Probabilistic Integration for the Semantic Web

Violeta Damjanovic
Salzburg Research, Austria
violeta.damjanovic@salzburgresearch.at

ABSTRACT

Along with the evolution from the Semantic Web to the Pervasive Semantic Web, the importance of taking into account ill-defined domains and imprecise information plays more important role. In this paper, we propose a solution to integrate ill-defined knowledge with classical Description Logic that will be extended to the management of uncertain information. The proposed solution is based on integration between the Probabilistic Asynchronous Process Algebra and OWL DL. It is ground on meta-metamodelling approach in representing both Probabilistic Asynchronous Process Algebra and OWL DL.

Key words: Meta-modelling, Pervasive computing, Probabilistic Asynchronous Pi-Calculus, Semantic Web, Web Ontology Language (OWL)

1. Introduction

The emergence of the Semantic Web technology enables both human and machine semantics to be used for formalizing and representing knowledge, as well as combining and inferring new one. Recently, a vision of the Pervasive Semantic Web has been appeared [1], in which *semantically connected information that represents the knowledge* (Semantic Web) meets *pervasively and unobtrusively connected computing devices, which are embedded in the environment (*pervasive computing).

Pervasive Semantic Web environment additionally connects semantics to *self-adaptive* and *self-organizing* services in order to semantically drive the interaction of local self-systems, theirs processes and system components to the level of the global distributive system behaviour.

Along with the evolution from the Semantic Web to the Pervasive Semantic Web, the importance of taking into account ill-defined domains and imprecise information (e.g. more facts are not true/false) plays more important role. Specifically, building Pervasive Semantic Web applications faces the problem of dealing with uncertain information that explains environment, users, process lifecycles, system' behaviour. For the sake of considering uncertain information, classical Description Logics (DL) that represents the logical foundation for ontologies become unsuited to a large range of the real world problems.

In this paper, we represent the Probabilistic Asynchronous Process Algebra (the π_{pa} for short) of O.M. Herescu [2] that we have found to be the most convenient formalism to express the real nature of processes taking place in a pervasive environment. The π_{pa} is fully based on Pi-Calculus, which is extended in the sense to enable the asynchronous nature of the processes (*events that are occurring independently of the program flow*), as well as dealing with uncertain knowledge collected using certain pragmatic mechanism, e.g. sensors networks, Semantic Web agents, Semantic Web services.

This paper is organized as follows: In Section II we briefly describe the problem that is addressed; In Section III we describe our research hypothesis; In Section 4 we further explain the proposed mechanism for integrating probabilistic processes and existing knowledge expressed in OWL (Web Ontology Language) DL. The proposed probabilistic integration mechanism, named *π_{pa}2OWL*, follows the Model-driven Approach (MDA). In Section 5 we review related work in the field of probabilistic extensions to OWL. The paper concluded in Section 6 with some conclusion remarks and directions for future research.

2. Problem Definition and Scope

The Semantic Web vision brings us a set of new technologies to capture the semantic relationships between information on the Web and to make them machine-consumable (readable, understandable, and (in)directly processable by machines). Semantic technologies include various languages, e.g. RDF (Resource Definition Language), OWL that is used for ontologizing of knowledge, querying ontologies and reasoning about knowledge. Moreover, the real world copes with changing information that express partial, inconsistent, and unreliable knowledge (uncertainty), which is often associated with:

- defective information and defective models of our knowledge,
- vague, fuzzy, incomplete, imprecise information,
- unpredictable user/agent behaviour,
- unpredictable environment events.

These limitations render DL unsuited to a large range of the real world problems (e.g. one of the serious limitations of DLs is that they can express little about the overlap between two concepts (classes of individuals) [3]).

Today, various approaches for dealing with uncertainty are defined. Their mainly differ in the underlying notion of uncertainty [17]. We briefly explain some of them below.

2.1 Bayesian Networks

Bayesian networks [3] allow a compact and natural representation of complex probability distributions by using independence assumptions, which is crucial to getting non trivial conclusions from a probabilistic knowledge base.

A Bayesian network is a Direct Acyclic Graph (DAG) in which the nodes are random variables. Each variable takes on a value in some predefined range. Each node in the network is associated with a Conditional Probability Table (CPT), which defines the probability of each possible value of the node, given each combination of values for the node's parents in the DAG.

From the perspective of using ontologies to express probability, a methodology has been proposed in [5] to translate the source and target ontologies into Bayesian networks and then map the concepts from the two ontologies based on evidential reasoning between the two translated Bayesian networks.

2.2 Fuzzy Theory

Fuzzy theory, introduced by Zadeh in [6], has been used in the context of searching and dealing with vague and imprecise knowledge, but not much work has been done in this field yet. Fuzzy theory allows to model vague memberships of individuals, while fuzzy IF-THEN rules allow evaluating good approximations of desired attribute values in a very efficient way [7].

The work explained in [8] has shown how fuzzy membership functions and fuzzy IF-THEN rules can be modelled with DL that support the concrete domain R and simple aggregate functions like min, max, sum, etc. A fuzzy logic extension of DL has been proposed in [9].

2.3 Paraconsistent Reasoning

Paraconsistent reasoning for the Semantic Web, as described in [10], involves several different approaches like:

(a) Relevant logics, which is based on "different worlds" developed by Routley and Meyer,

(b) Many-Valued systems, which represents the logic with more than two truth values,

(c) Non-Adjunctive systems,

(d) Non-Truth-Functional logics.

The implementation of algorithms for paraconsistent reasoning with OWL, named ParOWL, can be found in [11].

This paper represents an ongoing research on developing a solution that integrates probabilistic knowledge and OWL DL constructs. Here, we propose and investigate an integration mechanism based on metamodel transformation from the Probabilistic Asynchronous Algebra into OWL DL. We call this integration mechanism *p* $_{pa}$*2OWL*.

3. Research Hypothesis: Model-Driven Integration of the Probabilistic Asynchronous Processes and OWL DL

To support probabilistic ontology representation and reasoning in the Semantic Web environment, we use MDA approach in integrating:

- the π_{pa}, which is an extension of the Asynchronous Process Algebra with a notion of random choice, and

- OWL DL, which is a standard ontology language.

The proposed probabilistic integration mechanism is based on using MDA that enables defining models at various levels of abstraction and developing transformations between those models. More precisely, the proposed solution is ground on Meta-Object Facility (MOF) that is used for specifying metamodels.

First, we have defined π_{pa} metamodel as a source metamodel and reused OWL metamodel [12] as a target model.

Second, we have identified a collection of the transformation rules between the source model (based on π_{pa} metamodel) and the target model (based on OWL metamodel).

Figure 1 illustrates the proposed integration solution driven by MDA principles. In general, the MOF framework includes three layers shown on Figure 1:

- the model layer (M1) that contains the definition of the required structures;

- the metamodel layer (M2) that defines the terms in which the model is expressed,

- the meta-metamodel layer (M3) that defines the terms used to specify metamodels.

Each model from the M1 level conforms to an appropriate metamodel (M2 level). The M2 and M3 levels belong to the MOF technical space, whereas the M1 level involves the Semantic Web technical space, MOF technical space and XML technical space. A technical space is a working context with a set of additional concepts, body of knowledge, tools, required skills, and possibilities [18]. In order to exchange models between different technical spaces, it is necessary to provide transformations from one space to another. These transformations are also models.

For example, in the Pervasive Semantic Web technical space, the probabilistic asynchronous processes are collected in the form of π_{pa} records and described as an *.xml* file. In order to enable communication between the Pervasive Semantic Web technical space and MOF technical space, we transform *.xml* file into an equivalent *.ecore* (ECore XML XMI) format, which is a metamodel that follows the specification of Essential MOF (EMOF). Then, we use the Atlas Transformation Language (ATL) engine [13] to describe and implement the relevant transformation rules between π_{pa} and OWL constructs.

Finally, we transform the resulting OWL file, which has an *.ecore* extension, espressed by using OWL RDF/XMI exchange syntax (XML Metadata Interchange (XMI) represents an interchange format used for serialization of models of other languages (metamodels)) into an executable OWL file defined in OWL RDF/XML presentation syntax (*.owl* file). The transformation model, named π_{pa}*2OWL*, describes transformation rules that hold between appropriate the source and the target metamodels.

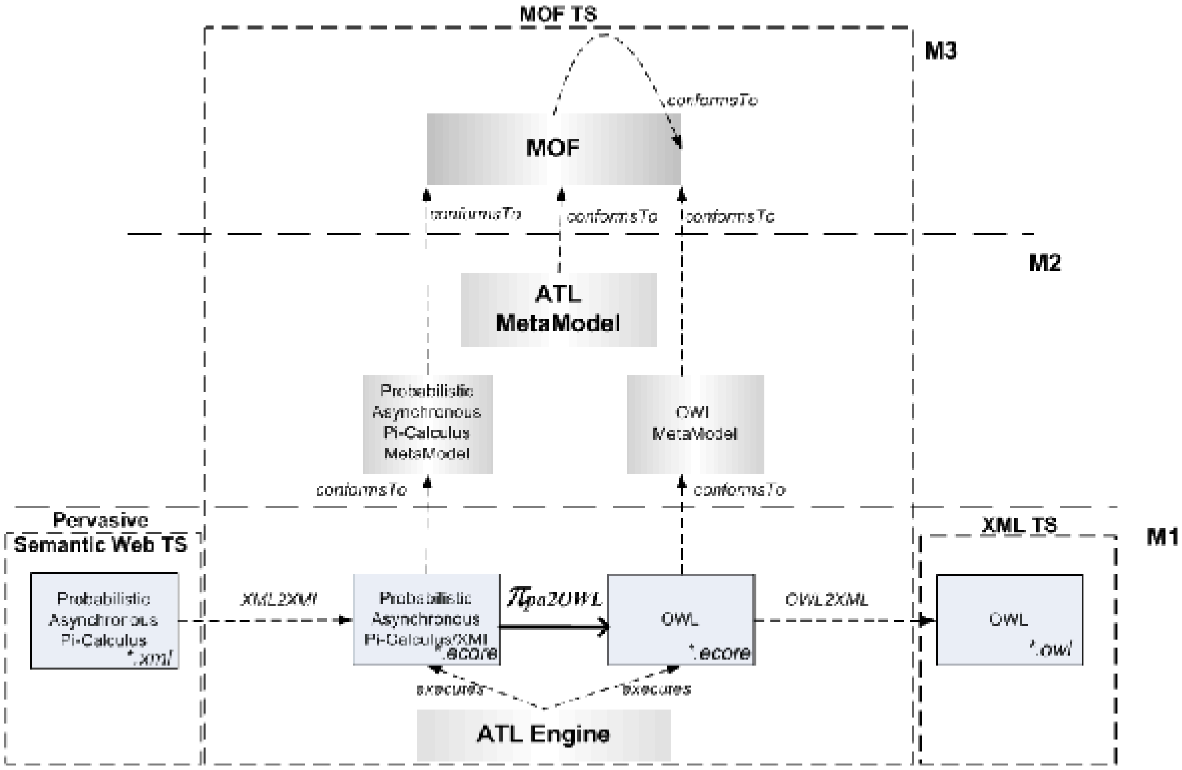

Figure 1. Transformation Scenario

4. π_{pa}2OWL: Probabilistic Integration for the Semantic Web

This Section gives a brief overview of the π_{pa} proposed in [2]. Then, based on the syntax and operational semantics of π_{pa}, we have defined π_{pa} metamodel and described π_{pa} metamodel in the form of Kernel Meta Metamodel (KM3) language. KM3 is a Domain Specific Language (DSL) for metamodel specification. Finally, we identify transformation rules between the source and the target metamodels.

4.1 Probabilistic Asynchronous Algebra

Probabilistic Asynchronous Pi-Calculus is based on both Robin Milner's Pi-Calculus of mobile processes and the probabilistic automata of Segala and Lynch [2]. It represents an improvement of the Pi-Calculus, considering *asynchronous algebra* on one hand and *probabilistic algebra* on the other hand. Asynchronous algebra is a subset of the Pi-Calculus in which communication is asynchronous and output processes are not allowed to go on continuously [2]. Additionally, the formalisms based on asynchronous communication are more suitable for a distributed implementation, compared with synchronous communication.

At the same time, the distributive problems require considering implementation of a certain probabilistic algorithms to enable a random choice, as well as asynchronous nature of processes that occur randomly in distributed architectures.

The operational semantics of the π_{pa} distinguishes between probabilistic and nondeterministic behaviour. *Probabilistic behaviour* is associated with a random choice of processes, whereas *nondeterministic behaviour* is related to the arbitrary decision of an external scheduler (agent) [2].

The π_{pa} is defined by the following grammar [2]:

$$a ::= x(y) \mid t \qquad (1)$$

$$P ::= \bar{x}y \,\Big|\, \sum_i p_i a_i P_i \,\Big|\, \nu x P \,\Big|\, P_i \,\Big|\, P_j \,\Big|\, X \,\Big|\, rec_X P \qquad (2)$$

As noted in (1), the π_{pa} includes input prefix, $x(y)$, and silent prefix, t , whereas output prefix is replaced by the output-action processes described as $\bar{x}y$ in (2). In addition, the π_{pa} processes are described with the *probabilistic choice operator*, $\sum_i p_i a_i P_i$, where p_i represents probabilities, and a_i is input or silent prefix.

We have defined the π_{pa} metamodel, based on the syntax and operational semantics of π_{pa} that is described in [2].

4.2 π_{pa} Metamodel

The π_{pa} metamodel is shown in Figure 2.

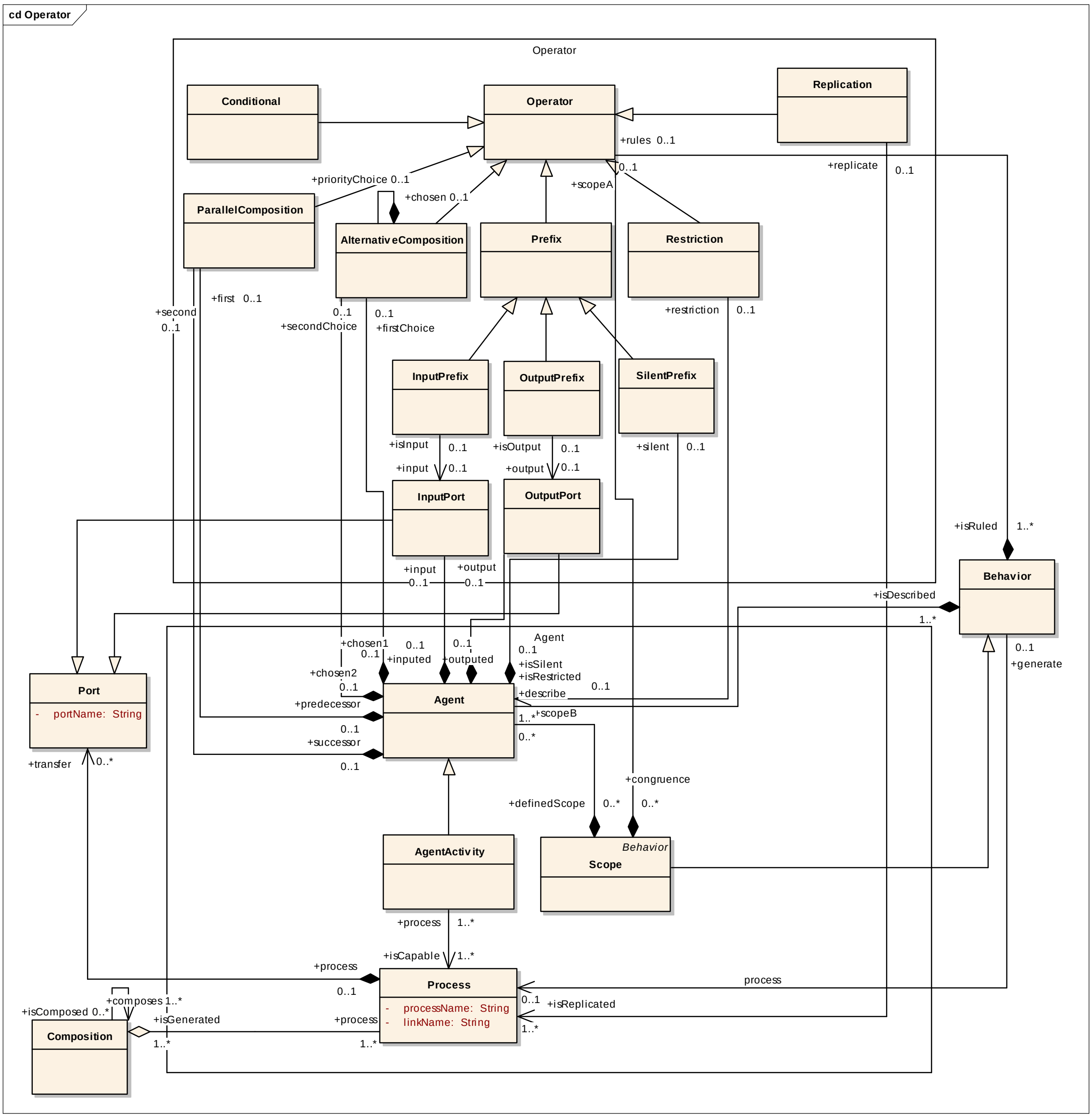

Figure 2. π_{pa} Metamodel

4.3 Probabilistic Integration between π_{pa} and OWL DL

The probabilistic integration of π_{pa} and OWL is based on building transformation models with the role to specify the way of producing the target models from the source models. At the same time, the transformation models have to confirm to a transformation metamodels that define the transformation semantics, as well as to confirm to the considered meta-metamodel [13]. We use the ATL that enables generating OWL model that conforms to the OWL metamodel (target), starting from the π_{pa} model that conforms to the π_{pa} metamodel (source) (shown in Figure 2).

A collection of transformation rules have been identified and applied to enable the probabilistic integration of the π_{pa}*2OWL*. Some of these transformation rules are represented in Table 1.

Table 1. Overview of the π_{pa}*2OWL* Transformation Rules

π_{pa} metamodel classes	OWL metamodel classes	Comments (based on an analogy between OMG' ODM specification described in [16] and the π_{pa} metamodel)
`Composition`	`OWLGraph`	
`Process`	`OWLOntology`	The name of the process is connected with the name of the ontology, which is further presented by URI.
`AgentActivity`	`OWLStatement`	If `OWLGraph` is linked with an `OWLOntology`, then they must exists an `OWLStatement`, which is a subset of `RDFStatement`
`Operator`	`OWLClass`	`OWLClass` differs between the following kinds of classes: `ComplementClass`, `EnumeratedClass`, `IntersectionClass`, `UnionClass`...
`ParallelComposition`	`IntersectionClass`	`OWLintersectionOf` links a parallel composition of processes with the rest of the processes that participate in process execution. It is analogous to logical conjunction.
`AlternativeComposition`	`UnionClass`	`owlUnionOf` statement is analogous to logical disjunction. It describes a class for which the class extension contains those individuals that occur in at least one of the class extensions of the class descriptions in the list.
`Restriction`	`ComplementClass`	`OWLcomplementOf` statement describes a class for which the class extension contains exactly those individuals that do not belong to the class extension of the class description that is the object of the statement. It is analogous to logical negation. The RES rule of the operational semantics of the π_{pa} models restriction on channel *y*: only the actions on channel different from *y* can be performed and possibly synchronized with an external process.
`Prefix`	`OWLRestriction`	`OWLRestriction` class have exactly one triple linking the restriction to a particular property, using the `owl:onProperty`. The restriction class should also have exactly one triple that represents the value or cardinality constraint on the property under consideration.
`InputPrefix`, `OutputPrefix`, `SilentPrefix`	`HasValueRestriction`	It describes a class of all individuals for which the property concerned has at least one value semantically equal to V. `OWLhasLiteralValue` links the restriction class to the literal that fills its value role.
`Scope`	`OWLDatatypeProperty`	Datatype properties are used to link individuals to data values. It is defined as an instance of the built-in OWL class `owl:DatatypeProperty`.
`InputPort`, `OutputPort`	`RDFSResource`	It is used to uniquely identify an RDF resource globally. It contains the `uriRef` property, which links the URI reference(s) with a resource.
`Behavior`	`RDFProperty`	It describes the relation between subject resource and object resource. Every property is associated with a set of instances, called the property extension.
`Agent`	`RDFStatement`	`RDFStatement` is a collection of subjects, predicates, and objects. It always points towards the object.
`Port`	`URIReference`	It identifies resources and properties.

5. Related Work

Recently, there have been some attempts to probabilistic extensions in DLs, such as the following examples:

a) P-Classic [3] is a probabilistic version of the DL CLASSIC that uses Bayesian networks to express uncertainty about the basic properties of an individual, the number of fillers for the different roles, and the properties of these fillers. Also, it allows the specification of a probability distribution over the properties of individuals.

The probabilistic component of a P-CLASSIC knowledge base includes:

- a number of different p-classes (probabilistic classes), each of which is a Bayesian network over basic properties,
- the number of fillers (for the different roles), and
- the p-classes from which the role fillers are chosen.

b) P-SHOQ(D) [14] is the probabilistic extension of DL SHOQ(D), which is the semantics behind DAML+OIL (without inverse roles), based on the notion of probabilistic lexicographic entailment from probabilistic default reasoning. It is able to represent assertional probabilistic knowledge about concepts and role instances.

c) PTDL [15] extends Tiny Description Logic (TDL) with "Conjunction" and "Role Quantification" operators.

d) BayesOWL [16] is to translate a given ontology to a Bayesian network in a systematic and practical way, and then treats ontological reasoning as probabilistic inferences in the translated Bayesian networks (it's not to extend OWL with probability theory). It is non-intrusive approach in the sense that neither OWL nor ontologies defined in OWL need to be modified.

6. Conclusion and Future Work

Nowadays, a novel search engine technology becomes designed to support ontology-based search refinements in a way that ontology formalisms can capture uncertainty and express relevant uncertainties about the entities (classes of individuals) and relationships between classes.

To date there exist two ways of expressing probabilistic knowledge in the Semantic Web. The first approach is focused on *achieving a probabilistic extension* of a DL-based language, whereas the second approach deals with *integrating probabilistic and deterministic knowledge* taking place in the Semantic Web environment. However, none of these existing attempts has considered the possibility to treat the probabilistic processes in a way that enables model-driven transformation mechanism to integrate the probabilistic knowledge into OWL DL.

In this paper, as direction to connect research from the domain of asynchronous communication and probabilistic behaviour, as well as pervasive and unpredictable behaviour of processes on one hand with the Semantic Web technologies on the other hand, we propose the $\pi_{pa}2OWL$ integration mechanism between π_{pa} and OWL DL.

To our knowledge, this work represents the first attempt to study how the MDA and MOF can be handled to provide a mechanism for integrating probabilistic knowledge into OWL DL.

References:

[1] J. I. Vazquez, D. L. de Ipiña and I. Sedano, "SoaM: A Web-powered Architecture for Designing and Deploying Pervasive Semantic Devices," International Journal of Web Information Systems (IJWIS), Vol. 2, No. ¾, 2006, pp. 212-224.

[2] O. M. Herescu, *Probabilistic Asynchronous Pi-Calculus.* A PhD Thesis, Pennsylvania State University, December 2002.

[3] D. Koller, A. Levy and A. Pfeffer, "P-Classic: A Tractable Probabilistic Description Logic," Proceedings of the 14th National Conference on Artificial Intelligence, 1997, pp. 390-397.

[4] Z. Ding, Y. Peng, R. Pan and Y. Yu, "A Bayesian Methodology Towards Automatic Ontology Mapping," in First international workshop on Contexts and Ontologies: Theory, Practice and Applications, *AAAI-05*, 2005.

[5] R. Pan, Z. Ding, Y. Yu and Y. Peng, "A Bayesian network approach to ontology mapping," in Proceedings of the 4th International Semantic Web Conference (ISWC-05), 2005.

[6] L. A. Zadeh, "Fuzzy sets," in Information and Control, Vol. 8, 1965, pp. 338–353.

[7] R. R. Yager, S. Ovchinnikov, R. M. Tong and H. T. Nguyen, (Eds.) *Fuzzy Sets and Applications - Selected Papers by L. A. Zadeh*, John Wiley & Sons, New York, USA, 1987.

[8] S. Agarwal, P. Hitzler, "Modeling Fuzzy Rules with Description Logics," in Proceedings of Workshop on OWL Experiences and Directions, Ireland, 2005.

[9] U. Straccia, "Reasoning Within Fuzzy Description Logics," Journal of Artificial Intelligence, 2001, pp. 137-166.

[10] S. Schaffert, F. Bry, P. Besnard, H. Decker, S. Decker, C. Enguix and A. Herzig, "Paraconsistent Reasoning for the Semantic Web," Uncertainty Reasoning for the Semantic Web (URSW05) at ISWC05, Ireland, 2005.

[11] Y. Ma, P. Hitzler, Z. Lin, "Algorithms for Paraconsistent Reasoning with OWL," in E. Franconi, M. Kifer, W. May, (Eds.), The Semantic Web: Research and Applications. Proceedings of the 4th European Semantic Web Conference, ESWC2007, Innsbruck, Austria, Springer, 2007, pp. 399-413.

[12] IBM & Sandpiper Software, *Ontology Definition Metamodel*, Sixth Revised Submission, OMG, 2006. Online available: http://www.omg.org/cgi-bin/doc?ad/2006-05-01

[13] ATLAS Group, *ATL User Manual*, February 2006.

[14] R. Giugno and T. Lukasiewicz, "P-SHOQ(D): A Probabilistic Extension of SHOQ(D) for Probabilistic Ontologies in the Semantic Web," in INFSYS Research Report 1843-02-06, Austria, 2002.

[15] P. M. Yelland, "Market Analysis Using Combination of Bayesian Networks and Description Logics," in Sun Microsystems Technical Report TR-99-78, 1999.

[16] Z. Ding, Y. Peng and R. Pan, "BayesOWL: Uncertainty Modelling in Semantic Web Ontologies," in Soft Computing in Ontologies and Semantic Web, October 2005.

[17]Y. Loyer, U. Straccia, "Default Knowledge in Logic Programs with Uncertainty," in Proc. of the 19th International Conference on Logic Programming, 2003.

[18] I. Kurtev, J. Bezivin, M. Aksit, "Technological Spaces: An Initial Appraisal," CoopIS, DOA'2002 Federated Conferences, Industrial Track, Invine, 2002.