our notion of context does not coincide with the use of the term in C-OWL, since in the C-OWL terminology, every O_k is conceived as a context on its own. However, we argue that our terminology captures the intention more precisely, since e.g. the information present in the mappings would in C-OWL terms be considered as outside of any context.

Obviously, O_k's "global theory" $S'(O_k, C)$ – all derivable consequences taking into account the distributed information of the considered context space – in general deviates from $S(O_K)$.

Characterizing this kind of semantic, we find that it is obviously conservative, extensive, knowledge-monotone, and context-monotone.

Furthermore, [BS02] shows that a knowledge base in distributed description logic (the underlying logic of C-OWL) only considering into and onto rules (the two first ones) between concepts can be transformed into a classical $\mathcal{SHIQ}$ knowledge base. Then, under these restrictions, the C-OWL definition of context can be considered to be (at least) dependently reducible.

4.3.3 Usage Scenarios

C-OWL for Supporting Viewpoints C-OWL has a direct application for supporting reasoning with different viewpoints. Here, each viewpoint is represented in terms of a separate model with a local interpretation [GG98]. Relations between different viewpoints are represented by context mappings that constrain the local interpretations.

C-OWL for Pervasive Computing Although work from McCarthy and Guha consider contexts as independent theories related to some particular knowledge field, Fausto Giunchiglia rather consider contexts as concurrent viewpoints on the same information. He expresses the relations between contexts as ń mappings ż used for importing information under some context into another. This approach can be useful in pervasive computing when several information sources provide comparable information. These works found their way within semantic web tools through the C-OWL language [BGH$^+$03]. A comparison of both approaches is made in [SB04].

4.4 Context Logics

4.4.1 Overview

In artificial intelligence, the notion of context generally concerns the representation and use of information. The notion is used to account for phenomena such as the context of validity of information [dK86] and the efficiency of reasoning in narrower contexts [Guh95].

An early, influential but sketchy attempt to formalize context dependency was made by John McCarthy [McC86]. The basic idea was to mark the dependency of propositions on context and track this dependency through changes of context and stages of reasoning. The basic step was to move from a (simple) proposition to the (meta) proposition that the proposition in question is true in a context. The syntax for such meta-propositions, retained in later developments, was

$$ist(c, p), \text{ for proposition } p \text{ and context } c$$

Contexts thus entered the theory as objects, enabling the theory to express changes of context and the effects of such changes on propositions. Changes of context typically involve making or dropping assumptions, so the syntax also allows compound, functional terms for contexts such as $assuming(p, c)$ (the context obtained from c by assuming p). With such terms, the theory can cover logically interesting consequences of context change, expressed as

$$ist(c, p) \Longrightarrow ist(d, q)$$

and similar formulas, called *lifting axioms* (in reference to the case when the change of context involves dropping assumptions). McCarthy's ideas were later developed by others, and found their way into a working AI system (Cyc, in the form of micro-theories [Guh95]). The present state of this line of development is characterized by the formal system PLC (Propositional Logic of Context) [BBM95], which has also been applied to the semantic Web (the aggregation problem [GMF04]).

The second main line of developing the logic of context stems from the ideas of Fausto Giunchiglia [Giu93]. This line of development does not reify contexts in propositions (i.e. it does not allow to speak about contexts as first-class citizens), but uses them on the meta-level to index or collect propositions (in sub-theories). The model-theoretic part of the resulting theory has also been applied in the theory of databases as the Local Relational Model [SGMB03]. The proof-theoretic part that formalizes contextual reasoning – MCS (MultiContext Systems) – [GS94] uses *bridge rules* in the style of natural deduction in (the approximate) place of lifting axioms. Relating these approaches to each other, a recent comparison of the two sorts of context logic shows that MCS can encode PLC, but not the other way around [SB04]. More generally, without going into the (considerable) technical details of both the logics and their comparison, it can be summarily said that "bridging is better than lifting". That is, the comparison shows that MCS works better than PLC for modelling context dependency not only in reasoning but also in truth-value assignment. The common reason why PLC lags behind MCS along all these dimensions of modelling adequacy might be characterized by saying that PLC takes context dependency too literally: it presumes some sort of initial, implicit unity that is then diversified into different contexts, whereas MCS does not presume such a unity, and emphasizes the opposite process (any unity only comes from “stitching” together the contributions of different contexts).

4.4.2 Characterization Along Our General Dimensions

Comparing McCarthy’s formalization of context with the formalization proposed in this deliverable, certain differences are immediately obvious. For one thing, McCarthy’s ist predicate works on individual propositions, whereas the S and S' functions work on theories. Second, any ist statement is itself made in context, whereas S' statements are themselves context-free. If we neglect this point, we could express the relation between the two approaches by the formula $S(ist(c,p)) \Longrightarrow S'(p,c)$ (”if you know what it means that p is true in c, you know what p means in c”). This formula falls under the general idea that explicating truth-conditions is a way of explicating meaning.

The second main line of context logics mentioned above is realized in the C-OWL proposal and is shown how to fit our general dimensions in 4.3.2.

4.4.3 Usage Scenarios

Context logics can be applied to many usage scenarios where the context can be modeled explicitly. These include for example:

Context Logics for Dealing with Temporal Information Here, the context is used to state that a given statement is valid at a particular point in time.

Context Logics for Supporting Viewpoints Similarly the context can be used that a statement belongs to a particular viewpoint.

Context Logics for Pervasive Computing This kind of approach can be used in pervasive computing in order to integrate and interpret data provided by sensors. Taking advantage of the theory associated with the sensor enables to reduce the ambiguity of the data it delivers. In that view, raw data issued from sensors, are generally not weakened but rather enriched (and aggregated with other information sources allowing to further precise their interpretation).

4.5 Temporal Models

4.5.1 Overview

One particular kind of contextual information is temporal: in order to adequately model events and changing domains, an appropriate formalism has to accommodate means to describe and reason about how events are temporally related. Hereby it is essential to distinguish whether the change which has to be described takes place in the described domain or in the description itself. Clearly, the information “Peter became father in 2006” is different from “in 2006, the database has been updated by the fact that Peter is a father”.

Providing temporal context can be subdivided into two tasks:

- providing a temporal model of the described domain specifying for the concepts used in the model how they can be temporally situated to each other. A prominent example would be causal relationships: knowing that event E_1 causes event E_2 it would be natural to specify that E_2 starts after – or at most simultaneously with – E_1. Likewise, there are many other events with fixed order: birth and death of a person, takeoff and landing of an airplane at a particular flight etc. See Figure 4.1 for a simple temporal model of a tsunami. The model relates three events: Earthquake, Waves and Tsunami with temporal relations, and specifies the duration of an event.

- providing a "temporal reasoning environment" taking into account special interpretation constraints on temporal predicates due to the temporal structure. As a simple example: knowing that some event E_1 occurs before event E_2 and that event E_3 happens during event E_2, we can conclude that E_1 occurs before E_3 as well. Such relationships and events have been modelled by e.g. Allen’s *interval algebra* [All83]. In Allen’s algebra, events are represented by time intervals (in contrast to time points), and an exhaustive set of possible relations between them is defined (see Figure 4.2). The algebra provides computational mechanism to find

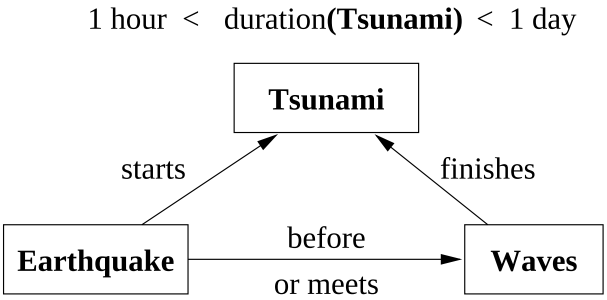


Figure 4.1: A simple temporal model of a tsunami

relations between indirectly related events, to detect possible inconsistencies, and to order events on a linear timescale.

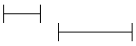
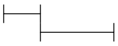
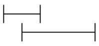
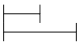
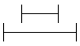
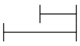
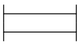
Relation	**Symbol**	**Inverse**	**Meaning**
X before Y	b	bi	
X meets Y	m	mi	
X overlaps Y	o	oi	
X starts Y	s	si	
X during Y	d	di	
X finishes Y	f	fi	
X equals Y	eq		

Figure 4.2: Allen's basic temporal relations

Using these two kinds of contextual information would enable reasoning on given instances. E.g. it could be deduced, whether two events (departure and arrival) can belong to the same "process" (a special trip) with respect to their temporal arrangement.

4.5.2 Characterization Along Our General Dimensions

Imagine an analysis of the past history of a news feed, together with some background knowledge was used for interpreting the meaning of a new news item. More precisely, the background knowledge consists of various temporal models describing different situations or processes of interest. Given a new news item N, some temporal model is selected by taking the past history of the news feed into account. The model gives a possible interpretation to the news item N, which could otherwise be just lexically related to the news history. We can describe this by means of our abstract context definition.

The temporal model selected based on the past history of the news feed is a context C for the new news item N. Without the model, N would be transformed into some ontological knowledge K, with semantics $S(K)$. The context C (consisting of the information conveyed by the previous news items as well as possibly a set of Allen-style rules of how time intervals can be related to each other), however, allows to create (more accurate) ontological knowledge K' from N, with associated semantics $S(K')$. The context semantics function S' can thus be described by $S'(K, C) = S(K')$.

4.5.3 Usage Scenarios

Dealing with Temporal Information Temporal models can be used to interpret and relate data with explicit temporal information. A typical example is news analysis. News items come from different sources but have a timestamp attached. If temporal terms within news can be identified, then an appropriate temporal model is invoked, and news items (which are lexically unrelated) can be found to be semantically related (through temporal relations in the model).

4.6 Context-based Selection Functions

4.6.1 Overview

Many non-standard reasoning tasks (including paraconsistent reasoning, ontology diagnosis, ontology evolution) rely on the notion of a selection function to determine for a given knowledge base K a subset $K' \subset K$ that is *relevant* for a particular task or context. Often the context is provided in the form of additional background knowledge, represented in some context language, which may or may not be the same language as the knowledge language itself. For example, [HvHtT05] uses a given query as context information for selecting a relevant subset of an (inconsistent) knowledge base for meaningful query answering.

Selection functions such as those defined in [HvHH$^+$05] only rely on syntactic properties, e.g. determining how the axioms in C are structurally connected with those in K, in order to select the relevant subset. Of particular interest however are non-uniform, semantically inspired selection functions exploiting domain-specific background knowledge. Another option is to use weak background knowledge like co-occurrence of concept-names on the Web as the basis for a lightweight semantic selection function.

4.6.2 Characterization Along Our General Dimensions

We can formalize a selection function in terms of our knowledge and context language as follows: A selection function sf is a mapping

$$sf : \mathcal{KB}(L) \times \mathcal{KB}(C) \rightarrow \mathcal{KB}(L) \text{ such that } sf(K, C) \subseteq K)$$

The modified semantics of K in the context C is then simply that of the original semantics after applying the selection function to K:

$$S'(K, C) := S(sf(K, C))$$

It is easy to see that in general S' is not extensive, as it is required that $sf(K, C) \subseteq K)$. This is on purpose, as the goal of the selection function is to determine a relevant subset of K to separate wanted from unwanted consequences. The context semantics of a selection function is dependently reducible, in fact, the selection function itself constitutes the reduction function σ.

Whether S' is conservative, idempotent, context-monotone and knowledge-monotone is not determined by the general definition above, but will depend on further properties of the selection function.

4.6.3 Usage Scenarios

Supporting Viewpoints and Perspectives Context-based selection functions can support different viewpoints or perspectives for a given knowledge base by selecting the knowledge that is relevant for the given perspective. For example, consider a knowledge base containing statements about Rudi Studer like
$Human(Rudi), headOf(AIFB, Rudi), Institute(AIFB),$
$Professor \sqsubseteq \exists headOf^{-1}.Institute, Human \sqsubseteq Mammal$

If we now assume a selection function that is able to select relevant axioms for a given context for the reasoning task of instance classification, we might obtain in a research context: $Professor(Rudi), Human(Rudi)$, while a biology context might return: $Mammal(Rudi), Human(Rudi)$.

Dealing with Inconsistent Information Dealing with inconsistent information is a classical application domain of selection functions. The basic idea is that based on an inconsistent knowledge base KB a maximal consistent subset $KB' \subset KB$ is selected, which allows meaningful query answering [HvHtT05]. In general, there may be many such maximal consistent subsets, which may result in different answers to a given query. With a context-dependent selection function, it is possible to select the subset that is most appropriate for a given context.

Personalization A similar intuition can be applied for personalization. Depending on the user profile, different subsets of a given knowledge base may be relevant. Personalized query answering can thus be realized using a context-dependent selection function, where the user profile serves as the context.

Dealing with Uncertain Information In cases where knowledge bases contain uncertain knowledge, it may be required that a certain subset is selected for further processing, e.g. for the task of query answering. Often information about the uncertainty is available as contextual information, e.g. in the form of provenance information. It can thus be exploited for the selection of the certain subset. This kind of context exploitation has for example been used in [HV05a] for the task of ontology learning: The learned ontologies are augmented with information about the confidence and relevance of ontology elements and then used for generating consistent ontologies that contain knowledge that is "most likely correct".

Scalability One approach to improve the performance of reasoning processes is to decrease the size of the knowledge base by selecting the subset of knowledge that is relevant for the particular reasoning task. E.g. for the task of query answering, the context is provided by the query under consideration. A context-dependent selection function would select the a subset of the knowledge base based on this context information.

4.7 Approximate Reasoning

4.7.1 Overview

In different application areas, the requirements for reasoning services may be quite distinct; while in certain fields (as in safety-critical technical descriptions) precision is to be rated as a crucial constraint, in other fields less precise answers could be acceptable if this would be the price for a faster response behavior. A context could provide the preference of particular tasks with respect to precision vs. time requirements.

Introducing approximate reasoning in the Semantic Web field is motivated by the following observation: most nowadays' specification languages for ontologies are quite expressive, reasoning tasks are supposed to be very costly with respect to time and other resources - this being a crucial problem in the presence of large-scale data. As a prominent example, note that reasoning in most description logics including general concept inclusion axioms (which is simply standard today) is at least EXPTIME complete, if individuals are involved (as it is the case for OWL DL) even NEXPTIME complete. Although those worst case time complexities are not likely to be thoroughly relevant for the average behavior on real-life problems, this indicates that not every specifiable problem can be solved with moderate effort. In many cases, the time costs will be the most critical ones, as a user will not be willing to wait arbitrarily long for an answer. More likely, he would be prone to accept “controlled inaccuracies” as tradeoff for a quicker response behavior. However, the current standard reasoning tools (though highly optimized for accurate, i.e., sound and complete reasoning) do not comply with this kind of approach: in an all-or-nothing manner, they will provide the whole answer to the problem after the complete computation but provide no intermediate “approximate” information at all. So, a desirable feature of a new kind of reasoning tool would be an “anytime behavior” denoting the capability of yielding approximate answers to reasoning queries during ongoing computation: as time proceeds, the answer will be continuously refined to a more and more accurate state until finally the precise result is reached. Clearly, one has to define this kind of behavior (and especially the notion of the intermediate inaccuracy) more formally; one approach related to DL formalisms would be to employ the depth of concept descriptions for measuring similarity (as a possible threshold for accuracy) between them.

Altogether, there are several starting points for approximation in an ontology reasoning setting. The first distinction to be made is whether approximation (and thus: controlled inaccuracy) is introduced at the language level – i.e. on the reasoning input side – or in the reasoning algorithms themselves. The first option would mean to weaken the description of the terminology, the instances, and/or queries, the second one to modify the reasoning methods or even completely design them anew. Therefore, in general, the first option is to be preferred since this allows to reuse existent and optimized reasoning systems.

An obvious technique for approximation on the ontology description and query level is to reduce the complexity of a reasoning task by *simplification.* There are many ways for this, we will just refer to two concrete approaches of them

- “Cutting” formulae at certain role depth. Deeper subformulae are substituted by $\top$ or $\bot$. Depending on the concrete substitution technique, one can keep either soundness or completeness.

- Alter disjunction into conjunction.
  Since disjunction is known to be a major source for computational blowup (roughly spoken, every possibility expressed in the disjunction has to be considered independently), one soundness-preserving technique would be to substitute all disjunctions occurring in the (normal form of the) specification the by conjunctions [HV05b].

There is also a body of work on achieving scalability for semantic web reasoning, which we will use as a base. We can distinguish at least three aspects. First, there is recent work concerning distributed reasoning with ontologies, Second, there are novel reasoning algorithms which for the first time lead to worst-case optimal implementations for certain reasoning tasks. Third, we aim to build on recent developments in using heuristics and approximate reasoning techniques for achieving scalability. These approaches need to be extended and adapted to the specific needs of NeOn.

4.7.2 Characterization Along Our General Dimensions

With respect to our semantics' definition, the following statements could be made: Language simplification refers to embedding mappings $s : L_{\mathrm{k}} \to L'_{\mathrm{k}}$ with $L'_{\mathrm{k}} \subseteq L_{\mathrm{k}}$ as well as $s(\varphi) = \varphi$ for all $\varphi \in L'_{\mathrm{k}}$. Thus, because of losing information, the approximate semantics $S_{\mathrm{appr}} : \mathcal{KB}(L'_{\mathrm{k}}) \to \mathcal{KB}(L'_{\mathrm{k}})$ becomes “coarser” than the original (accurate) one (hereby note that approximate semantics has always to be defined against the intended “limit”). It remains to formalize the influence, a context C has on this. In this case we think of a context as providing a simplification function s, so we would end up with the semantics:

$$S' : \mathcal{KB}(L_{\mathrm{k}}) \times \mathcal{KB}(L_{\mathrm{c}}) \to \mathcal{KB}(L_{\mathrm{k}}) \text{ with } S'(K, C) = S_{\mathrm{appr}}(\{s(\varphi) \mid \varphi \in K\})$$

The further aim would be to proceed via a sequence $L_{\mathrm{k}}' \subset L_{\mathrm{k}}'' \subset L_{\mathrm{k}}''' \ldots L_{\mathrm{k}}$, such that we again end up with the accurate semantics.

4.7.3 Usage Scenarios

Scalability Approximate reasoning is closely related to the scalability issue in our usages of context. In this case, contextual information is used to select relevant parts of the knowledge base to improve the efficiency of reasoning. For example, the simplification function s defined above selects a subset of a knowledge base.

Ontology Adaption and Views Approximate reasoning can be realized via ontology adaptation and views on ontologies. For example, the projection and selection operations proposed in Section 3.8 may be useful to define a simplification function.

Supporting Different Viewpoints and Perspectives The approximate reasoning approaches can be applied to improve efficiency of multi-viewpoint reasoning. A preliminary work has been done in [Stu06] where a notion of approximate entailment in the propositional logic is applied to deal with multi-viewpoint reasoning in OWL ontologies.

4.8 Reasoning with Inconsistent Ontologies

4.8.1 Overview

Inconsistency reasoning addresses the problem that answers obtained via classical reasoning are useless if the underlying knowledge base is inconsistent, i.e. contains contradictory information. This is due to the fact that in classical logic inconsistency causes *everything* to be a logical consequence of the knowledge base, which effectively means that reasoning breaks down in this case.

However, realistic application scenarios, in particular for networked ontologies over real data, should be more robust towards contradictory information. Inconsistency reasoning thus strives to develop methods and algorithms for performing meaningful and useful reasoning over inconsistent data.

To date, one can distinguish two major approaches to inconsistency reasoning, and we will shortly discuss them both.

The first approach is based on the idea that contradictory information be removed before applying classical reasoning algorithms. This can be realized e.g. by starting with an empty (thus consistent) ontology and to incrementally select and add such axioms to that ontology, which do not result in inconsistency. This is done until a satisfying answer to a given query is obtained. This selection is typically guided via a selection function, which is based on some notion of relevance. This relevance may of course depend on the particular context. Depending on the context, the inconsistency reasoner could thus return different answers suitable for the context currently at hand. The approach just described is of course closely related to that in 4.6.

The second approach does not modify the knowledge base but changes the semantics under which it is reasoned with, employing a so-called *paraconsistent semantics*. The semantics employed in this case uses four truth values, namely for *true* (t), *false* (f), *undetermined* (u) and *overdetermined* (o). The fourth truth value, *overdetermined*, stands for contradictory information. I.e., intuitively, if an assertion gets assigned the truth value *overdetermined*, then this assertion is considered to be *true* and *false* at the same time. Reasoning in such a *four-valued logic* can be done based on a corresponding model-theoretic semantics. Naturally, four-valued reasoning should coincide with two-valued reasoning in case the knowledge base is consistent. Apart from this constraint, there exist a number of different proposals for paraconsistent semantics, which serve different needs.

Contextual information is important for paraconsistent reasoning in order to determine the parameters for the particular paraconsistent semantics to be employed, i.e. decisions on how to exactly deal with contradictory information will be guided by contextual information.

4.8.2 Characterization Along Our General Dimensions

Let K be an inconsistent ontology or knowledge base. In the first approach based on the removal of contradictory information, let $K' \subseteq K$ be the consistent subset of K which is used for answering some given query. The contextualized semantics of K is thus expressed as the classical semantics of K'. The context semantics in this case is dependently L-reducible. For the second approach based on paraconsistent reasoning, the semantics of K is changed to a paraconsistent semantics, based on the contextual information.

4.8.3 Usage Scenarios

Dealing with Inconsistent Information The approaches to reasoning with inconsistent ontologies can be applied to deal with inconsistent information in a single knowledge base. That is, when different knowledge bases are merged into a single knowledge base, it is most likely that inconsistency will arise. Traditional logics, such as first order logic or description logics, suffer from the triviality problem. That is, a single contradiction in the knowledge base leads to a knowledge base which entails everything. Our approaches can be applied to obtain meaningful information even if the knowledge base is inconsistent.

Dealing with Uncertain Information Our approaches can also be applied to deal with uncertainty which is due to incomplete or partial information. For example, given a possibilistic description logic knowledge base, a revision-based approach in [QLB06] can be used to resolve the inconsistency in the knowledge base.

4.9 Probabilistic Logics and Bayesian Networks

4.9.1 Overview

Dealing with probabilistic uncertainty in the Semantic Web has been recognized as an important problem in the recent decades. Many approaches have been proposed to extend description logics with probabilistic reasoning [Jae94, Hei94, GL02, DS05, NF04, DP04, KLP97]. These approaches can be classified according to ontology languages, the supported forms of probabilistic knowledge and the underlying probabilistic reasoning formalism. Heinsohn [Hei94] and Jaeger [Jae94] independently present a probabilistic extension of the description logic $\mathcal{ALC}$ which is based on probabilistic reasoning in probabilistic logics. The work in [Hei94] allows to represent generic probabilistic knowledge about concepts and roles. However, it does not allow for assertional knowledge about concept and role instances. In contrast, the work in [Jae94] allows for terminological and assertional probabilistic knowledge about concept, roles and instances respectively, but does not support assertional probabilistic knowledge about role instance. In [GL02], the authors present a probabilistic extension of the description logic $\mathcal{SHOQ}$(**D**), called P-$\mathcal{SHOQ}$(**D**). The underlying probabilistic reasoning formalism is based on the notion of probabilistic lexicographic entailment. The probabilistic knowledge about concept instances is expressed by a conditional constraint $(D|\{o\})[l, u]$ with concept D, individual o and real numbers $l, u{\in}[0, 1]$. However, P-$\mathcal{SHOQ}$(**D**) does not allow for probabilistic knowledge on role instances. Recently, Dürig and Studer [DS05] proposed another probabilistic extension of $\mathcal{ALC}$, which allows for assertional probabilistic knowledge about concept and role instances. However, it does not allow for terminological probabilistic knowledge.

In [KLP97], a probabilistic generalization of the CLASSIC description logic, called P-CLASSIC, is proposed. Unlike the work mentioned above, it is based on inference in Bayesian networks as underlying probabilistic reasoning formalism. Furthermore, P-CLASSIC only allows for terminological probabilistic knowledge about concepts and roles, but does not support assertional knowledge about concept and role instances. Later, Ding and Peng [DP04] propose a probabilistic extension to ontology language OWL, called BayesOWL, which incorporates Bayesian networks and OWL. They develop a set of translation rules to convert an OWL ontology into a Bayesian network.

The above work only discusses the probabilistic extension of an ontology language(s). To support networked ontologies, we need to consider the mapping between two ontologies where uncertainty exists. In [PDYP05], the authors propose a Bayesian network approach to ontology mapping, which is based on BayesOWL. In [MNJ05], a probabilistic ontology mapping tool, which is based on Bayesian networks, is proposed. More work can be done along this direction.

4.9.2 Characterization Along Our General Dimensions

Let K be a knowledge base in an ontology language, for example, description logics. The context language is the conditional probabilistic terminology for probabilistic logic or conditional probability table for Bayesian networks. The semantics of the context language is a knowledge base which consists of K and probabilistic terminologies.

Let us take P-$\mathcal{SHOQ}$(**D**) for example. The knowledge base K is simply a description logic knowledge base with DL semantics. The context language C is a set of probabilistic terminology which consists of a generic part and an assertional part. The semantics of the context language is modeled by a *lexicographic consequence* or a *tight lexicographic consequence*, where a conditional constraint $(D|C)[l,u]$ is a lexicographic consequence of a set of terminology axioms and conditional constraints $\mathcal{F}$ under a *general probabilistic terminology* $\mathcal{P}_g$ iff $Pr(D){\in}[l,u]$ for every lex-minimal Pr of $\mathcal{F} \cup \{(C|\top)[1,1]\}$.

4.9.3 Usage Scenarios

Dealing with Uncertain Information The probabilistic approaches can successfully be applied to deal with uncertain information. The probabilistic information can be either learned from the data or provided by the experts.

Scalability The probabilistic approaches can be also useful for scalable reasoning. For example, if probabilistic information is available, we can choose a subset of a knowledge base containing those formulae which have highest probability values.

Supporting Different Viewpoints and Perspectives When there are different viewpoints, we need context mappings to link them. These context mappings can be simply set by experts or automatically produced. Probabilistic information can provide additional information that can be used to guide ontology mapping [MNJ05].

4.10 Possibilistic logic

4.10.1 Overview

Possibilistic logic [DLP94] or possibility theory offers a convenient tool for handling uncertain or prioritized formulas and coping with inconsistency. It is very powerful to represent partial or incomplete knowledge [BLP04]. There are two different kinds of possibility theory: one is qualitative and the other is quantitative. Qualitative possibility theory is closely related to default theories and belief revision [DP91, BDP92] while quantitative possibility can be related to probability theory and can be viewed as a special case of belief function [DP98]. One of the major problems with the quantitative possibility theory is that the weights attached to formulas are usually hard to obtain. When numerical information is not available, we often use qualitative possibility theory. In this case, a possibilistic knowledge base can be viewed as a stratified knowledge base, i.e. knowledge bases in which all pieces of information are assigned a rank.

The application of possibilistic logic to deal with uncertainty in the Semantic Web was first studied in [Hol95]. However, their approach inherits a serious problem, called drowning problem, from possibilistic logic. That is, too much information is lost after inconsistency handling. In [MLB05], an algorithm, called refined conjunctive maxi-adjustment (RCMA for short) was proposed to weaken conflicting information in a stratified DL knowledge base and some consistent DL knowledge bases were obtained. This algorithm is an adaptation of an existing inconsistency handling algorithm in the possibilistic logic setting. To weaken a terminological axiom, they introduced a DL expression, called cardinality restrictions on concepts. An interesting problem is to explore other DL expressions to weaken a conflicting DL axiom (both terminological and assertional). In [QLB06], a revision-based approach is proposed to deal with inconsistency in a stratified DL knowledge base. This approach weakens both conflicting Tbox and Abox statements using nominals.

The approaches for inconsistency handling in possibilistic logic are related to the work in Section 4.8. There are two challenging problems here. The first problem is how to stratify the DL knowledge bases or how to assign ranks to each DL statement. A preliminary work on this problem has been discussed in [QP06]. The second problem is how to resolve inconsistency in a stratified knowledge base.

4.10.2 Characterization Along Our General Dimensions

Let K be a knowledge base in an ontology language L. The context information can be used to obtain the weights or priority levels attached to formulae in the knowledge base. That is, $S(K,C)$ is a weighted knowledge base or a prioritized knowledge base. Furthermore, the context information can be also used to guide how to resolve inconsistency in the knowledge base. It is clear that this context language is conservative. However, it is neither extensive nor knowledge-monotonic, because the underlying logic (i.e. possibilistic logic) is inherently nonmonotonic.

4.10.3 Usage Scenarios

Dealing with Uncertain or Vague Information The possibilistic logic based approaches can be used to deal with uncertain information where uncertainty is due to incomplete or partial information. The semantics of the possibilistic logic is closely related to the membership function in fuzzy logics.

Dealing with Inconsistent Information Another important feature of possibilistic logic is that it can be used to handle inconsistency. Several inconsistency-tolerant consequence relations have been proposed in possibilistic logic (or its variants).

Supporting Different Viewpoints and Perspectives Possibilistic logic based approaches may be useful to integrate different existing viewpoints. Many possibilistic merging approaches have been proposed [Qi06] and they can be adapted to the integrate different viewpoints. For example, we may be interested in a context dependent approach which was proposed in [LQB06] to merge different possibilistic knowledge bases.

Chapter 5

Conclusion

5.1 Summary

The goal of this deliverable was to provide a state-of-the-art overview of context representation formalisms as a basis for a number of subsequent deliverables. We first gave a generic definition of context. We then identified several use cases of our context representation formalisms. These use cases are important for the evaluation of the approaches to representing and reasoning with context. Finally, we gave an overview of some approaches for representing and reasoning with context which may be relevant for NeOn. To facilitate the comparison of different approaches, we proposed several dimensions which are directly derived from our generic context definition.

5.2 Roadmap

In deliverable D3.1.2 we will define the NeOn formalism for context representation; in the accompanying deliverable D3.2.1 we will develop a prototype for reasoning with contexts based on that formalism.

In this deliverable we have provided an overview of both different usages of context as well as approaches for representing and reasoning with context. While at this point we cannot provide definite statements, which approaches will be supported or not, in the following we attempt to identify those usages and approaches that will be particularly relevant for NeOn.

NeOn has in its core the ambitious scenario that ontologies are developed in the open environment in a distributed fashion. Moreover, it is not just the ontologies and metadata that are distributed, but we also assume that they are built by distributed teams. In terms of the usages of context, this means that *supporting viewpoints and perspectives* will play a paramount role in NeOn. In the scenarios addressed by NeOn, information sources typically cannot be easily integrated without violating the overall consistency of the system. Thus *dealing with inconsistent information* will be another important usage of context, where information about the provenance of ontological structures, about various contexts and user profiles leads to the generation of ŚlocalŠ, consistent views out of a globally inconsistent network of ontologies. Closely related is the problem of *dealing with uncertain and vague information*, which will play an important role in the NeOn scenarios, where ontologies are generated from a variety of sources that may be imprecise, vague and contextualized in the first place (e.g. natural language text) and where automated ontology learning algorithms introduce an additional dimension of uncertainty.

To be able to address these usage scenarios for context, we believe that the following approaches for contexts are relevant for NeOn: The *networked ontology model* developed in WP1 will provide the most obvious form of context: Ontologies will be embedded in a network of ontologies, which forms the context for its interpretation. Depending on the types of relationships in the network of ontologies, different forms of context can be realized: The context may be a temporal one if the network represents a version space; it may be used to connect different viewpoints via alignments, etc. *Reasoning with inconsistent ontologies* exploiting context information will be important when different information sources with contradicting information will be integrated. *Context-based selection functions* appear promising for addressing a number of different problems. However, the development of such functions that go beyond the state-of-the-art syntactic-based functions will require significant research efforts. Finally, a combination of *possibilistic* and *probabilistic logics* seems to be required to deal with the various forms of vagueness and uncertainty in a contextualized way.

The rather wide range of different forms of context poses challenges for an integrated representation format of context, which we will deliver as part of D3.1.2. A major design goal for the representation format will be the compatibility with the current OWL 1.0 standard and possibly upcoming future versions of OWL 1.1 and OWL 2.0. In [VVHC06] we have already presented a proposal of how to embed annotations about ontology elements within an ontology. This representation formalism may serve as a basis for carrying contextual annotations, as we have shown in [VVHC06] for the case of information about provenance. With respect to contextual information about the ontologies themselves, we envision a representation format compatible with OMV (Ontology Metadata Vocabulary, [HPS$^+$05]). This will provide a flexible mechanism to allow extensions for particular forms of context. One such context form currently under consideration as an OMV extension is argumentation information, i.e. information about why certain ontology elements have been introduced.

Bibliography

[AKtKvH06] Zharko Aleksovski, Michel Klein, Warner ten Kate, and Frank van Harmelen. Matching unstructured vocabularies using a background ontology. In *Proc. EKAW, Praha (CZ)*, 2006.

[All83] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[AM97] P. Smets A. Motro. *Uncertainty Management In Information Systems*. Springer, 1997.

[AS96] V. Akman and Mehmet Surav. Steps Toward Formalizing Context. *AI Magazine*, 17(3):55–72, 1996.

[BBG01] M. Benerecetti, P. Bouquet, and C. Ghidini. On the dimension of context dependence: partiality, approximation and perspective. In *Proc. of Third international and interdisciplinary conference on Modeling and Using Context (CONTEXT'01)*, 2001.

[BBM95] Sasa Buvac, Vanja Buvac, and Ian A. Mason. Metamathematics of contexts. *Fundamenta Informaticae*, 23(2/3/4):263–301, 1995.

[BDP92] Salem Benferhat, Didier Dubois, and Henri Prade. Representing default rules in possibilistic logic. In *KR*, pages 673–684, 1992.

[BGGB03] P. Bouquet, C. Ghidini, F. Giunchiglia, and E. Blanzieri. Theories and uses of context in knowledge representation and reasoning. *Journal of Pragmatics*, 35(3):455–484, 2003.

[BGH$^+$03] P. Bouquet, F. Giunchiglia, F. Van Harmelen, L. Serafini, and H. Stuckenschmidt. C-owl: Contextualizing ontologies. In D. Fensel, K. P. Sycara, and J. Mylopoulos, editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2003.

[BLP04] Salem Benferhat, Sylvain Lagrue, and Odile Papini. Reasoning with partially ordered information in a possibilistic logic framework. *Fuzzy Sets and Systems*, 144(1):25–41, 2004.

[BS02] Alexander Borgida and Luciano Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 36–53, London, UK, 2002. Springer-Verlag.

[CCDG05] J. Coutaz, J. Crowley, S. Dobson, and D. Garlan. Context is key. *Communications of the ACM*, 48(3):49–53, 2005.

[Dey01a] A. Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.

[Dey01b] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, 2001.

[dK86] J. de Kleer. An assumption-based tms. *Artificial Intelligence*, 28(2):127–162, 1986.

[DLP94] Didier Dubois, Jérôme Lang, and Henri Prade. Possibilistic logic. In *Handbook of logic in Aritificial Intelligence and Logic Programming*, pages 439–513, 1994.

[Dou01] P. Dourish. Seeking a foundation for context-aware computing. *Human-Computer Interaction*, 16:2–3, 2001.

[DP91] Didier Dubois and Henri Prade. Epistemic entrenchment and possibilistic logic. *Artif. Intell.*, 50(2):223–239, 1991.

[DP98] Didier Dubois and Henri Prade. Possibility theory: qualitative and quantitative aspects. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 169–226, 1998.

[DP04] Z. Ding and Y. Peng. A Probabilistic Extension to Ontology Language OWL. In *Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37).*, Big Island, Hawaii, January 2004.

[DS05] Michael Dürig and Thomas Studer. Probabilistic abox reasoning: Preliminary results. In *Description Logics*, 2005.

[ERP06] Jérôme Euzenat, Fano Ramparany, and Jérôme Pierson. A context information manager for pervasive environments. In *Proc. 2nd ECAI workshop on contexts and ontologies (C&O), Riva del Garda (IT)*, pages 25–29, 2006.

[FGM06] B. Fortuna, M. Grobelnik, and D. Mladenič. Background knowledge for ontology construction. In *Proceedings 2nd ECAI workshop on "Contexts and Ontologies"*, Riva del Garda, Italy, 2006.

[GG98] Fausto Giunchiglia and Chiara Ghidini. Local models semantics, or contextual reasoning = locality + compatibility. In *KR*, pages 282–291, 1998.

[GGMO03] Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari. Sweetening WordNet with DOLCE. *AI Magazine*, 24(3):13–24, 2003.

[Giu93] F. Giunchiglia. Contextual reasoning. *Epistemologia*, 16, 1993.

[GL02] Rosalba Giugno and Thomas Lukasiewicz. P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In *JELIA*, pages 86–97, 2002.

[GMBM05] Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors. *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, volume 3729 of *Lecture Notes in Computer Science*. Springer, 2005.

[GMF04] Ramanathan V. Guha, Rob McCool, and Richard Fikes. Contexts for the semantic web. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2004.

[GMG04] M. Grcar, D. Mladenic, and M. Grobelnik. User profile specification and data gathering module. SEKT deliverable D5.5.1, 2004.

[GS94] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics, or: How we can do without modal logics. *Artificial intelligence*, no. 1, 1994.

[Guh95] R.V. Guha. Contexts : A Formalization and Some Applications. Technical report, 1995.

[Hei94] Jochen Heinsohn. Probabilistic description logics. In *UAI'94*, pages 311–318, 1994.

[Hol95] Bernhard Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. *Int. J. Approx. Reasoning*, 12(2):85–109, 1995.

[HPS$^+$05] Jens Hartmann, Raúl Palma, York Sure, María del Carmen Suárez-Figueroa, Peter Haase, Asunción Gómez-Pérez, and Rudi Studer. Ontology metadata vocabulary and applications. In *OTM Workshops*, pages 906–915, 2005.

[HS05] Zhisheng Huang and Heiner Stuckenschmidt. Reasoning with multi-version ontologies: A temporal logic approach. In Gil et al. [GMBM05], pages 398–412.

[HV05a] Peter Haase and Johanna Völker. Ontology learning and reasoning - dealing with uncertainty and inconsistency. In Paulo Cesar G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey, and Michael Pool, editors, *ISWC-URSW*, pages 45–55, 2005.

[HV05b] Pascal Hitzler and Denny Vrandecic. Resolution-based approximate reasoning for owl dl. In Y. Gil et al., editor, *Proceedings of the 4th International Semantic Web Conference, Galway, Ireland, November 2005*, volume 3729 of *Lecture Notes in Computer Science*, pages 383–397. Springer, Berlin, NOV 2005.

[HvHH$^+$05] Peter Haase, Frank van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. A framework for handling inconsistency in changing ontologies. In Gil et al. [GMBM05], pages 353–367.

[HvHtT05] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with inconsistent ontologies. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 454–459. Professional Book Center, 2005.

[Jae94] Manfred Jaeger. Probabilistic reasoning in terminological logics. In *KR'94*, pages 305–316, 1994.

[KLP97] Daphne Koller, Alon Y. Levy, and Avi Pfeffer. P-classic: A tractable probablistic description logic. In *Proc. of AAAI'97*, pages 390–397, 1997.

[KLWZ03] Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. E-connections of description logics. In Diego Calvanese, Giuseppe De Giacomo, and Enrico Franconi, editors, *Description Logics*, volume 81 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[KLWZ04] Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. E-connections of abstract description systems. *Artif. Intell.*, 156(1):1–73, 2004.

[LG90] Doug Lenat and Ramanathan V. Guha. *Building large knowledge-based systems.* Addison Wesley, Reading (MA US), 1990.

[LQB06] Weiru Liu, Guilin Qi, and David A. Bell. Adaptive merging of prioritized knowledge bases. *Fundam. Inform.*, 73(3):389–407, 2006.

[MB06] I. Mozetič and D. Bojadžiev. Reasoning with temporal context in news analysis. In *Proceedings 2nd ECAI workshop on "Contexts and Ontologies"*, Riva del Garda, Italy, 2006.

[McC86] John McCarthy. Notes on formalizing contexts. In Tom Kehler and Stan Rosenschein, editors, *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 555–560, Los Altos, California, 1986. Morgan Kaufmann.

[MCH05] D. Moldovan, C. Clark, and S. Harabagiu. Temporal context representation and reasoning. In *Proceedings IJCAI-05*, pages 1099–1104, Edinburgh, Scotland, 2005.

[MLB05] Thomas Meyer, Kevin Lee, and Richard Booth. Knowledge integration for description logics. In *AAAI*, pages 645–650, 2005.

[MNJ05] Prasenjit Mitra, Natasha F. Noy, and Anuj Jaiswal. Omen: A probabilistic ontology mapping tool. In *International Semantic Web Conference (ISWC'05)*, pages 537–547, 2005.

[MVB$^+$04] Claudio Masolo, Laure Vieu, Emanuele Bottazzi, Carola Catenacci, Roberta Ferrario, Aldo Gangemi, and Nicola Guarino. Social roles and their descriptions. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *KR*, pages 267–277. AAAI Press, 2004.

[MW04] Prasenjit Mitra and Gio Wiederhold. An ontology-composition algebra. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 93–116. Springer, 2004.

[NF04] Henrik Nottelmann and Norbert Fuhr. pdaml+oil: A probabilistic extension to daml+oil based on probabilistic datalog. In *Proceedings Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 227–234, 2004.

[NP01] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS)*, pages 2–9, 2001.

[PDYP05] Rong Pan, Zhongli Ding, Yang Yu, and Yun Peng. A bayesian network approach to ontology mapping. In *International Semantic Web Conference*, pages 563–577, 2005.

[Qi06] Guilin Qi. *Fusion of uncertain information in the framework of possibilistic logic*. PhD thesis, University of Belfast, 2006.

[QLB06] Guilin Qi, Weiru Liu, and David A. Bell. A revision-based approach for handling inconsistency in description logics. In *NMR'06*, pages 124–132, 2006.

[QP06] Guilin Qi and Jeff Pan. A stratification-based approach for inconsistency handling in description logics. In *Technical Report*, 2006.

[SB04] Luciano Serafini and Paolo Bouquet. Comparing formal theories of context in ai. *Artif. Intell.*, 155(1-2):41–67, 2004.

[SGMB03] L. Serafini, F. Giunchiglia, J. Mylopoulos, and P. A. Bernstein. Local relational model: A logical formalization of database coordination. In P. Blackburn, C. Ghidini, R. M. Turner, and F. Giunchiglia, editors, *CONTEXT*, volume 2680 of *Lecture Notes in Computer Science*, pages 286–299. Springer, 2003.

[SSW05] Luciano Serafini, Heiner Stuckenschmidt, and Holger Wache. A formal investigation of mapping languages for terminological knowledge. In Katja Verbeeck, Karl Tuyls, Ann Nowé, Bernard Manderick, and Bart Kuijpers, editors, *BNAIC*, pages 379–380. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, 2005.

[Sto06] Giorgos Stoilos. D2.5.4 analysis of requirements for further language extensions. Technical Report 2.5.4, Knowledge Web Consortium, 2006.

[Stu06] Heiner Stuckenschmidt. Toward multi-viewpoint reasoning with owl ontologies. In *ESWC*, pages 259–272, 2006.

[SvHB$^+$04] Heiner Stuckenschmidt, Frank van Harmelen, Paolo Bouquet, Fausto Giunchiglia, and Luciano Serafini. Using c-owl for the alignment and merging of medical ontologies. In Udo Hahn, editor, *KR-MED*, volume 102 of *CEUR Workshop Proceedings*, pages 88–101. CEUR-WS.org, 2004.

[VVHC06] Denny Vrandecic, Johanna Völker, Peter Haase, and Philipp Cimiano. A metamodeling approach for annotating ontology elements in owl dl. In *Proceedings of the 2nd International Workshop on Meta-modelling and Ontologies, WoMM*, 2006. to appear.

[ZMS04] Hongwei Zhu, S. Madnick, and M. Siegel. Reasoning about temporal context using ontology and abductive constraint logic programming. Technical Report 15, CISL, MIT, Cambridge, 2004.