Chapter 1
Logical formalisms for Agreement Technologies

Antoine Zimmermann

Abstract This chapter provides an overview of the logical formalisms that have been proposed to define the formal semantics of knowledge systems that are distributed, heterogeneous and multi contextual. The chapter starts with the abstract notions that are common to many of these logics, then focusing on individual formalisms.

1.1 Introduction

Semantic Web standards offer a good basis for representing the knowledge of local agents[1], the schemata, the functionalities and all things that matter in order to achieve a goal in agreement with other agents. However, the formalisms behind these technologies have limitations when dealing with the distributed, open and heterogeneous nature of the systems concerned by Agreement Technologies. In particular, since agents are inherently autonomous, they define their knowledge according to their own beliefs, which can differ from one another or even be inconsistent with other agents' beliefs. Since the standards of the Semantic Web are not concerned about belief and they do not provide the means to compartment knowledge from distinct sources, the conclusions reached when using the global knowledge of disagreeing agents are inevitably inconsistent. Hence, by virtue of the "principle of explosion", all possible statements are entailed.

For these reasons, a number of logical formalisms have been proposed to handle the situations in which pieces of knowledge are defined independently in various contexts. These formalisms extend classical logics—sometimes the logics of Semantic Web standards—by partitioning knowledge from different sources and lim-

Antoine Zimmermann
École Nationale Supérieure des Mines, FAYOL-ENSMSE, LSTI, F-42023 Saint-Étienne, France,
e-mail: antoine.zimmermann@emse.fr

[1] We use the term "agent" to denote any entity which can act towards a goal, such as a service, an application, a device, or even a person or organisation.

iting the interactions between the parts in the partition in various ways. We collectively call these logics *contextual logics*, although they have been called sometimes *distributed logics* [**?**, **?**, **?**] or *modular ontology languages* [**?**]. This chapter aims at presenting a variety of proposals for contextual reasoning, where each approach addresses to a certain extent the problems of heterogeneity, inconsistency, contextuality and modularity.

1.2 General definitions for contextual logics

1.2.1 Networks of Aligned Ontologies

In most of the formalisms presented here, it is generally agreed that local knowledge, defined to serve one purpose from one viewpoint, should conform to a classical semantics, that is, the semantics of standard knowledge representation formats. For instance, the ontology that defines the terms used in the dataset of a single semantic website could be defined in OWL and all the conclusions that can be drawn from it are determined according to the W3C specification. Similarly, the functionalities of a single Web service could be described in WSML, and using this description alone would yield the inferences defined by the WSML specification.

To simplify the terminology, we will use the term *ontology* to denote a logical theory in a language which is local to an agent and a specific purpose. An agent may own several ontologies to describe different types of knowledge, such as describing the domain associated with the application's data, describing local policies, functionalities or computational resources. Agreement Technologies are working on systems composed of many software agents, therefore contextual logics provide a semantics to systems of multiple ontologies. Besides, ontologies developed independently are likely to use disjoint sets of terms (or at least, different identifiers for terms). So, if local ontologies are the only constituent of a contextual logic formalism, then there is no possible interaction between the knowledge associated with a context and the knowledge of another. For this reason, we assume that additional knowledge is present to "bind" ontologies together. We call this additional knowledge *ontology alignments*, which provide an explicit representation of the correspondences between ontologies. In practice, an alignment can take many forms, which depend on the actual contextual logic used. In this chapter, we do not discuss how the alignments are produced.[2]

As a result, the structure for which a contextual logic defines a semantics is a graph-like structure that we call a *network of aligned ontologies* (NAO) where vertices are ontologies and edges are alignments. In theory, ontology alignments could express correspondences between more than two ontologies, so the structure should be a hypergraph in general. But practical ontology matching tools always

[2] This is the subject of Chapter **??**.

produce binary alignments, so we will often consider that NAOs are standard directed graphs.[3]

1.2.2 Local semantics

This section recapitulates the definitions that are common to classical logical formalisms, especially introducing the notions of ontology, interpretation, satisfaction and model.

A local ontology is a logical theory, written in a language of a logic. A *logic* is characterised by:

- a syntax, that is a set of symbols and sentences (or formulas) that can be built with them;
- a notion of interpretations, which define a domain of interpretation and associate symbols with structures over the domain;
- a satisfaction relation, which relates interpretations to the sentences they satisfy.

For example, Description Logics allow symbols for atomic concept, roles and individuals, as well as constructs such as $\exists$, $\forall$ to build ABox or TBox axioms. Interpretations must assign a subset of the domain to a concept name, and a set of pairs to a role name. A subsumption axiom $C \sqsubseteq D$ is satisfied by an interpretation if the set denoted by C is contained in the set denoted by D.

An ontology is simply a set of sentences and when an interpretation satisfies all sentences in an ontology, we say that it is a model of the ontology.

For a logic L, we will write Sen_L to denote the set of sentences (or formulas) defined by L; we write Int_L to denote the interpretations; $\models_L$ to denote the satisfaction relation, and given an ontology O, we note $\text{Mod}(O)$ the set of models of O.

1.2.3 Contextual logics

A contextual logic provides a semantic to networks of aligned ontologies. We describe this particular kind of logics very much like a standard logic is defined, that is, by presenting the syntax, the interpretations and models. First, a contextual logic is defined on top of a set of local logics $\mathbf{L}$, which determine the languages used in local ontologies. The sentences of a contextual logic are of two types: local axioms and cross-ontology correspondences.

First, we assume the existence of a set $\mathbf{C}$ of *context identifiers*. Each context $c \in \mathbf{C}$ is associated with a fixed language $L_c \in \mathbf{L}$. A *local axiom* is written $c{:}\alpha$, where c is a context identifier, and α is a local sentence in the language L_c. A (cross-ontology) *correspondence* is a sentence in an alignment language L_A, which can be of many

[3] Nonetheless, $\mathscr{E}$-connection is a formalism where non-binary alignments can be expressed, as explained in Section 1.6.

forms depending on the actual logic used—as we will see later—but expresses a relation between some terms from distinct contexts. Generally, correspondences express binary relations between terms of two ontologies, such that most contextual logics work on correspondences of the form $\langle e_c, e_{c'}, r\rangle$, where e_c (resp. $e_{c'}$) is an entity (term or construct) from context c (resp. c') and r denotes a type of relations, such as equality or subsumption.[4]

A network of aligned ontologies is in fact a set of local axioms together with cross-ontology correspondences. They can be defined as logical theories in a contextual logic. More formally:

Definition 1.1 (Network of aligned ontologies). A *network of aligned ontologies* (or NAO) in a contextual logic $\mathscr{L}$ is a pair $\langle \Omega, \Lambda \rangle$ where:

- $\Omega = (o_c)_{c\in K}$ is a tuple of local ontologies indexed by a finite set of contexts $K \subseteq \mathbf{C}$, such that $o_c \subset \mathrm{Sen}_{L_c}$;
- Λ, called the set of correspondences, is a finite set of formulas in language L_A.

Interpretations in a contextual logic are composed of two parts: (1) a family of local interpretations, which intuitively assigns an interpretation to each ontology in an NAO, and (2) a structure that interprets cross-context knowledge. Formally:

Definition 1.2. $\langle (I_c)_{c\in C}, \Gamma \rangle$ is an interpretation in the contextual logic if and only if $C \subseteq \mathbf{C}$ and for all $c \in C$, I_c is an interpretation in the language L_c. The structure of Γ depends on the alignment language used by the formalism and it varies depending on the contextual logic the same way the structure of an interpretation varies depending on the local logic used.

To be precise, Γ could be described as an object in a mathematical category which depends on the contextual logic. Many definitions are needed to present the theory of categories, so we prefer to keep the definition looser and simply say that Γ is a structure that depends on the contextual formalism. We provide examples thereafter.

Satisfaction in a contextual logic is constrained by the local semantics, which impose the way local axioms are satisfied:

Definition 1.3 (Satisfaction of a local axiom). A local axiom $i{:}\alpha$ is satisfied by an interpretation $(I_c)_{c\in C}$ if $i \in C$ and $I_i \models_{L_i} \alpha$.

This means that a local axiom is satisfied by a contextual interpretation if it assigns a local interpretation to the context of the local axiom, and the local interpretation satisfies (according to the local semantics) the axiom.

Satisfaction of correspondences are not particularly constrained and the exact definition depends on the contextual logic at hand. As correspondences are usually binary, correspondences of the form $\langle e_c, e_{c'}, r\rangle$ typically constrain the relationship between the local interpretations of e_c and $e_{c'}$ according to the relation symbol r.

[4] Chapter **??** gives a more detailed account on how to discover implicit binary correspondences between two ontologies.

Such constraints often express a form of "equivalence", but even in this restricted setting, semantics vary. A discussion on the semantics of binary correspondences is found in [**?**].

Now a model of a network of aligned ontologies necessarily contains a tuple of *local models*. More formally, if $\langle \Omega, \Lambda \rangle$ is a an NAO where Ω is a (finite) set of ontologies and Λ a (finite) set of alignments, and $\mathbf{I} = \langle (I_c)_{c \in C}, \Gamma \rangle$ is a contextual interpretation, then $\mathbf{I}$ is a model of $\langle \Omega, \Lambda \rangle$ iff $I_o \models_{L_o} o$ for each ontology $o \in \Omega$ and $\mathbf{I}$ satisfies all alignments in Λ. So the set of models of the NAO $\mathrm{Mod}(\Omega, \Lambda)$ restricts the possible local models to a subset of the Cartesian product $\mathrm{Mod}(o_1) \times \ldots \times \mathrm{Mod}(o_k)$, with k the cardinality of Ω.

The remainder of the chapter presents various formalisms that instantiate the notion of contextual logic by setting a concrete syntax for correspondences and defining the satisfaction of the alignments. In each section, we summarise the components of the contextual logic.

1.3 Standard logics as contextual logics

Here we show that a standard logic can be used as a simple contextual logics. Let us assume that the local logics are reduced to a single logic and the alignment language is again the same as the ontology language. Correspondences are satisfied if: (1) the domains of interpretation of all local interpretations are the same; (2) all local interpretations agree on the interpretation of identical local terms; and (3) the correspondence is satisfied by the union of the local interpretations.

In fact, this formalisation exactly corresponds to a standard logic where the meaning of the NAO is the same as a single ontology obtained by making the union of all ontologies and alignments. Therefore, from an inference perspective, there is no difference with a single standard logic, as all axioms will influence all ontologies equally, just as if everything was local. However, there can still be an interest in having local axioms compartmentalised, especially to track the provenance of some knowledge, or in a query mechanism that allow requests on a specific context, which SPARQL allows thanks to the dataset structure. Indeed, SPARQL engines are not simply managing single RDF graphs, they are required to work on a structure composed of separated graphs which are labelled with URIs. However, SPARQL is agnostic with respect to how knowledge from one graph influence knowledge in another. It only enables one to query a portion of the knowledge based on the graph identifiers.

Using a standard logic in a multi-contextual setting is common as it is easier to understand and not controversial, although it is very sensitive to heterogeneity and disagreements across contexts. Therefore, other non-standard approaches were proposed that we discuss next.

Local logics: local logics are all the same but can be of any type.
Correspondence syntax: correspondences are expressed as axioms built using the terms of local ontologies.

Contextual interpretation: an interpretation of a network of aligned ontologies in this logic is simply a tuple of local interpretations such that the domain of interpretation is the same for each context.

Satisfaction of correspondences: since correspondences are standard axioms, the satisfaction of correspondences is as in the local logic.

1.4 Distributed Description Logics (DDL)

Distributed Description Logics (DDL) [**?**] is a formalism which was developed to formalise contextual reasoning with Description Logic ontologies. Therefore, local logics are Description Logics, which is well adapted for the Semantic Web standard OWL. Moreover, cross-context formulas can be defined to relate different terminologies in the form of so called *bridge rules* and written either $i:C \xrightarrow{\sqsubseteq} j:D$ or $i:C \xrightarrow{\sqsupseteq} j:D$ where i and j are two different contexts, and C and D are terms from the contextual ontologies O_i and O_j respectively. A bridge rule $i:C \xrightarrow{\sqsubseteq} j:D$ (resp. $i:C \xrightarrow{\sqsupseteq} j:D$) should be understood as follows: from the point of view of O_j (*i.e.*, in the context j), C is a subclass (resp. superclass) of D.

Local logics: local logics are description logics.

Correspondence syntax: correspondences take the form of into- ($i:C \xrightarrow{\sqsubseteq} j:D$) or onto-bridge rules ($i:C \xrightarrow{\sqsupseteq} j:D$).

Contextual interpretation: DDL interpretations are tuples of local interpretations together with domain relations for each pair of contexts, formally $\langle (I_i),(r_{ij})\rangle$ where I_i are local DL interpretations over domains Δ_i for all i and r_{ij} is a set $r_{ij} \subseteq \Delta_i \times \Delta_j$ for all contexts i and j.

Satisfaction of correspondences: an interpretation $\langle (I_i),(r_{ij})\rangle$ satisfies a bridge rule $i:C \xrightarrow{\sqsubseteq} j:D$ (resp. $i:C \xrightarrow{\sqsupseteq} j:D$) iff $r_{ij}(C^{\mathscr{I}_i}) \subseteq D^{\mathscr{I}_j}$ (resp. $r_{ij}(C^{\mathscr{I}_i}) \supseteq D^{\mathscr{I}_j}$).[5]

This formalism allows different contexts to model the same domain in different ways with a reduced risk of causing inconsistencies due to heterogeneity. Yet, it still allows for cross-ontology inferences, such as: $i:A \sqsubseteq B$, $i:A \xrightarrow{\sqsupseteq} j:C$, $i:B \xrightarrow{\sqsubseteq} j:D$ together entail $j:C \sqsubseteq D$. The reasoner procedure for DDL has been implemented in a peer-to-peer system where each peer embed a local reasoner extended with message exchanges based on the bridge rules they detain [**?**].

1.5 Package-based Description Logics

In package-based Description Logics (P-DL [**?**]), local logics are again description logics and cross-ontology knowledge can only take the form of *semantic imports* of

[5] For a set S, $r_{ij}(S) = \{x \in \Delta^{\mathscr{I}_j} \mid \exists y \in S, \langle x,y\rangle \in r_{ij}\}$.

ontological terms. this formalism was essentially designed to compensate the drawbacks of the OWL import mechanism and improve modularity of Web ontologies. Imports are satisfied when the local interpretation of the imported terms are the same in the importing and imported ontologies.

Local logics: local logics are description logics.
Correspondence syntax: correspondences take the form $O_i \xrightarrow{t} O_j$, which can be read "ontology O_j imports the term t defined in ontology O_i".
Contextual interpretation: in its first definition, P-DL interpretations were simply tuples of local interpretations [**?**]. In later publications, the formulation was revised (yet is equivalent) using domain relations as in DDL, imposing furthermore that the domain relations are one-to-one, that r_{ij} is the inverse of r_{ji} and that the composition of r_{ij} with r_{jk} must be equal to r_{ik}, for all i, j and k [**?**].
Satisfaction of correspondences: an interpretation $\langle (I_i), (r_{ij}) \rangle$ satisfies an import $O_i \xrightarrow{t} O_j$ iff $r_{ij}(t^{\mathscr{I}_i}) = D^{\mathscr{I}_j}$. In earlier versions of the semantics, the condition was that the local interpretations of an imported term must be equal in both the importing and imported ontology.

1.6 $\mathscr{E}$-connections

$\mathscr{E}$-connections is another formalism for reasoning with heterogeneous ontologies [**?**]. Again, different ontologies are interpreted distinctly but formally related using particular assertions. Instead of expressing correspondences of ontological terms, an ontology can connect to another by using special terms (called *links*) which can be combined in conjunction with terms from another ontology. The semantics of links is very similar to the semantics of roles in Description Logics, except that instead of relating elements from the same domain of interpretation, they relate two different domains. So, in $\mathscr{E}$-connections, in addition to local interpretations, domain relations are assigned to each link. The difference with DDL is that the domain relations are not unique per pair of interpretations: they are specific to a link, so there can be many over two different interpretation domains. Moreover, links are used like roles in DL, with the difference that using a link imposes that terms from distinct ontologies are used. For instance, one can define the sentence $C_i \sqsubseteq \exists\langle L_{ij}\rangle D_j$, where $\langle L_{ij}\rangle$ denotes a link between ontologies O_i and O_j, C_i denotes a term of O_i and D_j denotes a term of O_j. Finally, a sentence with multiple links can involve terms from more than two ontologies. Therefore, it is not possible in general to represent a NAO in $\mathscr{E}$-connection as a simple directed graph.

In principle, $\mathscr{E}$-connections serve to relate ontologies about very different domains of interest. For instance, an ontology of laboratories could be connected to an ontology of medical staff. However, $\mathscr{E}$-connection is not particularly appropriate to relate ontologies of similar domains, as there is no way to express formally a form of equivalence between terms of distinct ontologies. Also, in $\mathscr{E}$-connection, links have to be defined for each pairs of ontologies, so it is hardly possible to build up an $\mathscr{E}$-connected NAO from automatic ontology matching techniques.

Local logics: $\mathscr{E}$-connections were originally defined on a more general set of local logics, but later results, algorithms, proofs and practical developments were all defined on networks of description logic ontologies.

Correspondence syntax: correspondences exist in the form of local DL axioms where special relations called *links* appear. Links appear in axioms where roles normally would in role restriction constructs such as $\exists R.C$. Axioms with links are tied to a local ontology, but the links relate them to foreign terms. When a DL construct calls for a role with a concept (such as $\exists$, $\forall$, $\leq n$, $\geq n$), a link can be used instead of the role, together with a concept from a foreign ontology. For instance, $i\!:\!C_i \subseteq \exists R^{ij}.C_j$ indicates a relationship between the term C_i of ontology O_i and the term C_j of O_j.

Contextual interpretation: in addition to a tuple of local interpretations, an $\mathscr{E}$-connection interpretation has a special interpretation that assigns to each link R^{ij} from i to j a domain relation, that is, a subset of $\Delta_i \times \Delta_j$.

Satisfaction of correspondences: since correspondences are essentially DL axioms, they are satisfied in the same way as in DL. However, the difference is in the way concepts constructed from links are interpreted. Concepts with links are interpreted according to the same definitions as in normal DL role restrictions, with the exception that instead of relying on a binary relation over the local domain (that is, a subset of $\Delta_i \times \Delta_i$), they rely on a domain relation (a subset of $\Delta_i \times \Delta_j$).

From a practical perspective, the designers of $\mathscr{E}$-connections provided a set of tools and guidelines to integrate them in the Semantic Web infrastructure [**?**]. Notably, they extended the ontology editor SWOOP to model connections and integrated an $\mathscr{E}$-connections reasoner into Pellet[6], but it no longer supports it.

1.7 Integrated Distributed Description Logics.

Integrated Distributed Description Logics (IDDL [**?**]) is a formalism that addresses similar issues as DDL but takes a different paradigm than other contextual frameworks. Usually, cross-ontology assertions (*e.g.*, bridge rules in DDL, links in $\mathscr{E}$-connections, semantic imports in P-DL) define knowledge from the point of view of one ontology. That is to say that the correspondences are expressing the relations "as witnessed" by a local ontology. On the contrary, IDDL asserts correspondences from an "external" point of view which encompasses both ontologies in relation. One consequence of this approach is that correspondences can be manipulated and reasoned about independently of the ontologies, allowing operations like inversing or composing ontology alignments, as first class objects [**?**].

[6] `http://clarkparsia.com/pellet/`

In terms of model theory, this is represented by using an additional domain of interpretation to the whole network of ontologies, as if it was a single ontology. The local domains of interpretation, assigned to all ontologies, are then related to the global domain by way of the so-called *equalizing functions* (ε_i). These functions map the elements of local domains to elements of the global domain. Formally, a correspondence $i:C \xleftrightarrow{\sqsubseteq} j:D$ from a concept C of ontology O_i to concept D of ontology O_j is satisfied whenever $\varepsilon_i(C^{\mathscr{I}_i}) \subseteq \varepsilon_j(D^{\mathscr{I}_j})$.

A reasoning procedure for this formalism has been defined [**?**], where a central system detaining the correspondences can determine global consistency of a network of ontologies by communicating with local reasoners of arbitrary complexity. This formalism can be used for federated reasoning systems, when the interactions between local ontologies are rather weak. By separating local reasoning and global reasoning, it better prevents interactions between contexts, thus being quite robust to heterogeneity.

Local logics: local logics are description logics.

Correspondence syntax: correspondences take the form of cross-ontology subsumption ($i:C \xleftrightarrow{\sqsubseteq} j:D$), cross-ontology disjointness ($i:C \xleftrightarrow{\perp} j:D$) where C (respectively D) is either a concept or a role, possibly complex, of ontology O_i (O_j respectively).

Contextual interpretation: in addition to a tuple of local interpretations, an IDDL interpretation contains a non empty set Δ called the *global domain of interpretation*, and a tuple of functions $\varepsilon_i : \Delta_i \rightarrow \Delta$ which map elements of local domains to elements of the global domain.

Satisfaction of correspondences: an interpretation $\langle (I_i), (\varepsilon_i), \Delta \rangle$ satisfies a cross-ontology subsumption $i:C \xleftrightarrow{\sqsubseteq} j:D$ whenever $\varepsilon_i(C^{\mathscr{I}_i}) \subseteq \varepsilon_j(D^{\mathscr{I}_j})$ and it satisfies a cross-ontology disjointness $i:C \xleftrightarrow{\perp} j:D$ whenever $\varepsilon_i(C^{\mathscr{I}_i}) \cap \varepsilon_j(D^{\mathscr{I}_j}) = \emptyset$.

1.8 Modular Web Rule Bases

Although this approach is not based on current Semantic Web standards, it is relevant to this survey of formalisms. The framework proposed in [**?**] makes the distinction between global knowledge, local knowledge and internal knowledge. The framework is based on a rule-based language rather than OWL or a Description Logic, and allows one to express and reason modularly over data across the Web. In this framework, each predicate in a rule base is constrained with "uses" and "scope", which in turn determine the reasoning process. It also treats different forms of negation (weak or strong) to include Open-World Assumption (OWA) as well as Closed-World Assumption (CWA) [**?**]. The assumption and type of negation used lead to 4 different "reasoning modes" (specified *s*, open *o*, closed *c*, normal *n*). This rule-based framework provides a model-theoretic compatible semantics and allow certain predicates to be monotonic and reasoning is possible with inconsistent knowledge bases. This framework addresses a few issues of our example scenario because rules can express some DL axioms and can be exchanged with certain restrictions (private, global or local).

Local logics: the local logic define "rule bases" as the local theories, which roughly correspond to 4-tuples of logic programmes (one for each reasoning mode). The authors propose two formal semantics, one based on answer set programming, the other as well-founded semantics.

Correspondence syntax: correspondences take the form of import statements tied to a local rule base r, which abstractly are triples $\langle p, m, i \rangle$ where p is a predicate, m is a reasoning mod (which tells whether weak negation is allowed or if OWA or CWA is used) and i is a set of external rule bases (*i.e.*, $r \notin i$).

Contextual interpretation: there is no particular structure for interpreting import statements. Local interpretations are subsets of a Herbrand base that must satisfy conditions based on the reasoning modes of the predicates.

Satisfaction of correspondences: the notion of model in this formalism requires several formal definitions. To avoid an extensive description, we present the idea informally: intuitively, the meaning of the predicates in a rule base r in reasoning mode m depends on the meaning of the predicates of a rule base r' with reasoning mode m', when r imports terms from r' and the mode m' is "more restrictive" than m (where restrictiveness can be ordered as follows $s < o < c < n$). The detailed semantics is found in Section 4 and 5 of [**?**].

Although the formalism behind modular rule bases form a contextual logic, it has features that do not fit well with open, distributed environments. In particular, a rule base can restrict the way other rule bases describe their knowledge. This would be hard to enforce in a system of autonomous agents.

1.9 Other relevant formalisms

In this section, we discuss other formalisms that are not fitting the general definition of contextual logic provided in Section 1.2 but partially address the problem of multi-contextual, heterogeneous and distributed knowledge. Especially, we avoided the presentation of contextual logics where the knowledge *about* context is mixed with the knowledge *inside* a context, as in the seminal approach of McCarthy [**?**] or Lenat [**?**].

1.9.1 Contextualised Knowledge Repositories

Homola and Serafini [**?**] define a contextual logic where cross-context knowledge does not take the form of correspondences between entities in different contexts, but expresses relationships between the contexts themselves. So they split information in a network of ontologies into a tuple of local ontologies and a structure called "meta knowledge", which define a hierarchy of contexts. Contexts and their relationships are described in a DL ontology which describes how knowledge is reasoned with across context. The formalism also provides different dimensions of context, which makes the approach very close to the work of [**?**].

1.9.2 Contextual RDF(S)

Guha *et al.* [**?**] proposed an extension of RDF(S) to incorporate contextual knowledge within RDF model theory. A simpler version of OWL is assumed to be interoperable with the proposed context mechanism. As opposed to the aforementioned formalisms, this contextual version of RDF does not separate the knowledge of different contexts in distinct knowledge base. On the contrary, context is "reified" such that multiple contexts can be described within the same knowledge base, and context itself can be described. The most basic change in RDFS model-theory introduced by the addition of contexts is that the denotation of a resource is not just a function of the term and the interpretation (or structure), but also of the context in which that term occurs. Most importantly, the proposed context mechanism allows RDF statements to be true only in their context.

1.9.3 Reasoning with Inconsistencies

Robustness to heterogeneity is an important aspect in Agreement Technologies. One of the most problematic consequences of heterogeneity is the occurrence of undesired inconsistencies. Therefore, we believe it useful to investigate formal approaches for handling inconsistencies. There are two main ways to deal with inconsistent ontologies. One is to simply accept the inconsistency and to apply a non-standard reasoning method to obtain meaningful answers in the presence of inconsistencies. An alternative approach is to resolve the error, that is, to repair the ontology or the alignment, whenever an inconsistency is encountered.

Repairing or revising inconsistent ontology is, in principle, a possible solution for handling inconsistency. However, one major pragmatic issue we observe is that some agents may not expose and/or allow repair of their knowledge bases due to various legal or privacy constraints. Also, in a typical Semantic Web setting, importing ontologies from other sources makes it impossible to repair them, and if the scale of the combined ontologies is too large then repair might appear ineffective. Other work focus on revising mappings only [**?**], but they are meant to be used at alignment discovery time, which we are not discussing in this chapter.

Reasoning with inconsistencies is also possible without revision of the ontology. One effective way of tolerating inconsistencies consist of using paraconsistent logics [**?**]. Paraconsistent logics use a "weaker" inference system that entails less formulas than in classical logics. This way, reasoning can be done in the presence of inconsistency. A paraconsistent extension of OWL was proposed in [**?**]. Alternatively, defeasible argumentation [**?**] and its implementation Defeasible Logic Programs (DeLP [**?**]) have been introduced to reason and resolve inconsistencies. In this case, the TBox is separated into 2 subsets, one being *strict*, which means that it must always be used in reasoning, the other being *defeatable*, which means that an argumentation process may defeat them and nullify them for a particular reasoning task.

While we want to tolerate inconsistency when reasoning with an ontology defined in another context, it is not desirable to tolerate local inconsistencies as an agent should normally be self consistent. The system should have a strict logical framework when it only treats local data, that are existing in a unique and well understood context. Unfortunately, the approaches mentioned here are not able to distinguish local knowledge and external knowledge. They do not allow specification of the types of mappings we need, and are not capable of treating policies.

1.10 Discussion

With the development of the Semantic Web where more and more knowledge is made available from multiple sources, and subsequently integrated by Linked Data search engines, the need for taking into account the context of information has been made much clearer. Still, no logical formalism has yet managed to gain enough traction to be integrated in standards. Some researchers debates the qualities of each formalisms and compare them, such as [**?**, **?**] while others prefer to avoid taking the route to contextual knowledge, advocating pragmatic choices in implementations to counter the effect of heterogeneity, incoherence and scale [**?**, **?**]. Yet, formal justifications are needed to help implementers understand why some practical choices are sensible.

A first step towards an agreed formalism for multi-contextual knowledge is Named Graphs [**?**] and the SPARQL notion of a dataset [**?**], which many triple stores implement. While these specifications do not make explicit what inferences are allowed from multiple contexts, they acknowledge the need to separate knowledge into identified subsets.