A framework for context-sensitive metadata description

Oleksiy Khriyenko*

Industrial Ontologies Group,
Agora Center, University of Jyväskylä,
P.O. Box 35, FIN-40014, Jyväskylä, Finland
E-mail: olkhriye@cc.jyu.fi
*Corresponding author

Vagan Terziyan

Industrial Ontologies Group,
MIT Department, University of Jyväskylä,
P.O. Box 35, FIN-40014, Jyväskylä, Finland
E-mail: vagan@it.jyu.fi

Abstract: Expectations regarding the new generation of Web depend on the success of Semantic Web technology. Resource Description Framework (RDF) is a basis for explicit and machine-readable representation of semantics. However RDF is not suitable for describing dynamic and context-sensitive resources (eg. processes). We present the Context Description Framework (CDF) as an extension of the RDF by adding a 'TrueInContext' component to the basic RDF triple ('subject-predicate-object'), and consider contextual value as a container of RDF statements. We also add a probabilistic component, which allows multilevel contextual dependence descriptions as well as presumes possibility for Bayesian reasoning with the RDF model.

Keywords: Semantic Web; context-sensitive metadata description; probabilistic context; contextual extension of RDF.

Reference to this paper should be made as follows: Khriyenko, O. and Terziyan, V. (2006) 'A framework for context-sensitive metadata description', *Int. J. Metadata, Semantics and Ontologies*, Vol. 1, No. 2, pp.154–164.

Biographical notes: Oleksiy Khriyenko is a PhD student with the Department of Mathematical Information Technology, University of Jyväskylä, Finland. He has been a member of the 'Industrial Ontologies Group' research group since January 2003 and a researcher in the Agora Centre (University of Jyväskylä). His research interests include: artificial intelligence, Semantic Web, agent technology, web services, distributed resource integration, context aware adaptive environments and the industrial application of these and new mobile technologies (applications and services).

Vagan Terziyan received the Doctor of Technical Sciences (Dr. Habil Tech. equivalent) at the Software Engineering Department in 1993 from the Kharkov National University of Radioelectronics. He has been acting as Professor of Software Engineering since 1994 and as Head of the AI Department since 1997. His research and teaching profile is design of intelligent web applications, which use and integrate emerging knowledge-, agent-, machine-learning- and Semantic Web-based technologies and tools. Recently he has been working in the Agora Centre (University of Jyvaskyla) as Project Leader within the SmartResource TEKES Project, and he is the Head of 'Industrial Ontologies Group'.

1 Introduction

The amount of data within the World Wide Web is increasing continuously. That's why it is becoming more and more difficult to retrieve relevant information by using the current search engines that are based on pattern matching. The Semantic Web approach is intended to solve the problem by annotating resources and enabling semantic search engines. The key issue is that machines will be able to 'understand' the content of resources not only at the syntactic but also at the semantic level. To standardise such annotations, the RDF is used by the W3C consortium as a framework for managing metadata on the web and as a basis for further Semantic Web languages, technologies and tools. The emergent RDF is expected to enable metadata interoperability across different communities and applications by supporting common conventions on metadata syntax, structure, and semantics. RDF data can be regarded as a set of atomic sentences, each having a subject, a predicate and an object. These sentences are also called RDF statements or triples. Systems and tools for managing metadata repositories of RDF triples already exist.

However, storing triples without being able to track back to their original source (producer of the statement) or denote the condition under which it was true is not sufficient for many applications. Especially in RDF, which provides possibility for everybody to say anything about everything, it is mandatory for the users to know the context of the given information (source, time, place and any other contextual identifier). In the absence of this essential data, contradictive statements collected from a variety of sources can occur in RDF repositories, and users are not able to determine which ones they can trust. One possibility for making the RDF model more reliable in modelling context information is to use the RDF reified statements (statements about statements, possible in RDF syntax). MacGregor and Ko (2003) point out that this solution is not practical. The main reasons are that

- it results in a blow-up of needed triples
- it is difficult to read
- it is difficult to write queries to extract relevant material
- it is much more difficult to handle and therefore less efficient.

Another problematic issue is how to determine a reasonable definition a context that is useful within RDF. There are quite a few definitions for a context. In Joseph et al., for example, the definition for context reads:

> "The part of a text or statement that surrounds a particular word or passage and determines its meaning. The circumstances in which an event occurs; a setting." (Joseph et al., 2000)

However encompassing this explanation may be, there is still no clear and universally accepted definition for context in the area of knowledge base systems. An overview of existing interpretations of the term context in the area of knowledge base systems can be found in Jansen (1993). Regarding to Cyc technology[1], some issues concerning context description in knowledge base were resolved via Cyc Microtheory concept, which is an abstract informational thing that represents a context in Cyc.

We have at least three different situations where the term context is used in RDF. First, the context given by the surrounding graph; it is an internal context. The way how to handle and interpret this internal context is mainly discussed in Hayes and McBride (2004). In the second situation, it is an external context, such as source information, time of creation, name of the author and much more, which normally are not included in the RDF model itself, though they could be. Finally, context used to identify triples for a clear and easier handling of sets of triples, e.g., to merge/unmerge graphs (since this identification is not coming from inside the RDF model) (Tolle and Wleklinski, 2004).

A common argument against quads (or adding a fourth 'context' component to the existing RDF 'subject-object-predicate' triple) goes "We have triples; you want quads, where is it going to stop? Quintuples? Sextuples?" The answer, e.g., in MacGregor and Ko (2003), is, quadruples is all you need (this is a well-educated guess). Some RDF systems (Jena's RDB model is an example) internally implement a quad structure that adds a model column to the subject/predicate/object columns. This allows mapping a model to a set of statements. It might seem that adding contexts to their quads would turn their quads into quintuples. Although one could add a fifth context column, a better solution is to convert the models column to a context column and adopt the convention that each context belongs to exactly one model. That way, we have quads, and we can also directly map each statement to a model through its associated context. We agree with that argumentation from MacGregor and Ko (2003) and also think that a triples-plus model architecture can be converted to a quad architecture with no significant increase in storage requirements.

In this paper we represent a logical extension of the RDF to CDF. RDF is a basis for higher levels of computational semantics (OWL), and that is why we decided to start to make an extension on a lower than OWL level first. And again, first of all we aimed to extend resource (metadata) description language, and not an ontology representation. We were not fully satisfied by the RDF reification mechanism, because in our tasks we have to consider a context exactly in associating with a subject statement. And even more, here we have to deal with a restricted range of contextual properties for subject statement's predicate. Just because of these, we think that a quadruple for a statement representation would be an appropriate solution.

In Section 2, we specify the triple property description approach and define the CDF quadruple for a statement representation. Section 3 describes real implementation of the CDF approach. In an Appendix we call attention to a simplified ('lite') version of the CDF Vocabulary Description Language (CDF Schema Lite) as an extension of the existing RDF Schema.

2 Context Description Framework (CDF)

2.1 *New vision of a statement and a property representation*

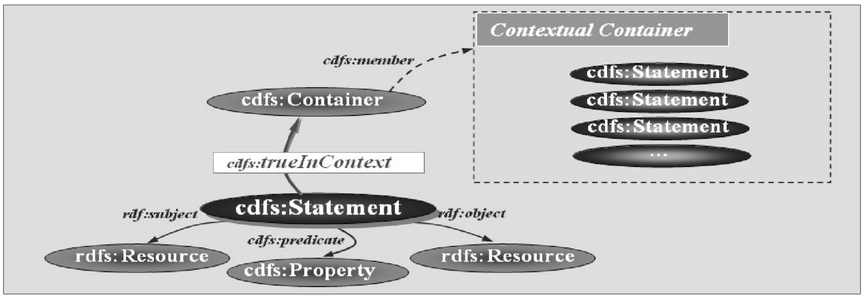
In our vision all properties have some sense in a certain context which should be specified by the context tolerance range. Thus we have a need to define a contextual range for a property, which plays the role of a statement predicate. Such approach to the property definition brings a new vision of a statement representation. Each statement may be true or false concerning the different conditions of an environment.

In this case we consider the context of a statement as a set of other statements that describe a certain condition (state) of an environment. Such descriptions among properties of an environment may contain also the source of the statement descriptions, and thus provide opportunity to manage trust in distributed systems. Each contextual statement itself may also have its own context (i.e., a nested context). A nested context provides new possibilities for vertical in-depth reasoning based on context-sensitive descriptions. We found out that using a triplet-based model for a statement-in-context description is not suitable, and therefore use quadruples for modelling, where the fourth component is a container of contextual statements.

With a goal to not contradict much of the existing standards, we have elaborated a contextual extension of a statement in the RDF. A CDF is a logical extension of the RDF and is meant to model the context dependence of the world properties. It allows us make two significant steps in the resource description approach. We logically go from a duplet (domain-range) vision of a property description in ontology to a triplet description (domain-range-context), and from a triple representation of a statement to quadruple representation (statement in a context of other statements).

Concerning the second significant step (the extension to a quadruple statement representation), we define a CDF quadruple (see Figure 1). A CDF quadruple contains four components: a subject, which is an RDF URI reference or a blank node; a predicate, which is an RDF URI reference; an object, which is an RDF URI reference, a literal or a blank node; and a contextual container (context), which is an RDF URI reference or a blank node. A CDF quadruple is conventionally written in the following order: subject, predicate, object, contextual container. A predicate is also known as the property of a quadruple. With a purpose to define a CDF quadruple we have inherited the rdf:Statement class and have added the additional cdfs:trueInContext property. A CDF statement is a statement made by a token of a CDF quadruple. A subject of a CDF statement is an instance of rdfs:Resource identified by the subject of the quadruple. A predicate of a CDF statement is an instance of cdfs:Property identified by the predicate of the quadruple. An object of a CDF statement is an instance of rdfs:Resource identified by the object of the quadruple. A context of a CDF statement is an instance of cdfs:Container identified by the contextual container of the quadruple. The cdfs:trueInContext property has the cdfs:Statement and cdfs:Container classes as the domain and range accordingly, where the cdfs:Container class is inherited class from the rdfs:Container and restricted with a content. The instances of the cdfs:Container class may contain only the instances of the cdfs:Statement class, which play a role of a statement context. Figure 1 shows a quadruple approach to statement representation. At the time we create the cdfs:trueInContext property, we also add a similar cdfs:falseInContext property to describe the context within which the subject statement is false. Now we can describe any statement with a binding to a context. Such a context-dependent representation of a statement entails a specification of the contextual container content range accordingly to a quadruple predicate. Thus we come to the necessity of making one more logical step in the resource description approach and go to a triple vision of a property.

Figure 1 A quadruple vision of the statement



Following the first step we extend an existing rdf:Property, which is described by rdf:domain and rdf:range, with a crdfs:context description (exactly with a 'context tolerance range' definition). As the RDF Concepts and Abstract Syntax specification (Klyne and Carroll, 2004) describes the concept of an RDF property, we describe the concept of a CDF property as a context-dependent relation between the subject resources and the object resources. A CDF triple property representation contains three components: a domain, which refers to a domain class; a range, which refers to a range class; and a context, which refers to a set of the contextual properties (context range). Figure 2 shows a new triple vision of a property. As a rdf:domain property defines a restricted area (rdfs:Class) of the subject property domain and a rdf:range property sets a subject property range (rdfs:Class), cdfs:context property defines a vector of the properties (cdfs:ContextContainer) that play role of a subject property context.

Class cdfs:ContextContainer is a subclass of the rdfs:Container in a general case. It contains a set of the cdfs:Property instances. They restrict the number of properties that can be used as the objects of a cdfs:predicate property in a contextual statement description. In other words, this container specifies a range (set of the object properties) for the cdfs:predicate properties of the statements in contextual container of the subject Statement (Figure 3).

Figure 2 A triple vision of the property

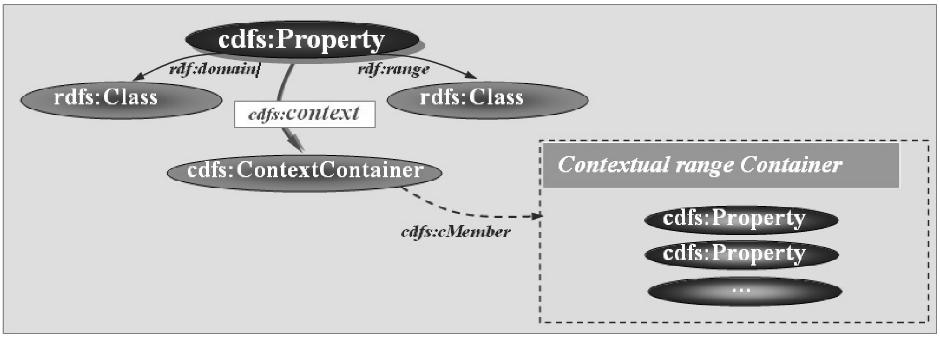
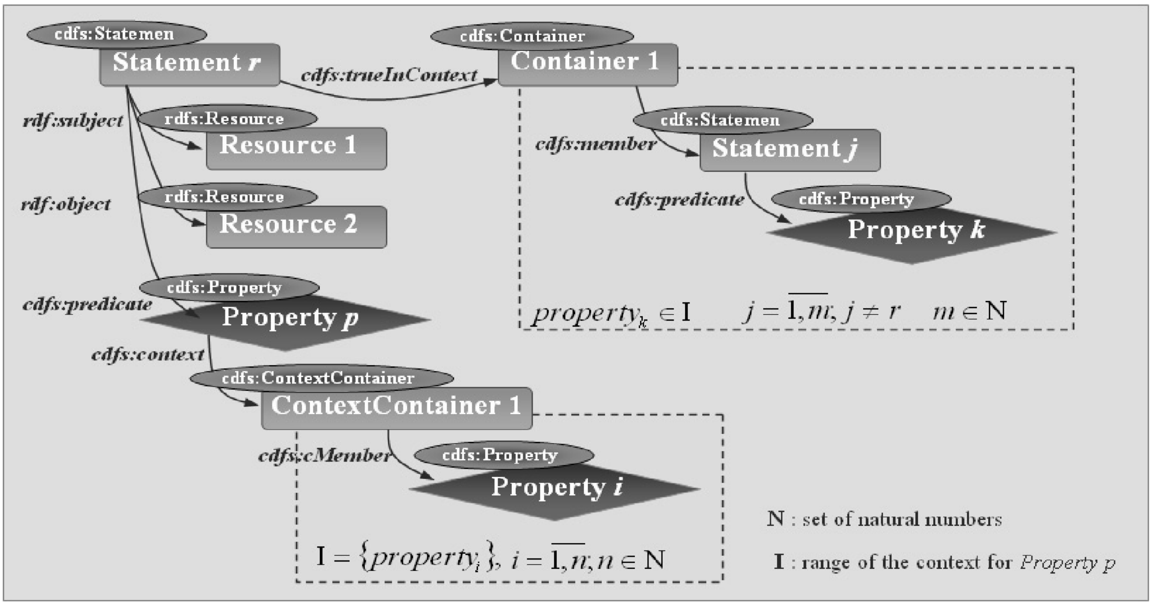


Figure 3 Context tolerance range definition



Due to the new vision of a resource description, we redefine the concept of a subproperty. The cdfs:subPropertyOf property may be used to state that one property is a subproperty of another one. If a property P is a subproperty of property P', then all triplets of resources (subject resource, object resource, and trueInContext container) that are related by P are also related by P'. The term super-property is often used as the inverse of subproperty. If a property P' is a super-property of a property P, then all triplets of resources that are related by P will be also related by P'.

Three rules correspond to the subproperty definition. In the same way as in the RDF specification, the domain and range classes of a subproperty should be the same classes or subclasses of the super-property domain and range classes. Additionally, the subproperty context (vector of the properties) should be covered by the context of the super-property. It means that each element of the subject property context vector (property) should be a subproperty of some super-property context vector element or a new property (is not presented in super-property context vector) (Figure 4).

Let us consider some example from an industrial domain. There are two devices, D#1 and D#2, (they are instances of exmpl:Device class), where D#2 is atomic resource and part of D#1. Additionaly D#1 is part of E#1 (an instance of exmpl:Environment class). The exmpl:Device and exmpl:Environment classes are subclasses of the rdfs:Resource class. A hierarchy of relational properties is represented in Figure 5. The figure shows a simple hierarchy of measurement and condition properties. Specifics of the measurement properties are based on the partOf relation of the resources. The measurement of an atomic resource is a physical measurement. That is why the context of the statement that describes a physical measurement is a statement about specific partOf relation of the subject statement.

Figure 4 Definition of the subproperty concept

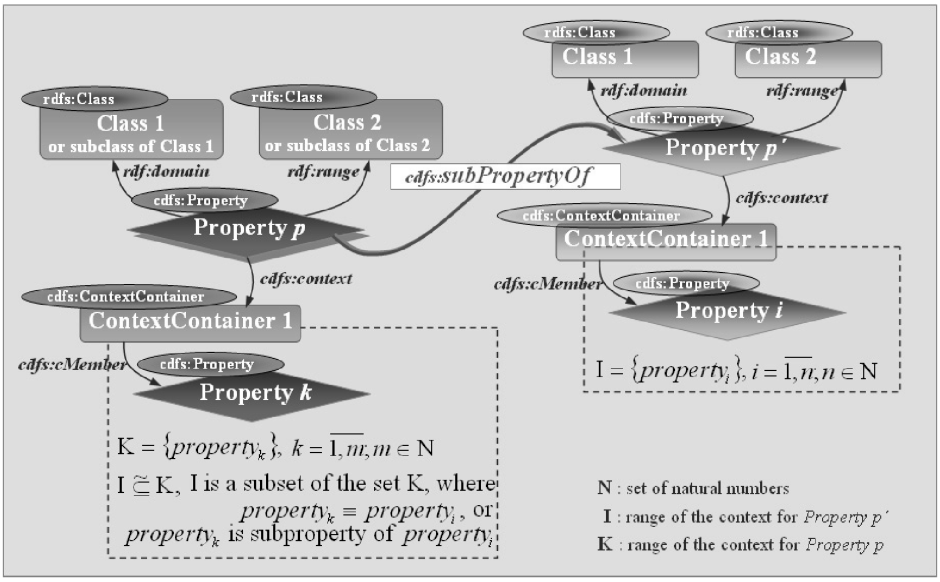
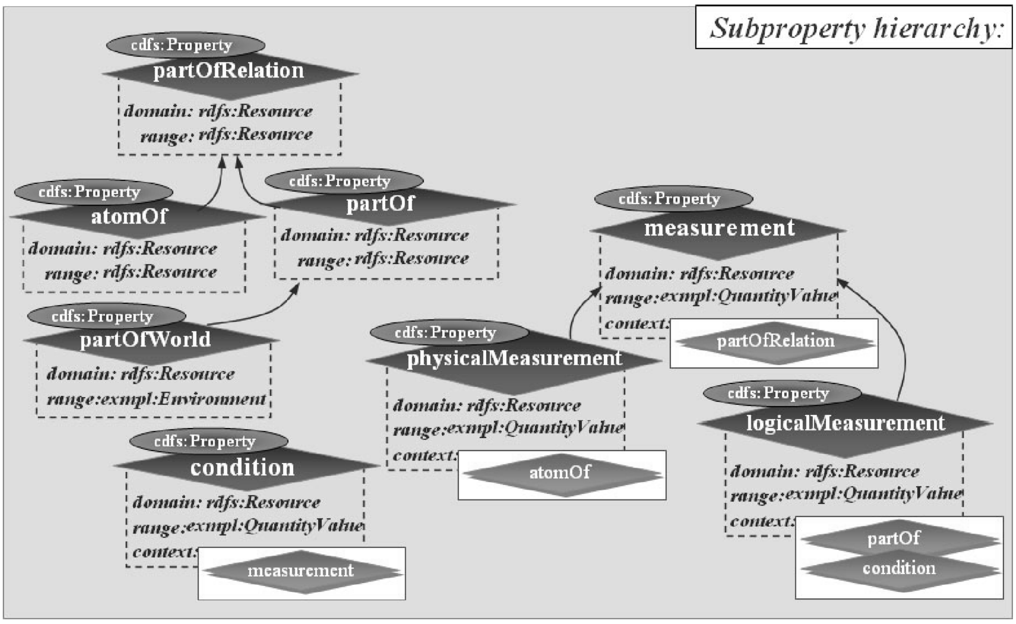


Figure 5 Subproperty hierarchy

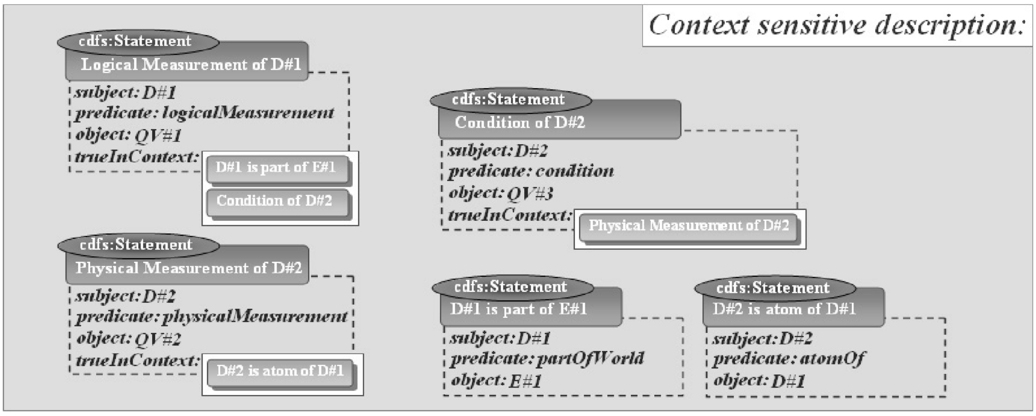


A logical measurement has a slightly different meaning. Logical measurement of the 'mother' resource is based on the 'daughter' resource condition (which is based on own measurement). In this case the context of the statement that describes a logical measurement is the statements about the partOf relation of the subject statement (but not the atomic relation) and condition that forms the basis for this measurement. The context for the statement about the resource condition is the statement about the subject resource measurement. The values of the measurements and condition we will consider as the instances of the exmpl:QuantityValue class of the values: QV#1, QV#2, QV#3.

Based on this set of the properties, we can describe the partOf relation between the resources, the measurements and condition with all necessary context relations accordingly to CDF Schema. Figure 6 graphically shows the description.

From Figure 6 we see that the statement about logical measurement of D#1 is true in context of other two statements: that D#1 is part of E#1 (which totally fits to the context restriction for the logicalMeasurement property) and the second statement is about a condition of the daughter resource D#2 (that D#2 has condition QV#3), which is basis for the value of this logicalMeasurement. Here we have a nested context, because the condition property itself has its own context and statement about the D#2 condition, which is true in context of the statement about the D#2 physical measurement. Such nesting can be performed until reaching the atomic statement, which has property with undefined context.

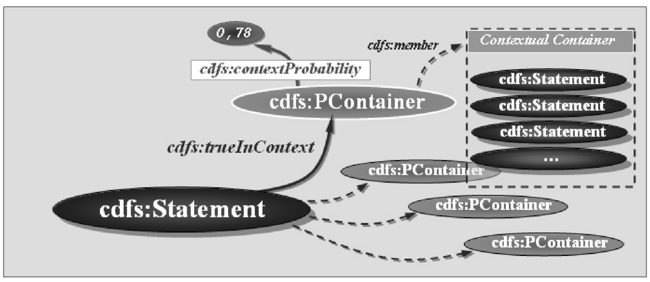
Figure 6 Context-sensitive description



2.2 Context probabilistic model

As was mentioned earlier, each statement can be true or false within certain context, which is a set of contextual statements. The number of contexts (contextual containers) is not limited by any one context. A statement can have a set of contexts that make it true or false. But we can not expect that each of these contexts makes this for 100% sure. It seems reasonable to define a probability of a statement to be true in each possible context for this statement. With this aim, we have defined cdfs:PContainer as a subclass of cdfs:Container and extended the number of properties of this class with a cdfs:contextProbability property (Figure 7). Now we can specify a probability value (between 0 and 1) for each contextual container of the subject statement. It gives us a possibility to build a probabilistic model on a top of this and to enable probabilistic reasoning based on it. However, even after extending a context-dependent resource description with a probability value, we still need one more element. Since the 'significances' (relevancies) of the contextual properties might differ from one other and the significance of the property depends on a certain context, then we have to model these as well. And it will be possible to define probabilistic significance of contextual properties via utilisation of the same CDF approach.

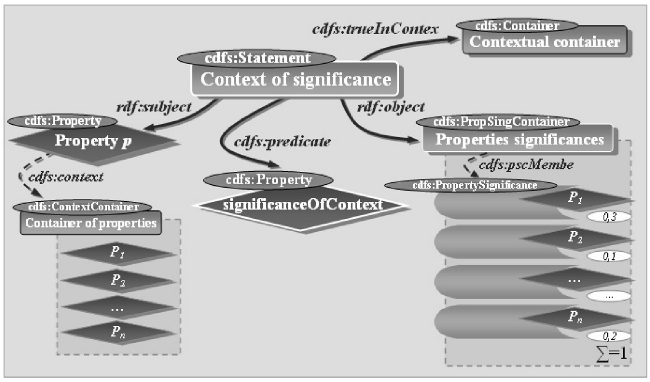
Figure 7 Probability of the statement context



Firstly we define the cdfs:PropertySignificance class. An instance of this class sets the significance of the subject property via the cdfs:subjectProperty (refers to the subject property) and cdfs:pSignificance (with a value between 0 and 1). Then we define a property to be able to specify a significance of the subject property context. The cdfs:significanceOfContext property points to a cdfs:PropSignContainer class instance – a container of cdfs:PropertySignificance instances (contextual properties with correspondent significances for the subject property [cdfs:Property instance]).

And finally we can create a statement that defines the significance of the contextual properties dependent upon a certain context (Figure 8).

Figure 8 Context dependent significance of the contextual properties



In many cases probabilistic models of the contextual reasoning might not be enough and should be enhanced by fuzzy models. From that point of view there still room for improving CDF based on related work on Fuzzy Description Logic. Existing contribution in that area, which is based on the earliest research of Straccia (2001), has been done by Miguel-Angel Sicilia and Elena Garcia-Barriocanal. They have been proposed fuzzy Description Logics (fDLs) (Sicilia and García-Barriocanal, 2004) as an extension to conventional Description Logics (DLs) (Tresp and Molitor, 1998) to handle with uncertainty and imprecision in a numerical way. DLs, as a logical reconstruction of the frame-based representation languages aims to provide a simple declarative semantics to capture the meaning of the features of structured representation of knowledge. The mentioned effort targets a pragmatic way of extending current tools and interfaces for DL-based applications with a fuzzy processing layer for some specific scenarios of uncertainty handling associated to ontologies.

3 Implementation

CDF was elaborated during the first year of the 'Smart Resource Project'[2] by Industrial Ontologies Group (IOG)[3] and successfully applied to the dynamic and context-sensitive industrial data description.

The main objective of the Industrial Ontologies Group is to contribute to fast adoption of Semantic Web and related technologies to local and global industries. It includes research and development aimed to design a Global Understanding Environment (GUN) as next generation of web-based platforms by making heterogeneous industrial resources (files, documents, services, devices, business processes, systems, organisations, human experts, etc.) web-accessible, proactive and cooperative in a sense that they will be able to automatically plan own behaviour, monitor and correct own state, communicate and negotiate among themselves depending on their role in a business process, utilise remote experts, web-services, software agents and various web applications. Three fundamentals of such platform are Interoperability, Automation and Integration. Interoperability in GUN requires utilisation of Semantic Web standards, RDF-based metadata and ontologies and semantic adapters for the resources. Automation in GUN requires proactivity of resources based on applying the agent technologies. Integration in GUN requires ontology-based business process modelling and integration and multi-agent technologies for coordination of business processes over resources. For more details about GUN, see Terziyan (2003, 2004) and Kaikova et al. (2004).

The SmartResource project in its research and development efforts analyses GUN decomposing it into three frameworks:

- *General Adaptation Framework (GAF)*, for Interoperability (1st project year – 2004). GAF provides a framework to describe domain resources (declarative knowledge). It includes Resource State/Condition Description Framework (RSCDF), appropriate RSCDF-based domain ontology, appropriate RSCDF Engine and the family of so called 'Semantic Adapters for Resource' to provide an opportunity to transform data from a variety of possible resource data representation standards and formats to RSCDF and back.

- *General Proactivity Framework (GPF)*, for Automation (2nd project year – 2005). GPF provides a framework to describe individual behaviours (procedural knowledge). It includes Resource Goal/Behaviour Description Framework (RGBDF), appropriate RGBDF-based domain ontology, appropriate RGBDF engine and a family of 'Semantic Adapters for Behaviour' to provide an opportunity to transform data from a variety of possible behaviour representation standards and formats to RGBDF and back.

- *General Networking Framework (GNF)*, for Integration (3rd project year – 2006).

RSCDF is an extension of RDF, which introduces upper-ontology for describing maintenance-oriented characteristics of resources: states and correspondent conditions, dynamics of state changes that happen, target condition of the resources and historical data about previous states. The 'lite-version' of the CDF Vocabulary Description Language (CDF Schema Lite) was extracted as a context description oriented part from the more complex Resource State/Condition Description Framework Vocabulary Description Language (RSCDF-Schema). More detailed information and implementation examples of CDF concepts as a part of RSCDF can be found from Kaykova et al. (2005a).

In continuation to the idea of CDF, such approach has been applied to context sensitive Resource Goal and Behaviour Description Framework. RGBDFS-Lite is an upper schema for description of resource goal and behaviour. It is based on the CDF schema and extends it as well as the Resource State and Condition Description Framework Schema (RSCDFS) does. One of the main features of the CDF is its ability to describe context-dependent facts (fact-statements) about resources. At the same time RGBDF brings a new (additional) vision to resource description. It is a description of a resource mental state. If we consider an agent (software agent) as a resource in the GUN, then we have to consider and model its believes, desires, intentions, etc. Now we can model not just the statements that describe the facts, but also goals-statements, that describe wishful for resource (agent) state of environment, other resources states and etc. Publication (Kaykova et al., 2005b) provides more information regarding RGBDF and gives the examples of CDF concept implementation.

4 Conclusions

Recent expectations regarding a new generation of the web strongly depend upon the success of Semantic Web technology. The RDF is a basis for an explicit, machine-readable representation of semantics of various web resources and enables a framework for interoperability of future Semantic Web-based applications. However previous research indicates that RDF is not suitable for describing highly dynamic and context-sensitive resources (e.g., industrial devices, processes, etc.). Therefore an appropriate extension of the existing RDF is necessary. We presented the CDF as a logical extension of the RDF as well as the CDF Vocabulary Description Language (CDF-Schema). We added a 'TrueInContext' component to the basic RDF triple ('subject-predicate-object') and considered the contextual value as a container of RDF statements. Some examples describing context-sensitive industrial resources with CDF were presented. We also added a probabilistic component to the model, which allows for not only describing the multilevel contextual dependence but also presumes the possibility for Bayesian reasoning within RDF model.

In this paper we have extended the existing RDF Property description with a cdfs:context property, which defines a context tolerance range for the subject property. Regarding the second significant part (the extension to a quadruple statement representation) which resulted to triple-based property description, we have defined a CDF quadruple with a container of the contextual statements as the fourth component.

Such a 'TrueInContext Statement' approach was elaborated during the first year of the 'Smart Resource Project' by Industrial Ontologies Group and successfully applied to the dynamic and context-sensitive industrial data description. The 'lite' version of the CDF Vocabulary Description Language (CDF Schema Lite) was extracted as a context description oriented part from the more complex Resource State/Condition Description Framework Vocabulary Description Language (RSCDF-Schema) designed by IOG.

As a logical continuation of this approach, we consider an elaboration of the context-dependent query language and quadruple storing. Another significant challenge will be utilising the nested and probabilistic context for advanced reasoning based on the CDF model.

Acknowledgements

This research has been performed as part of SmartResource ("Proactive Self-Maintained Resources in Semantic Web") project in Agora Center (University of Jyvaskyla, Finland) and funded by TEKES and industrial consortium of Metso Automation, TeliaSonera, TietoEnator and Science Park.

References

Hayes, P. and McBride, B. (2004) *RDF Semantics*, W3C Recommendation, February.

Jansen, B. (1993) 'Context: a real problem for large and shareable knowledge bases', *Building/Sharing Very Large Knowledge Bases (KBKS'93)*, December, Tokyo.

Joseph, P. *et al*. (2000) 'The American heritage dictionary of the English language', *Pickett*, 4th ed., Houghton Mifflin Company, Boston, http://www.houghtonmifflinbooks.com/epub/ahd4.shtml.

Kaikova, H., Khriyenko, O., Kononenko, O., Terziyan, V. and Zharko, A. (2004) 'Proactive self-maintained resources in semantic web', *Eastern-European Journal of Enterprise Technologies*, ISSN: 1729-3774, Kharkov, Ukraine, Vol. 2, No. 1, pp.4–16.

Kaykova, O., Khriyenko, O., Naumenko, A., Terziyan, V. and Zharko, A. (2005a) 'RSCDF: a dynamic and context sensitive metadata description framework for industrial resources', *Eastern-European Journal of Enterprise Technologies*, ISSN: 1729-3774, Vol. 3, No. 3, June, p.30.

Kaykova, O., Khriyenko, O., Terziyan, V. and Zharko, A. (2005b) 'RGBDF: Resource Goal and Behaviour Description Framework', in Bramer, M. and Terziyan, V. (Eds.): *Industrial Applications of Semantic Web, Proceedings of the 1st International IFIP/WG 12.5 Working Conference IASW-2005*, August 25–27, Springer, IFIP, Jyvaskyla, Finland, Vol. 188, pp.83–99.

Klyne, G. and Carroll, J.J. (2004) *Editors: Resource Description Framework (RDF): Concepts and Abstract Syntax*, W3C Recommendation, 10 February, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, latest version available at http://www.w3.org/TR/rdf-concepts/.

MacGregor, R. and Ko, I-Y. (2003) 'Representing contextualized data using semantic web tools', *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems*, ISWC 2003, October, Sanibal Island, Florida, USA.

Sicilia, M.A. and García-Barriocanal, E. (2004) 'Notes on extending terminological software frameworks with fuzziness through aspect oriented design', *WSEAS Transactions on Systems*, Vol. 3, No. 3, pp.1083–1089.

Straccia, U. (2001) 'Reasoning within fuzzy description logics', *J. Artificial Intelligence Research*, Vol. 14, pp.137–166.

Terziyan, V. (2003) 'Semantic web services for smart devices in a 'global understanding environment', in Meersman, R. and Tari, Z. (Eds.): *On the Move to Meaningful Internet Systems 2003*, OTM 2003 Workshops, *Lecture Notes in Computer Science*, Vol. 2889, Springer-Verlag, Catania, Sicilia, Italy, pp.279–291.

Terziyan, V. (2004) 'Semantic web services for smart devices based on mobile agents', in Marx Gomez, J. (Ed.): *Intelligent Mobile Agents in Peer-to-Peer Networks: Proceedings of the Forth International ICSC Symposium on Engineering Intelligent Systems (EIS-2004)*, Madeira, Portugal, February 29–March 2, ICSC Academic Press, Canada, ISBN: 3-906454-35-5, pp.10–16.

Tolle, K. and Wleklinski, F. (2004) *Trust and Context Using The RDF-Source Related Storage System (RDF-S3) and Easy RQL (eRQL)*, URL: http://www.eworks.de/research/2004/07/RDFS3AndErql/RDF-S3_and_eRQL.pdf.

Tresp, C.B. and Molitor, R. (1998) 'A description logic for vague knowledge', *Proceedings of the 13th Biennial European Conference on Artificial Intelligence (ECAI'98)*, J. Wiley & Sons, Brighton, UK, pp.361–365.

Notes

[1]The world's largest and most complete general knowledge base and commonsense reasoning engine (www.opencyc.org).

[2]Smart Resource Project (2004–2006) URL: http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm.

[3]Industrial Ontologies Group (IOG) URL: http://www.cs.jyu.fi/ai/OntoGroup/.

[4]RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, 10 February 2004 http://www.w3.org/TR/2004/REC-rdf-schema-20040210/. Latest version available at http://www.w3.org/TR/rdf-schema/.

Appendix: CDF vocabulary description language (CDF-schema): context extension of the RDFS[4]

1 CDF container

1.1 cdfs:Container

The cdfs:Container class is the class of CDF Statement containers (which contain just instances of cdfs:Statement class). It is an instance of rdfs:Class and a subclass of rdfs:Container.

1.2 cdfs:ContextContainer

The cdfs:ContextContainer class is the class of CDF Property containers (which contain just instances of cdfs:Property class). It is an instance of rdfs:Class and a subclass of rdfs:Container.

1.3 cdfs:PContainer

The cdfs:PContainer class is the class of CDF Statement probabilistic containers. Container contains the instances of contextual statements. It is an instance of rdfs:Class and a subclass of cdfs:Container.

1.4 cdfs:PropSignContainer

The cdfs:PropSignContainer class is the class of CDF Property Significance containers (which contain just instances of cdfs:PropertySignificance class). It is an instance of rdfs:Class and a subclass of rdfs:Container.

1.5 cdfs:member

cdfs:member is an instance of rdf:Property and a subproperty of rdfs:member property, it is used to state the member of a CDF Statement container.

A triple of the form:

C cdfs:member S

states that *C* is an instance of cdfs:Container and that the member of *C* is *S*.

The rdfs:domain of cdfs:member is cdfs:Container. The rdfs:range of cdfs:member is cdfs:Statement.

1.6 cdfs:cMember

cdfs:cMember is an instance of rdf:Property and a subproperty of rdfs:member property, it is used to state the member of a CDF Property container.

A triple of the form:

C cdfs:cMember P

states that *C* is an instance of cdfs:ContextContainer and that the member of *C* is *P*.

The rdfs:domain of cdfs:cMember is cdfs:Context Container. The rdfs:range of cdfs:cMember is cdfs:Property.

1.7 cdfs:pscMember

cdfs:pscMember is an instance of rdf:Property and a subproperty of rdfs:member property, it is used to state the member of a CDF Property Significance container.

A triple of the form:

C cdfs:pscMember Cl

states that *C* is an instance of cdfs:PropSignContainer and that the member of *C* is *Cl*.

The rdfs:domain of cdfs:pscMember is cdfs:PropSignContainer. The rdfs:range of cdfs:pscMember is cdfs:PropertySignificance.

1.8 cdfs:contextProbability

cdfs:contextProbability is an instance of rdf:Property, it is used to state the probability of the context (subject statement container).

A triple of the form:

C cdfs:contextProbability L

states that *C* is an instance of cdfs:PContainer and that the member of *C* is *L*.

The rdfs:domain of cdfs:contextProbability is cdfs:PContainer. The rdfs:range of cdfs:contextProbability is rdfs:Literal.

2 CDF statement

2.1 cdfs:Statement

cdfs:Statement is an instance of rdfs:Class and subclass of rdf:Statement. It is intended to represent the class of CDF statements. cdfs:Statement belongs to the domain of the properties cdfs:predicate, rdf:subject, rdf:object and cdfs:trueInContext. Different individual cdfs:Statement instances may have the same values for their cdfs:predicate, rdf:subject, rdf:object and cdfs:trueInContext properties.

2.2 rdf:subject

rdf:subject is an instance of rdf:Property that is used to state the subject of a statement.

A triple of the form:

S rdf:subject R

states that *S* is an instance of cdfs:Statement and that the subject of *S* is *R*.

The rdfs:domain of rdf:subject is rdf:Statement (and cdfs:Statement accordingly). The rdfs:range of rdf:subject is rdfs:Resource.

2.3 cdfs:predicate

cdfs:predicate is an instance of rdf:Property and subproperty of rdf:predicate that is used to state the predicate of a statement.

A triple of the form:

S cdfs:predicate P

states that *S* is an instance of cdfs:Statement, that *P* is an instance of cdfs:Property and that the predicate of *S* is *P*.

The rdfs:domain of cdfs:predicate is cdfs:Statement and the rdfs:range is cdfs:Property.

2.4 rdf:object

rdf:object is an instance of rdf:Property that is used to state the object of a statement.

A triple of the form:

S rdf:object O

states that *S* is an instance of cdfs:Statement and that the object of *S* is *O*.

The rdfs:domain of rdf:object is rdf:Statement (and cdfs:Statement accordingly). The rdfs:range of rdf:object is rdfs:Resource.

2.5 cdfs:trueInContext

cdfs:trueInContext is an instance of rdf:Property that is used to state the true context (contextual container) of a statement.

A triple of the form:

S cdfs:trueInContext C

states that *S* is an instance of cdfs:Statement, and that the context of *S* is *C*.

The rdfs:domain of cdfs:trueInContext is cdfs:Statement and the rdfs:range is cdfs:Container.

2.6 cdfs:falseInContext

cdfs:falseInContext is an instance of rdf:Property that is used to state the false context (contextual container) of a statement.

A triple of the form:

S cdfs:falseInContext C

states that *S* is an instance of cdfs:Statement, and that the context of *S* is *C*.

The rdfs:domain of cdfs:falseInContext is cdfs:Statement and the rdfs:range is cdfs:Container.

3 CDF property

3.1 cdfs:Property

cdfs:Property is the class of CDF properties. cdfs:Property an instance of rdfs:Class and subclass of rdf:Property.

3.2 cdfs:PropertySignificance

cdfs:PropertySignificance is the class of CDF properties significances. cdfs:PropertySignificance an instance of rdfs:Class.

3.3 rdfs:range

rdfs:range is an instance of rdf:Property that is used to state that the values of a property are instances of one or more classes.

The triple of the form:

P rdfs:range C

states that *P* is an instance of the class cdfs:Property, that *C* is an instance of the class rdfs:Class and that the resources denoted by the objects of quadruples whose predicate is *P* are instances of the class *C*.

Whenever *P* has more than one rdfs:range property, then the resources denoted by the objects of quadruples with predicate *P* are instances of all the classes stated by the rdfs:range properties.

The rdfs:range property can be applied to itself. The rdfs:range of rdfs:range is the class rdfs:Class. This states that any resource that is the value of an rdfs:range property is an instance of rdfs:Class.

The rdfs:range property is applied to properties. This can be represented in RDF using the rdfs:domain property. The rdfs:domain of rdfs:range is the class rdf:Property. This states that any resource with an rdfs:range property is an instance of rdf:Property or subproperty of it (cdfs:Property as an instance).

3.4 rdfs:domain

rdfs:domain is an instance of rdf:Property that is used to state that any resource that has a given property is an instance of one or more classes.

A triple of the form:

P rdfs:domain C

states that *P* is an instance of the class cdfs:Property, that *C* is an instance of the class rdfs:Class and that the resources denoted by the subjects of quadruples whose predicate is *P* are instances of the class *C*.

Where a property *P* has more than one rdfs:domain property, then the resources denoted by subjects of quadruples with predicate *P* are instances of all the classes stated by the rdfs:domain properties.

The rdfs:domain property may be applied to itself. The rdfs:domain of rdfs:domain is the class rdf:Property. This states that any resource with an rdfs:domain property is an instance of rdf:Property or subproperty of it (cdfs:Property as an instance).The rdfs:range of rdfs:domain is the class rdfs:Class. This states that any resource that is the value of an rdfs:domain property is an instance of rdfs:Class.

3.5 cdfs:context

cdfs:context is an instance of rdf:Property that is used to state that any property that has a given property has a restriction of a context tolerance range in the form of a contextual properties set.

A triple of the form:

P cdfs:context C

states that *P* is an instance of the class cdfs:Property, that *C* is an instance of the class cdfs:ContextContainer and that the resources denoted by the statement context of quadruples whose predicate is *P* are instances of the class *C*.

3.6 cdfs:subPropertyOf

The property cdfs:subPropertyOf is an instance of rdf:Property and subproperty of rdfs:subPropertyOf that is used to state that all resources related by one CDF property are also related by another one.

A triple of the form:

P1 cdfs:subPropertyOf P2

states that *P1* is an instance of cdfs:Property, *P2* is an instance of cdfs:Property and *P1* is a subproperty of *P2*. The cdfs:subPropertyOf property is transitive.

The rdfs:domain of cdfs:subPropertyOf is cdfs:Property. The rdfs:range of cdfs:subPropertyOf is cdfs:Property.

3.7 cdfs:subjectProperty

cdfs:subjectProperty is an instance of rdf:Property, it is used to state the subject property of the property significance object.

A triple of the form:

C cdfs:subjectProperty P

states that *C* is an instance of the class cdfs:PropertySignificance, that *P* is an instance of the class cdfs:Property.

3.8 cdfs:pSignificance

cdfs:pSignificance an instance of rdf:Property, it is used to state the significance value (between 0 and 1) of the subject property of the property significance object.

A triple of the form:

C cdfs:pSignificance L

states that *C* is an instance of the class cdfs:PropertySignificance, that *L* is an instance of the class rdfs:Literal.

4 CDF property instance

4.1 cdfs:significanceOfContext

cdfs:significanceOfContext an instance of cdfs:Property, it is used to state the significances of the contextual properties for subject property in certain context.

A quadruple of the form:

P cdfs:significanceOfContext C1 C2

states that *P* is an instance of the class cdfs:Property, that *C1* is an instance of the class cdfs:PropSignContainer. *C2* is an instance of cdfs:ContextContainer. *C2* is empty (it means that any property can be contextual for cdfs: significanceOfContext property).