Context Representation for the Semantic Web

Jie Bao, Jiao Tao, Deborah L. McGuinness
Tetherless World Constellation
Rensselaer Polytechnic Institute
Troy, NY 12180
{baojie,taoj2,dlm}@cs.rpi.edu

Paul R. Smart
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
ps02v@ecs.soton.ac.uk

ABSTRACT

Context modeling is critical for the unambiguous and effective delivery of data and knowledge on the Web, and in particular, the Semantic Web. However, Contexts are often embedded in the application programs or are implied by the application- or community-specific agreements. In this paper, we propose a framework for contexts modeling that can provide formal and explicit representations for contexts on the Semantic Web. The core component of the framework is a new formalism of contexts based on the notion of jurisdictions of knowledge statements. We also discuss the semantic assumption (modeled as institutions) and provenance aspects of contexts.

1. INTRODUCTION

The unambiguous and effective delivery of data and knowledge on the Web relies heavily on the correct representation and understanding of the associated contexts. However, the current way of encoding contexts of data and knowledge on the Web is largely ad hoc. Contexts are often embedded in the application programs or are implied by the application- or community-specific agreements. This makes the linking and reusing of data and knowledge, and thus the integration of Web applications, a difficult problem. Therefore, building the architectural support for contexts is one of the major challenges for the Web, and in particular, for the Semantic Web [31].

While the research of contexts has been extensively discussed in AI (see [2] for a survey), there are still no comprehensive studies on the generic and formal representation for contexts on the Semantic Web. Notable exceptions include C-OWL [12] and a context model for RDF [45]. However, these work only cover specific aspects of contexts, for example, C-OWL addresses only contextual ontology mappings.

In this paper we present a formalism about representing contexts on the Web by extending McCarthy’s seminal work on context formalization [37] (later extended by [38]). Our main contributions are as following:

- McCarthy’s context theory relates a proposition to its context using the is true relation. However, this approach may cause problems when applied to an open system like Web since the truth of a proposition is often not applicable or unknown in a context. Our framework, instead, uses a more generic jurisdiction relation in (‘is in’). Thus, in(c,p) means that a proposition p should be interpreted in the context c (while p is not necessarily true in c). A set of context statements (as a generalized form of “lifting axioms” in [38]) can be used to state the relations between contexts, thus supporting automated reasoning about contexts.

- While [38] does not consider semantic assumptions (e.g., the Open World Assumption) in contexts, our framework allows such assumptions to be explicitly stated as institutions [27], hence effectively avoids the risk of reusing knowledge outside of situations with its intended assumptions.

- In [38], a context is an object without a provenance description (when, where, who, etc.). Utilizing recent advances on provenance representation (e.g., PML [22, 39]), our framework allows associating a provenance model to a context.

In summary, the proposed framework for contexts provides formal and explicit representations for the usually implicit contextual assumptions of data and knowledge on the Web. This is done by supporting the description of logic institutions, relations of contexts, and provenance. Our framework is able to tackle some critical issues for extending Web as a “Social Machine” [31], such as, permitting different views on the same data, faithful knowledge integration and situation awareness.

This paper is organized as follows: In Section 2 we introduce the notion of jurisdiction-based contexts and relations between contexts (e.g., compatibility and incompatibility). In Section 3 we discuss how to represent provenance information of a context and use it in inference with the context. In Section 4 we present how to use the institution theory to formally and explicitly capture semantic assumptions used in a context, and practical approaches to make machine-readable specifications of institutions. Section 5 gives related work in AI and Semantic Web research and Section 6 concludes the paper with future work.

2. FORMALIZING CONTEXTS

In this section we introduce a formal representation of contexts, extending the McCarthy’s context formalization [38] (referred as McCarthy-style contexts thereafter).

2.1 Contexts as Jurisdictions

One key notion in [38] is that contexts are modeled as first class objects. A basic relation between a proposition and a
context in [38] is
\[ \text{ist}(c, p) \]
meaning that the proposition \( p \) is true in the context \( c \).

Lifting axioms can be used to connect contexts. For example, to transfer truth from one context \( c_1 \) to a more general outer context \( c_2 \), we may use a lifting axiom:
\[ \text{ist}(c_1, p) \rightarrow \text{ist}(c_2, p) \]

However, when applied to an open system such as the Web, the McCarthy-style context modeling often can’t rightly capture the context of a proposition (often modeled as a knowledge statement in an ontology on the Semantic Web)\(^3\).

Some typical scenarios include the following:

- The truth value of a statement cannot be meaningfully determined in a context since it is not applicable in that context. For example, the statement “Eric Cartman lives in South Park”\(^2\) is true in the context of the television series “South Park”, but it is meaningless to assert whether this is true or false in other contexts, for example, the context of “The Simpsons” or the context of the real world.

- The truth value of a statement is unknown in the context where it is stated or queried against. This may be because determining the truth value of the statement itself is an open problem (e.g., whether \( P=NP \) in the theoretical computer science), or because of the open nature of the system (e.g., even if there is no real person named “Eric Cartman” currently listed on Wikipedia, answer to the question whether such a person exists is unknown in the context of Wikipedia since Wikipedia does not have complete knowledge about the world).

- The meaning of “true” itself may be different in different contexts. For example, an description logic statement is true against an ontology only if all models of the ontology entail the statement, whereas a statement is true against a logic programming knowledge base if some selected set of models (e.g., minimal Herbrand models) entail it. Thus, the meanings of truth in a DL-based context (e.g., OWL) and a LP-based context are not necessarily the same.

In this paper, we, instead, introducing a new context formalism based on the notion of “jurisdiction” based on the following assumptions about contexts:

- When we say a statement is related to a context, we do not necessarily mean the statement is true in the context, rather, we mean the context has the jurisdiction to interpret the meaning of the statement. If a context has no jurisdiction over a statement, the statement shall not be interpreted in that context\(^3\).

- A context may make explicit description of the semantic assumptions it makes and the precise meaning of truth in that context. We call such a description as the institution of the context.

- A context maybe a rich object that has descriptions about its properties (such as provenances) and relations to other contexts.

For these goals, we extend [38] by introducing a new relation “isin” (is in) to indicate the jurisdiction relation between a statement and a context:

\[ \text{isin}(c, \alpha) \]

where \( c \) is a context of the statement \( \alpha \), i.e., \( \alpha \) can be interpreted using the semantic assumptions made in the institution of \( c \) (see more on institutions in Section 4). Note, in general, we do not have

\[ \text{isin}(c, \alpha) \rightarrow \text{ist}(c, \alpha) \]

However, since a statement is only true in an applicable context, the reverse relation is always true:

\[ \text{ist}(c, \alpha) \rightarrow \text{isin}(c, \alpha) \]

An “isin” relation can be negated or quantified same as other statements. For example,

\[ \neg \text{isin}(c, \alpha) \]

means that \( \alpha \) is not applicable in the context \( c \), thus \( \text{ist}(c, \alpha) \) is not a valid formula. When we see such a relation, either explicitly stated or inferred using the context’s properties and axioms, we will be able to prevent using the statement in a wrong context (“out of context”).

The next statement, using universal quantification, says that if the negation of a statement is a Propositional Logic statement, then it is also a Propositional Logic statement:

\[ \forall \alpha, \text{isin}(pl, \neg \alpha) \rightarrow \text{isin}(pl, \alpha) \]

where \( pl \) stands for the context of Propositional Logic.

Note that in Propositional Logic of Context (PLC) [19], a logic based on the framework of [38], consistency of contexts requires that

\[ \forall \alpha, \text{ist}(c, \neg \alpha) \rightarrow \neg \text{ist}(c, \alpha) \]

This shows a clear difference between the \( \text{ist} \) relation and the \( \text{isin} \) relation.

### 2.2 Context Constructors and Axioms

Similar to lifting axioms in [38], there are many useful relations between contexts that can be captured by context constructors and axioms.

- **Union.** A union constructor “\( \vee \)”, e.g., in

\[ \text{isin}(c_1 \vee c_2, \alpha) \]

means that \( \alpha \) can be interpreted by either \( c_1 \) or \( c_2 \). For example, an OWL DL ontology \( O \) can be interpreted either by the OWL DL semantics (captured by a context \( s_{\text{DL}} \)) or the OWL Full Semantics (captured by a context \( s_{\text{FAll}} \)):

\[ \text{isin}(s_{\text{DL}} \vee s_{\text{FAll}}, O) \]

- **Intersection.** An intersection constructor, e.g., in

\[ \text{isin}(c_1 \land c_2, \alpha) \]
creates a new context that has properties from both c1 and c2. For example, Bart Simpson is a child appeared in both South Park (context "sp") and The Simpsons (context "ts"), therefore we have:

\[
isin(sp \land ts, \text{Child}(BartSimpson))
\]

**Extension.** A context c1 may extend another context c2, denoted as c1 ⇒ c2, such that it inherits institutional properties of c2 (e.g., c2's semantic assumptions), possiby with some additional properties of its own. c1 is said a subcontext of c2 and c2 is a supercontext of c1. Therefore, we have

\[
(c1 ⇒ c2 \land isin(c1, \alpha)) → isin(c2, \alpha)
\]

(if α is interpretable in c1 then it is interpretable in c2)

\[
(c1 ⇒ c2 \land ist(c1, \alpha)) → ist(c2, \alpha)
\]

(if α is true in c1 then it is also true in c2)

The extension relation is transitive, thus

\[
((c1 ⇒ c2) \land (c2 ⇒ c3)) → (c1 ⇒ c3)
\]

We always have that

\[
c1 \land c2 ⇒ c1 \land c2
\]

Note that c1 may not necessarily inherits other properties of c2, e.g., provenances.

**Nesting.** An isin assertion itself can be stated in another “outer” context. For example, “Wikipedia says that in the television series South Park, Eric Cartman lives in the place South Park” can be represented as

\[
isin(\text{wikipedia}, isin(sp, \text{livesIn}(EricCartman, \text{SouthPark})))
\]

where wikipedia is the context of Wikipedia, and sp is the context of the television series South Park.

In general, we assume that there is a universal context c0 as the default outer context of all other contexts, and it can be omitted whenever necessary.

Note that context nesting is different from context extension: an extension transfers truth and jurisdiction of a statement from a subcontext to a supercontext, whereas nesting does not transfer these to the outer context. Therefore,

\[
isin(\text{wikipedia}, \text{livesIn}(EricCartman, \text{SouthPark}))
\]

does not follow from the previous assertion.

**Incompatibility.** Context incompatibility declarations can prevent using a statement out of context. For a context c, ¬c stands for the union of all contexts that are incompatible with c. We have

\[
isin(\neg c, \alpha) ↔ \neg isin(c, \alpha)
\]

Therefore, we can express “c1 is incompatible with c2” as

\[
isin(c1, \alpha) → isin(\neg c2, \alpha)
\]

or in short form as c1 → ¬c2

We can easily see that if a context c1 has a supercontext that is incompatible with c2, then c1 must be also incompatible with c2:

\[
((c1 ⇒ c') \land (c' → ¬c2)) → c1 → ¬c2
\]

In the next section, we will show more examples on inferring context incompatibility based its provenance properties.

**Compatibility.** Compatibility declarations transfer jurisdiction from one context to another context. “c1 is compatible with c2”, denoted as c1 → c2, means that

\[
isin(c1, \alpha) → isin(c2, \alpha)
\]

Compatibility is weaker than extension since it only transfers jurisdiction but not truth of statements.

The set of context constructors and axioms is not meant to be complete. Other useful context description languages may be designed.

### 2.3 Context IRIs and Dereferencing

On the Semantic Web, a context can be represented as a Web resource and identified with an IRI (Internationalized Resource Identifier, a generalization form of URI and URL). A context description document, either formally described as a set of context axioms or informally stated as a human-readable document, may be dereferenced at the IRI. In addition, the dereferenced document may provide other information about the context, such the provenance and institution descriptions of the context (as will be outlined later in the paper).

### 3. PROVENANCES OF CONTEXTS

Contexts in our framework are rich objects such that they may have properties. In particular, a set of common and important properties are the provenance of a context, including the aspects of temporal (when), spatial (where), agent (who), casual (why) and other properties. Recent advances in provenance modeling (e.g., PML [22, 39]) allow us to precisely describe provenances and conduct inference with provenance knowledge4. The goal of this section is not to propose or advocate a particular provenance model for describing contexts – this decision is best left to domain experts who model the contexts. Instead, we demonstrate several typical scenarios when provenance information can leverage inferences about contexts.

**Quoting.** Identifying a context using its provenance information can help us to realize quoting as supported by the N3 Logic [7] and Named Graphs [20]. For example, in the N3 Logic (the modeling using named graphs is similar), we may use quoted formulae such as:

\[
:wikipedia :says{
  :sp :says {
  }
};
\]

In our context model “wikipedia” and “sp” are represented as two contexts (as shown in the last section), and we can attach provenance information to these contexts. For example, we may add the following RDF statements about “wikipedia” and “sp”:

\[
:wikipedia :sourcePage
:wikipedia :date "2010-03-25"^^xsd:date .
\]

Survey lists several survey papers on provenance.
Contexts that are from the Fox News: saying that news from CNN should not be interpreted in the same context, or a context is (in)compatible with another. This separation of the conceptual context and the physical location of a statement offers several advantages:

- It offers a flexible way to describe context. For example, the editors of Wikipedia’s pages about South Park may not be Trey Parker and Matt Stone themselves (thus, they are not the authors of the physical location of the statement), we are still able to correctly credit them as the creators of the South Park context.

- It leverages data integration. Knowledge statements from different graphs may share the same context, then we will immediately know that they can be combined context-safely.

- It allows us to associate multiple contexts to a statement whereas in N3 Logic or named graphs a statement can only reside in one graph. For example, we may associate to the statement livesIn(EricCartman, SouthPark), in addition to sp, also contexts “us” (United States) and “edu” (First Order Logic).

Inferring (In)compatibility. Provenance information may be used in determining whether a statement is applicable in a context, or a context is (in)compatible with another context. For example, we may have the following statements saying that news from CNN should not be interpreted in the contexts that are from the Fox News:

\[ orall c_1, c_2, \text{hasSource}(c_1, CNN) \land \text{hasSource}(c_2, FoxNews) \rightarrow (c_1 \rightarrow \neg c_2) \]

where \text{hasSource} relates a contexts to its source.

The next example shows that annual reports of mutual funds can be put together for comparison only if they are from the same year:

\[ orall c_1, c_2, \exists y, \text{hasYear}(c_1, y) \land \text{hasYear}(c_2, y) \land \text{AnnualReport}(c_1) \land \text{AnnualReport}(c_2) \rightarrow (c_1 \rightarrow c_2) \land (c_2 \rightarrow c_1) \]

Selective Importing. The current importing mechanism in OWL has the implicit assumption that all ontologies in the importing transitive closure are compatible to other each such that a union of them can be made and the importer ontology is interpreted under this union. However, this is not always desired as importing is often used as citation whereas the “cited” content is not necessarily always true in the importer’s context (e.g., in the case of citing for critique). Also, since importing can be transitive but the trust of content is not always transitive, using an indirectly imported ontology may result in unintended consequences.

Using the context mechanism, we will be able to realize selective importing such that only trusted ontologies in the importing transitive closure is taken into account. For example, the next strategy says that only ontologies published by an educational organization (with domain name “edu”) should be used. We assume, if an ontology does not declare a context where it is in, it uses a default context with a property \text{iri} which is the ontology’s \text{iri}; \text{imports} is the importing transitive closure relation; \text{topDomain} is a function that parses the top-level domain of an \text{iri}.

\[ \text{isin}(c, O) \land \text{imports}^+(O, O') \land \text{isin}(c', O') \land \text{iri}(c', x) \land (\text{topDomain}(x) == "edu") \rightarrow \text{isin}(c, O') \]

4. INSTITUTIONS OF CONTEXTS

4.1 Modeling Semantic Assumptions Using Institutions

The \text{isin} relation does not specify whether a statement is true in a context. This is left to be validated using the institution of the context, i.e., the set of semantic assumptions made in the context. The term “institution” is adopted under the influence of the institution theory [27] which specifies a framework for abstract description of a logic system. Note that a description of an institution does not necessarily be a formal or machine-readable description (while this is desired and more details are discussed in the next subsection). For example, N3 Logic [7] currently does not have a formal semantics; the OWL syntax and semantic document [41] gives a formal yet only human-readable description for OWL [40] and is currently implicitly used as the default institution document for OWL ontologies. In particular, if a context is not logic-dependent, it may have no associated institution.

Basically, an institution description specifies the following common aspects of a logic system:

- A signature, i.e., some vocabulary that can be used to construct sentences;
- A sentence generation grammar using the signature;
- A mapping from the signature to the models of the institution;
- A satisfaction relation between sentences and models.

These can be formally describe based on the category theory as given in Def. 1 (for basic notions in the category theory, cf. [5]):

Definition 1: ([27] Definition 1) An institution I consists of

1. a category \text{Sign}, whose objects are called \text{signatures},
2. a functor \text{SEN} : \text{Sign} \rightarrow \text{Set}, giving for each signature \Sigma \in \text{[Sign]} a set of \text{sentences} \Sigma \subseteq \text{[Sign]} of the signature,
3. a functor \text{MOD} : \Sigma \rightarrow \text{Cat}^{op}, giving for each signature \Sigma \in \text{[Sign]} a category whose objects are called \Sigma-models, and whose arrows are called \Sigma-(model) morphisms,
4. a relation \equiv_{\Sigma} \subseteq \text{MOD}(\Sigma) \times \text{SEN}(\Sigma) \text{ for each } \Sigma \in \text{[Sign]}, called \Sigma-satisfaction.
such that for each morphism \( \phi : \Sigma \to \Sigma' \) in \( \text{Sign} \), the Satisfaction Condition holds:

\[
m' \models_{\Sigma'} \text{SEN}(\phi)(e) \iff \text{MOD}(\phi)(m') \models_{\Sigma} e
\]

For example, OWL DL is an institution where the signature is the set of all named classes, properties and individuals; sentences are constructed following the grammar given in [41]; its models are the set of all first-order models; the satisfaction relation is given in [41] under the open world assumption and non-unique name assumption. This institution can be used as the default institution of an “OWL DL” context which handles all syntactically valid OWL DL ontologies. Similarly, we may have the RDFS institution and the OWL Full institution.

The institution theory also allows us to formally describe syntactical and semantic relations between two different institutions using institution morphisms:

**Definition 2:** ([27] Definition 32) Let \( \text{I}_1 = (\text{Sign}_1, \text{SEN}_1, \text{MOD}_1, \models^1) \) and \( \text{I}_2 = (\text{Sign}_2, \text{SEN}_2, \text{MOD}_2, \models^2) \) be two institutions. An institution morphism \( \Phi : \text{I}_1 \to \text{I}_2 \) consists of

1. A functor \( f : \text{Sign}_1 \to \text{Sign}_2 \)
2. A natural transformation \( \alpha : f \circ \text{SEN}_2 \Rightarrow \text{SEN}_1 \)
3. A natural transformation \( \beta : \text{MOD}_1 \Rightarrow f^{\text{op}} \circ \text{MOD}_2 \)

such that the following Satisfaction Condition holds:

\[
m \models^1 \alpha e \iff (f \circ \beta)(m) \models^2 \beta e
\]

for any \( \Sigma \)-model \( m \) from \( \text{I}_1 \) and any \( f(\Sigma) \)-sentence \( e \) from \( \text{I}_2 \).

For readers who are not familiar with the category theory, the above definition can be understood as a systematic mapping of the vocabularies, the sentence grammars and the models between the two institutions under some satisfaction preservation conditions. For example, the RIF RDF and OWL Compatibility document [23] describes an embedding of RDF in RIF, while it does not directly use institution theory notions, covers the above aspects of institution mapping using a formal, human-readable description.

Associating an explicitly stated institution to a context brings several benefits:

- It separates the making of a statement and that asserting the statement is true, which is useful to enable quoting, citing without supporting and expressing uncertainty, just for a few examples.
- It enables us to connect knowledge bases stated in different logics or using different semantic assumptions.
- It may better guide tools to process the ontology in an intended way. For example, one may publish an syntactically valid OWL-DL ontology but wish tools to use it as an OWL Full ontology (e.g., instead of performing tableau-based inference, perform a rule-based inference for the ontology); this can be realized by associating the OWL Full institution to the context of the ontology.


- It facilitates extending an existing institutions in a modular and explicit way. For example, one may want to use OWL-DL but also use the unique name assumption (UNA); this may be realized by defining an UNA context and institution, and then declare that the ontology uses an intersection context of the OWL-DL context and the UNA context.

### 4.2 Machine-readable Institution Specifications

While most of institution descriptions on the Semantic Web is formal, few of them is machine-readable. A machine-readable definition of the syntax and semantics of an institution will greatly facilitate the automatic validation and processing of contexts that use the institution. A notable exception is RIF-BLD (Basic Logic Dialect) [8], which, as a specialization of RIF-FLD (The Framework for Logic Diagrams) [9], is provided with a formal syntactic, semantic and XML serialization framework. In fact, RIF-FLD offers a comprehensive framework to specify “both syntax and semantics...through a number of mechanisms that are commonly used for various logic languages” [9]. RIF-FLD has the following main components:

- Syntactic framework, which defines the formal presentation syntax of the logic, e.g., alphabet, symbol space and well-formed formulas;
- Semantic framework, which specifies a model-theoretical semantics of the logic and relates the syntax to models;
- XML serialization framework, which defines the general principles for specifying concrete XML-based syntaxes of the logic.

We believe that the generic nature of RIF-FLD makes it an ideal candidate for creating machine-readable institution specifications. This has been evidenced by that various non-official dialects have defined using this framework, e.g., RIF-URD (Uncertainty Rule Dialect) [48] and RIF-CASPD (Core Answer Set Programming Dialect) [32].

Another potential approach to design an institution specification language is to extend Common Logic with a machine-readable syntax. CL also allows the definition of a variety of different dialects, including FOL and even higher-order logics.

### 5. RELATED WORK

Contexts have been extensively studied in AI and other fields. We can only discuss the most relevant work here due to page limits. For surveys on contexts in AI, cf. [14, 15, 1, 35] and for a survey on context representation for the Semantic Web pre-2004 cf. [10]. The Context Conference is also a good source about context research.

#### 5.1 Contexts in AI

**McCarthy-style Contexts:** Our work extends McCarthy’s work on context modeling [37, 38] by separating the notions of truth-based contexts and jurisdiction-based contexts. We have shown that such a separation is useful for Semantic Web applications where truth of a statement is often unknown or meaningless when it is applied in a wrong context.

6http://www.common-logic.org

7http://www.informatik.uni-trier.de/~ley/db/conf/context/index.html
Guha [28] has extended McCarthy’s notion of context and applied it in building the CYC system [34]. Knowledge statements in CYC are divided into microtheories which serve as the contexts of the statements. A microtheory, similar to contexts in [38], is an object that has a name and can be organized in a microtheory hierarchy. Guha’s approach shares the same limitations as McCarthy’s to be used for Semantic Web.

Buvac and others\(^8\) applied McCarthy’s framework in various logics, leading to the study of Propositional Logic of Context (PLC) [19, 17] and Quantification Logic of Context [16, 36]. Buvac showed that the “ist” relation is essentially a modal operator, and a formal semantics and calculus in the modal logic fashion can be given for McCarthy-style contexts in propositional logics and quantificational (predicative) logics. In fact, Buvac and Mason [18] have shown PLC can be reduced to propositional multi-modal logic. Buvac’s approach requires all contexts to be in the same institution whereas this is not required in our approach.

**MultiContext Logics.** Multicontext logics\(^9\) [26] is a family of logics based on two principles:

- Locality - In inference, only partial knowledge is available and this part is call the context for the inference process.
- Compatibility - There is compatibility among the reasoning performed in different contexts.

A multi-context system (MCS) is formally described using the Local Models Semantics [25] and its proof theory (a generalization of natural deduction) is composed of internal rules (for intra-context inference and bridge rules (for inter-context inference). Multicontext logics influenced Distributed First Order Logics [24] and Distributed Description Logics (DDL) [11], with the latter has a close relation to Semantic Web applications.

Bouquet and Serafini [13, 42] have compared multicontext logics and McCarthy-style contexts and have shown that lifting axioms are special forms of bridge rules in MCS, and in general, McCarthy-style contexts are less expressive than the multicontext logics.

Applying the multicontext logics in the Semantic Web, while is conceptually straightforward, has some practical issues in specifying compatibility relations between contexts. Two well-known problems are that knowledge can not be transitive reused in DDL and bridge rules offer only limited expressivities to relate ontologies since they require disjoint vocabularies in different ontologies [4].

**Contexts based on Situation Theory.** Akman and Surav [3] used Situation Theory [6] to model contexts. A situation is a limited viewpoint of the world from an agent. A context is a situation that contains a set of ground knowledge assertions (called “infons” in the theory) and constraints for deriving new facts. McCarthy-style context assertions can be represented as infons such as:

\[
\text{ist}(c, p(x)) \Rightarrow \ll p.c, x, 1 \gg
\]

where \(p.c\) is a new predicate (context \(c\)’s version of \(p\)), and “\(1\)” means \(p.c(x)\) is true in the situation.

`http://www-formal.stanford.edu/buvac/
9http://www.dit.unitn.it/~context/

The situation theory based approach may face an explosion blow-up of newly introduced predicates when there are many predicates and contexts. In addition, the lack of a natural deduction system in [3] is a critical limitation for its practical use.

### 5.2 Context Modeling in Semantic Web

Guha and others [29] applied McCarthy-style contexts for Semantic Web use, targeting aggregation of independently published data. This proposal shares some common properties with our framework, such that each document has a context which can be identified by an IRI/URL. It provides a small ontology to describe contexts, and an alternative approach to provide an extended RDF model theory for introducing contexts.

Stoermer[44, 45] also applied McCarthy-style contexts for RDF context management. For this purpose, this work introduces a Context Relations Ontology (CRO) without adding new language features to RDF (which is different from [29]).

Since both [29] and [44, 45] are based on the ist relation, inherent limitations of the McCarthy-style contexts also present in these proposals.

C-OWL [12] is an extension of OWL based on Distributed Description Logics. The expressivity of C-OWL is limited to be used largely as an ontology mapping language. For example, an “into” bridge rule can be used to establish subclass-like relationship between classes in two ontologies. In addition, since a context is not an object in C-OWL, one can not assert provenance or other properties about a context.

Several other researchers have applied context modeling in semantic desktops [30], information sharing and privacy protection [21], data integration [46] and semantic wikis [33, 47]. These works’ goals are not providing a generic context modeling framework, which is the focus of our paper.

### 6. CONCLUSIONS

We investigated extending McCarthy’s context framework to address context modeling on the Semantic Web. We show that by introducing the notion of context jurisdiction, we can model many scenarios that are not possible using McCarthy’s original framework, e.g., citation without supporting, asserting statements with unknown truth values, and preventing interpreting statements in wrong contexts. We showed that by using provenance information of a context we are able to realize quoting, incompatibility inference and selective importing. We also discussed making (usually implicit) semantic assumptions used in an ontology explicit as logic institutions and presented practical approaches to specify an institution.

This paper is only the first step towards an expressive context modeling framework. Many details of the framework are to be investigated in the future, including the following:

- A formal semantics to specify the precise meaning of the \(\text{isin}\) relation and to derive a natural deduction system for the framework.
- A concrete syntax for expressing contexts that can work with RDF and OWL.
- An investigation into how to use policy languages to manage multiple applicable contexts of an ontology and to realize situation-aware context selection.
7. REFERENCES

[32] Stijn Heymans and Michael Kifer. RIF Core Answer...


