PR-OWL 2.0 – Bridging the Gap to OWL Semantics

Rommel N. Carvalho, Kathryn B. Laskey, and Paulo C.G. Costa

Department of SEOR / Center of Excellence in C4I,
George Mason University, USA
rommel.carvalho@gmail.com, {klaskey,pcosta}@gmu.edu
http://www.gmu.edu

Abstract. The past few years have witnessed an increasingly mature body of research on the Semantic Web (SW), with new standards being developed and more complex use cases being proposed and explored. As complexity increases in SW applications, so does the need for principled means to represent and reason with uncertainty in SW applications. One candidate representation for uncertainty representation is PR-OWL, which provides OWL constructs for representing Multi-Entity Bayesian Network (MEBN) theories. This paper reviews some shortcomings of PR-OWL 1.0 and describes how they are addressed in PR-OWL 2. A method is presented for mapping back and forth between OWL properties and MEBN random variables (RV). The method applies to properties representing both predicates and functions.

Keywords: uncertainty reasoning, OWL, PR-OWL, MEBN, probabilistic ontology, Semantic Web, compatibility.

1 Introduction

Appreciation is growing within the Semantic Web community of the need to represent and reason with uncertainty. Several approaches have emerged to treating uncertainty in Semantic Web applications (e.g., [5, 6, 8, 9, 12, 14, 17, 18]). In 2007, the World Wide Web Consortium (W3C) created the Uncertainty Reasoning for the World Wide Web Incubator Group (URW3-XG) to identify requirements for reasoning with and representing uncertain information in the World Wide Web. The URW3-XG concluded that standardized representations are needed to express uncertainty in Web-based information [11]. A candidate representation is Probabilistic OWL (PR-OWL) [5], an OWL upper ontology for representing probabilistic ontologies based on Multi-Entity Bayesian Networks (MEBN) [10].

Compatibility with OWL was a major design goal for PR-OWL [5]. However, there are several ways in which the initial release of PR-OWL falls short of complete compatibility. First, there is no mapping in PR-OWL to properties of OWL. Second, although PR-OWL has the concept of meta-entities, which allows the definition of complex types, it lacks compatibility with existing types already present in OWL.

These problems have been noted in the literature [16]:

F. Bobillo et al. (Eds.): URSW 2008-2010/UniDL 2010, LNAI 7123, pp. 1–18, 2013.

> PR-OWL does not provide a proper integration of the formalism of MEBN and the logical basis of OWL on the meta level. More specifically, as the connection between a statement in PR-OWL and a statement in OWL is not formalized, it is unclear how to perform the integration of ontologies that contain statements of both formalisms.

This Chapter describes the need for a formal mapping between random variables defined in PR-OWL and properties defined in OWL, and proposes an approach to such a mapping. We then explain why PR-OWL 1.0 does not support such a mapping. Next, we present an approach to overcome the limitations in PR-OWL 1.0 by introducing new relationships created in PR-OWL 2.0. Finally, we present a scheme for the mapping back and forth from triples into random variables.

2 PR-OWL - An OWL Upper Ontology for Defining MEBN Models

PR-OWL was proposed as an extension to the OWL language to define probabilistic ontologies expressed in MEBN [10], a first-order probabilistic language (FOPL) [13]. Before delving into the details of PR-OWL, we provide a brief overview of MEBN.

As a running example, we consider an OWL ontology for the public procurement domain. A fuller treatment of the procurement ontology can be found in [4]. The ontology defines concepts such as procurement, winner of a procurement, members of a committee responsible for a procurement, etc. Figure 1 presents an OWL ontology with a few of the concepts that would be present in this domain. In the figure we can see that a front man is defined as a person who is a front for some organization (as shown in the equivalent class expression `Person and isFrontFor some Organization` for the `FrontMan` class in Figure 1).

Although there is great interest in finding people acting as fronts, it is in general unknown whether a given person meets this definition. This is a typical case where we would benefit from reasoning with uncertainty. For example, if an enterprise wins a procurement for millions of dollars, but the responsible person for this enterprise makes less than 5 thousand dollars a year or if that person has only a middle school education, then it is likely that this responsible person is a front for that enterprise. That is, we can identify potential fronts by examining the value of the procurement, the income of the responsible person, and his/her education level. Although we are not certain that this person is in fact a `FrontMan`, we would like to at least use the available information to draw an inference that the person is likely to be a front. This strategy is preferable to ignoring the evidence supporting this hypothesis and saying that we simply do not know whether or not this person is a front. It is also preferable to creating an arbitrary rule declaring that certain combinations of education level and income imply with certainty that a person is a `FrontMan`.

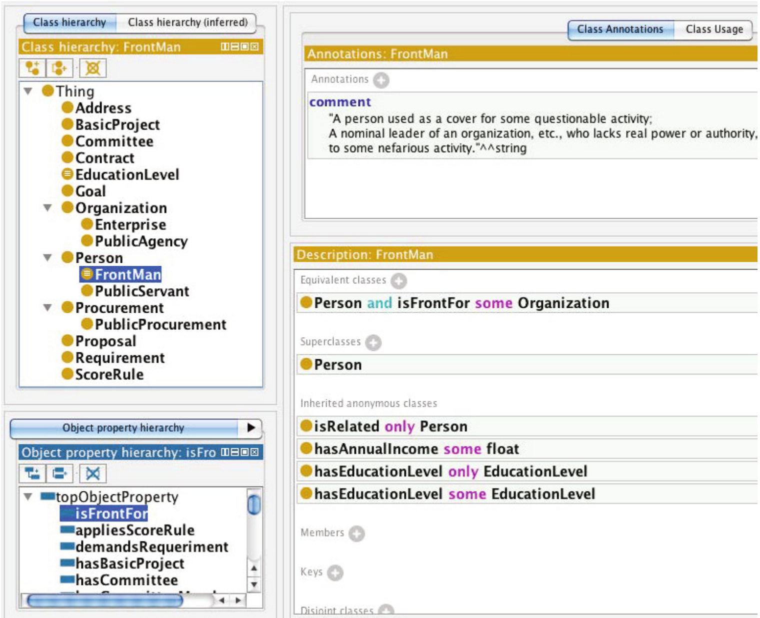

Fig. 1. OWL ontology for the public procurement domain

Figure 2 shows a formalization of this uncertain relationship in MEBN logic. MEBN represents knowledge as a collection of MEBN Fragments (MFrags), which are organized into MEBN Theories (MTheories).

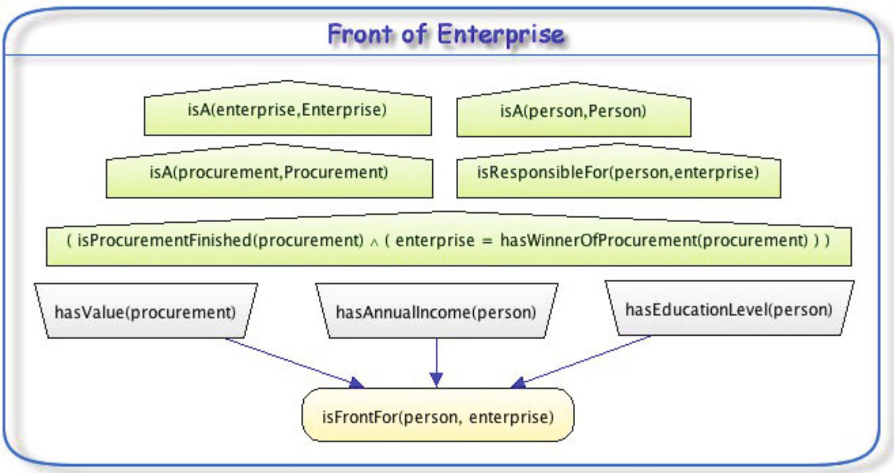

Fig. 2. Front of an Enterprise MFrag

An MFrag contains random variables (RVs) and a fragment graph representing dependencies among these RVs. It represents a repeatable pattern of knowledge that can be instantiated as many times as needed to form a BN addressing a specific situation, and thus can be seen as a template for building and combining fragments of a Bayesian network. It is instantiated by binding its arguments to domain entity identifiers to create instances of its RVs. There are three kinds of nodes: context, resident and input. Context nodes represent conditions that must be satisfied for the distributions represented in the MFrag to apply. Input nodes may influence the distributions of other nodes in an MFrag, but their distributions are defined in their home MFrags. Distributions for resident nodes are defined within the MFrag by specifying local distributions conditioned on the values of the instances of their parents in the fragment graph.

Figure 2 presents a MEBN Fragment, where we see that the education level and annual income of a responsible person and the value of a procurement influence whether the person is front for the procurement. However, in order for the probabilistic relations described to hold, some conditions have to be satisfied, namely that the person we are considering as a possible front must be responsible for the enterprise we are examining, which is the winner of the procurement that is already finished. In other words, if the person is not responsible for the enterprise, there is no reason for this person to be considered a front for this enterprise. The same principle holds if the enterprise did not win that procurement, *i.e.*, the value of a procurement that was not won by that enterprise will not affect the likelihood of having a front for that enterprise. These conditions that must be satisfied for the probabilistic relationship to hold are depicted inside the green pentagonal shapes in the figure.

Figure 2 shows only the structure of our reasoning just described. In order to be complete, we also need to define the conditional probability distribution, also called local probability distribution (LPD), for the random variable being defined. Listing 1.1[1] presents the LPD for the random variable `isFrontFor(person, enterprise)`. When a random variable has its LPD defined within the MFrag where it appears, we call it a resident node. Nodes that are not resident nodes, but influence the distribution of resident nodes, are called input nodes and they have their LPDs defined in another MFrag. A collection of MFrags that guarantees a joint probability distribution over instances of random variables form a MEBN Theory (MTheory).

Listing 1.1. LPD for `isFrontFor(person, enterprise)`

```
if any procurement have ( hasValue = From100kTo500k ) [
    if any person have ( hasAnnualIncome = Lower10k |
         hasEducationLevel = NoEducation ) [
         true = .9 ,
         false = .1
    ] else if any person have ( hasAnnualIncome = From10kTo30k |
         hasEducationLevel = MiddleSchool ) [
```

[1] This LPD is notional only. No real data or statistics was used.

```
            true = .6,
            false = .4
        ] else [
            true = .00001,
            false = .99999
        ]
] else if any procurement have ( hasValue = From500kTo1000k ) [
    if any person have ( hasAnnualIncome = Lower10k |
            hasEducationLevel = NoEducation ) [
            true = .95,
            false = .05
        ] else if any person have ( hasAnnualIncome = From10kTo30k |
            hasEducationLevel = MiddleSchool ) [
            true = .8,
            false = .2
        ] else if any person have ( hasAnnualIncome = From30kTo60k |
            hasEducationLevel = HighSchool ) [
            true = .6,
            false = .4
        ] else [
            true = .00001,
            false = .99999
        ]
] else if any procurement have ( hasValue = Greater1000k ) [
    if any person have ( hasAnnualIncome = Lower10k |
            hasEducationLevel = NoEducation ) [
            true = .99,
            false = .01
        ] else if any person have ( hasAnnualIncome = From10kTo30k |
            hasEducationLevel = MiddleSchool ) [
            true = .9,
            false = .1
        ] else if any person have ( hasAnnualIncome = From30kTo60k |
            hasEducationLevel = HighSchool ) [
            true = .8,
            false = .2
        ] else if any person have ( hasAnnualIncome = From60kTo100k |
            hasEducationLevel = Undergraduate ) [
            true = .6,
            false = .4
        ] else [
            true = .00001,
            false = .99999
        ]
] else [
    true = .00001,
    false = .99999
]
```

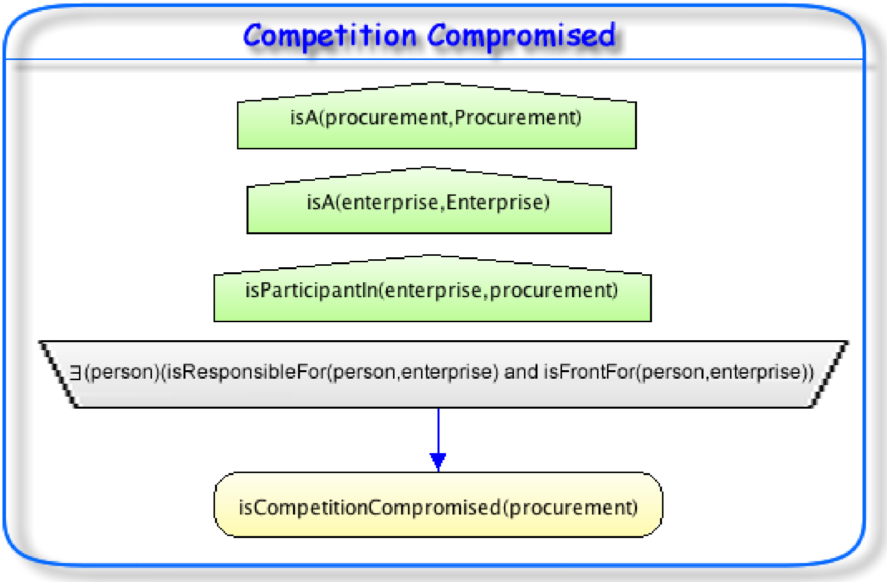

Fig. 3. Competition Compromised MFrag

The major advantage of MEBN when compared to Bayesian networks (BN), is the ability to represent repeated structure. In order to understand what kind of repetition can be represented and why it is important, let's introduce a new MFrag called Competition Compromised. In this MFrag (see Figure 3) we have the probabilistic rule that says that if any participant enterprise has at least one responsible person as a front, then it is more likely that the competition is going to be compromised in this procurement.

Now we can see how we benefit from the MEBN model. Figures 4 and 5 present two different BNs generated from our MEBN model given the information available in two different scenarios. In one we only have two participant enterprises (`enterprise2` and `enterprise3`), while in the other we have four participant enterprises (`enterprise1`, `enterprise2`, `enterprise3`, and `enterprise4`). Depending on the number of enterprises participating in a given procurement and the information available about the person responsible for those enterprises, we would have a different BN being used. In other words, with just one MEBN representation we can instantiate many different BN models representing problems with different numbers of individuals, as appropriate for each query of interest.

Probabilistic OWL (PR-OWL) is an upper ontology defined in OWL for representing MEBN theories. In other words, PR-OWL defines classes and properties for MEBN terms, like `MTheory`, `MFrag`, `hasMFrag`, and `ResidentNode` and restrictions on those terms (*e.g.*, MTheory is a collection of MFrags - *i.e.*, `hasMFrag some MFrag`) in order to allow the definition of MEBN models. In PR-OWL, a probabilistic ontology (PO) has to have at least one individual of class `MTheory`, which is basically a label linking a group of MFrags that collectively form a valid

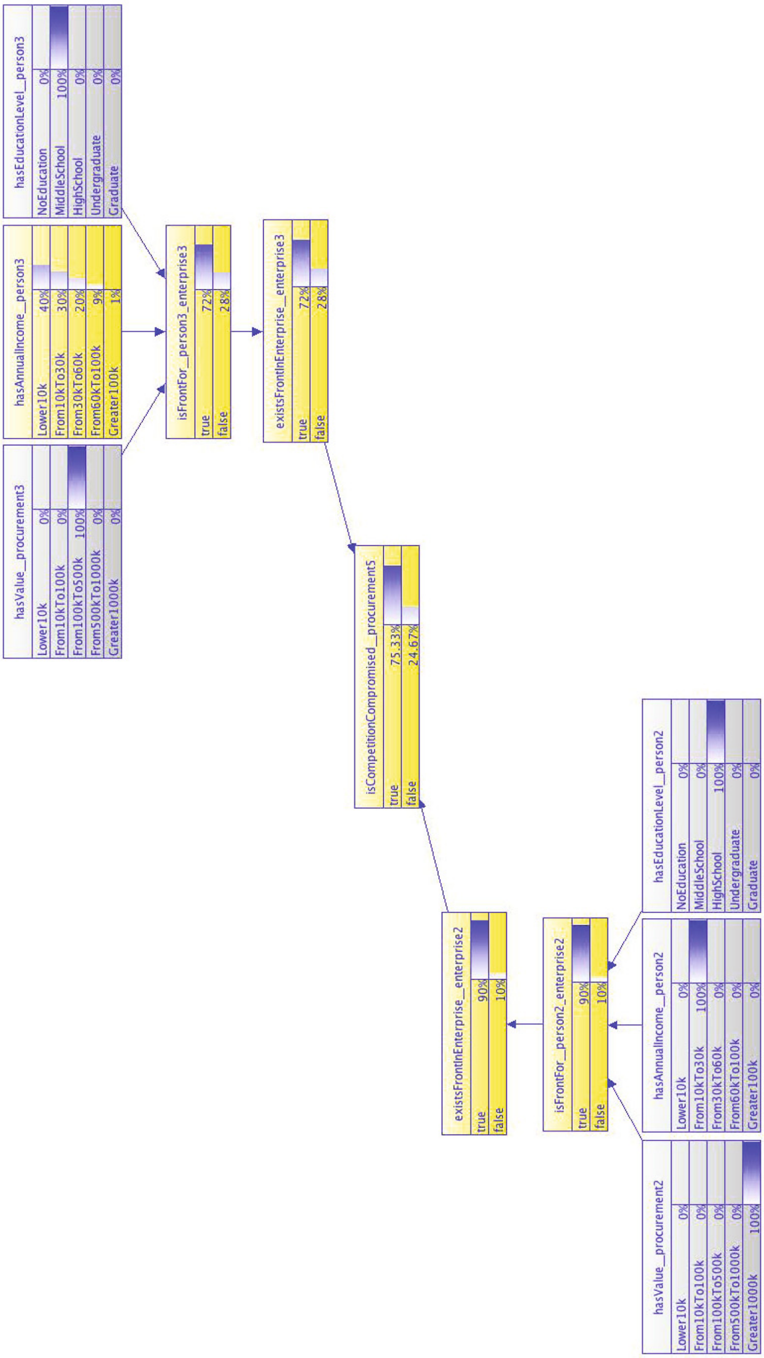

Fig. 4. Situation specific BN (SSBN) generated for the query `isCompetitionCompromised(procurement5)` given that there are two enterprises participating in this procurement

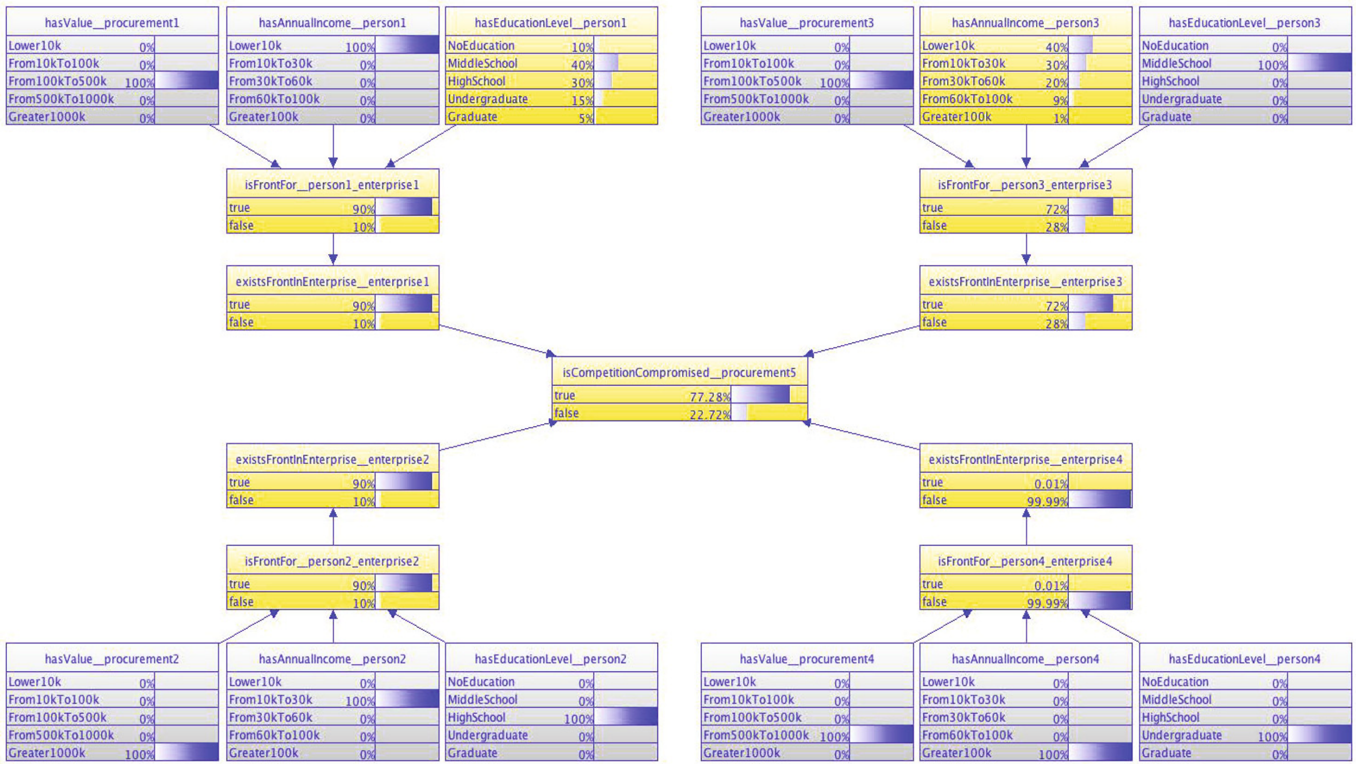

Fig. 5. SSBN generated for the query `isCompetitionCompromised(procurement5)` given that there are four enterprises participating in this procurement

MTheory. In actual PR-OWL syntax, that link is expressed via the object property `hasMFrag` (which is the inverse of object property `isMFragIn`). Individuals of class `MFrag` are comprised of nodes. Each individual of class `Node` is a random variable (RV) and thus has a mutually exclusive, collectively exhaustive set of possible states. In PR-OWL, the object property `hasPossibleValues` links each node with its possible states, which are individuals of class `Entity`. Finally, random variables (represented by the class `Node` in PR-OWL) have unconditional or conditional probability distributions, which are represented by class `ProbabilityDistribution` and linked to their respective nodes via the object property `hasProbDist`. This property would define a LPD like the one presented in Listing 1.1. Figure 6 presents an example of a PO for the procurement domain using the PR-OWL language.

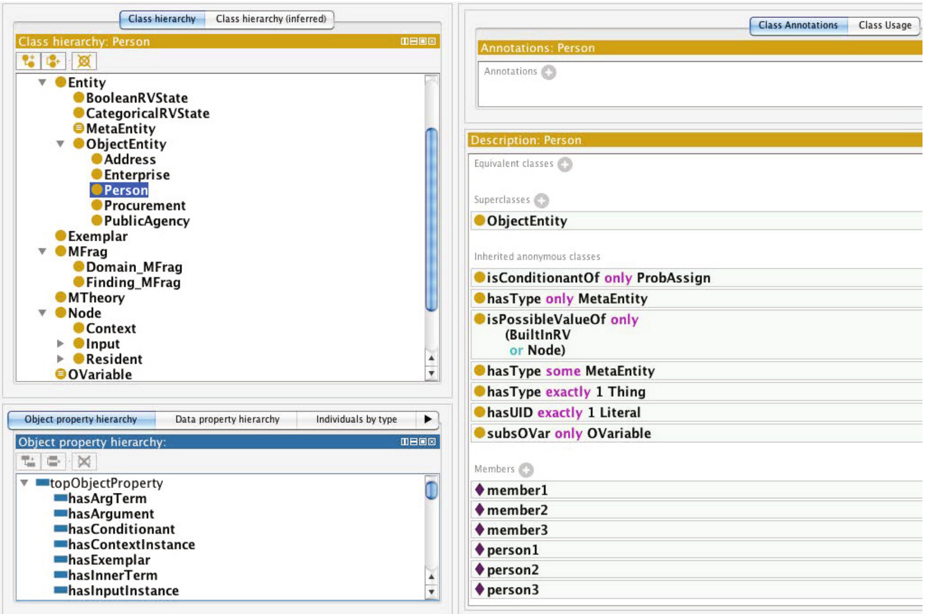

Fig. 6. Example of a PO for the procurement domain using PR-OWL language

3 Why Map PR-OWL Random Variables to OWL Properties?

Ideally, it should be possible to use PR-OWL to reason probabilistically about uncertain aspects of an existing ontology represented in OWL. For instance, Figure 7 presents some information we might have available in an OWL ontology for the procurement domain. Specifically, the ontology represents a person called Joe who has a middle school education level and an income of $5,000. As shown in the figure, we might want to generate a BN in order to draw inferences about whether Joe is a front for a procurement for which he is responsible. Although we cannot say for certain that `John_Doe` is a `FrontMan`, the likelihood is high given his low annual income, his low education level, and the high value of the procurement won by his enterprise. In order to construct the BN allowing us to draw this inference, we need to relate the knowledge expressed in the OWL ontology to PR-OWL random variables.

The problem with PR-OWL 1.0 is that it has no mapping between the random variables used in PR-OWL and the properties used in OWL. In other words, there is nothing in the language that tells us that the RV `hasEducationLevel(person)` defines the uncertainty of the OWL property `hasEducationLevel`. So, even if we have information about the education level of a specific person, we cannot connect that information In other words, even if we we have the triple `John_Doe hasEducationLevel middleSchool`, we would not be able to instantiate the random variable `hasEducationLevel(person)` for `John_Doe`. Although the OWL property `hasEducation` and the RV `hasEducationLevel(person)` have similar syntax, there is no formal representation of this link (as depicted in Figure 8). In other words, we cannot use the information available in an OWL ontology (the triples with information about individuals) to perform probabilistic reasoning. Full compatibility between PR-OWL and OWL requires this ability.

In fact, Poole *et al.* [15] states that it is not clear how to match the formalization of random variables from probabilistic theories with the concepts of individuals, classes and properties from current ontological languages like OWL. However, Poole *et al.* [15] suggests, "We can reconcile these views by having properties of individuals correspond to random variables." This is exactly the approach used in this work to integrate MEBN logic and the OWL language. This integration is a major feature of PR-OWL 2.0 [1].

4 The Bridge Joining OWL and PR-OWL

The key to building the bridge that connects the deterministic ontology defined in OWL and its probabilistic extension defined in PR-OWL is to understand how to translate one to the other. On the one hand, given a concept defined in OWL, how should its uncertainty be defined in PR-OWL in a way that maintains its semantics defined in OWL? On the other hand, given a random variable defined in PR-OWL, how should it be represented in OWL in a way that respects its uncertainty already defined in PR-OWL?

Imagine we are trying to define the RV `hasEducationLevel_RV`[2], which represents the MEBN RV `hasEducationLevel(person)` used in Figure 2. Let's also assume that we have an OWL property called `hasEducationLevel`, which is a functional property with domain `Person` and range `EducationLevel`,

[2] This is the OWL syntax for this RV. In MEBN we represent a RV by its name followed by the arguments in parentheses. In OWL the arguments are defined by the property `hasArgument`.

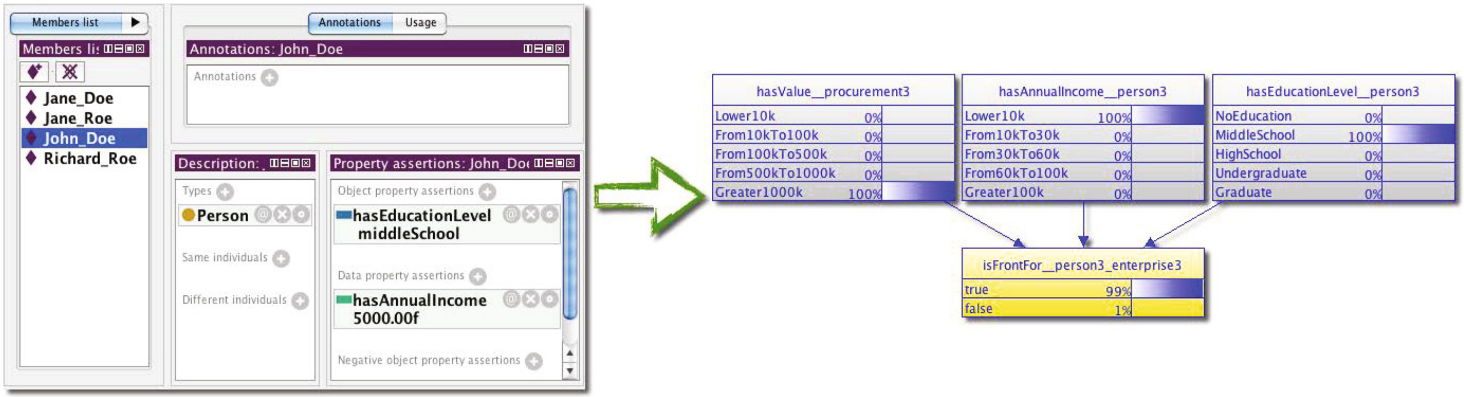

Fig. 7. Using triples for probabilistic reasoning

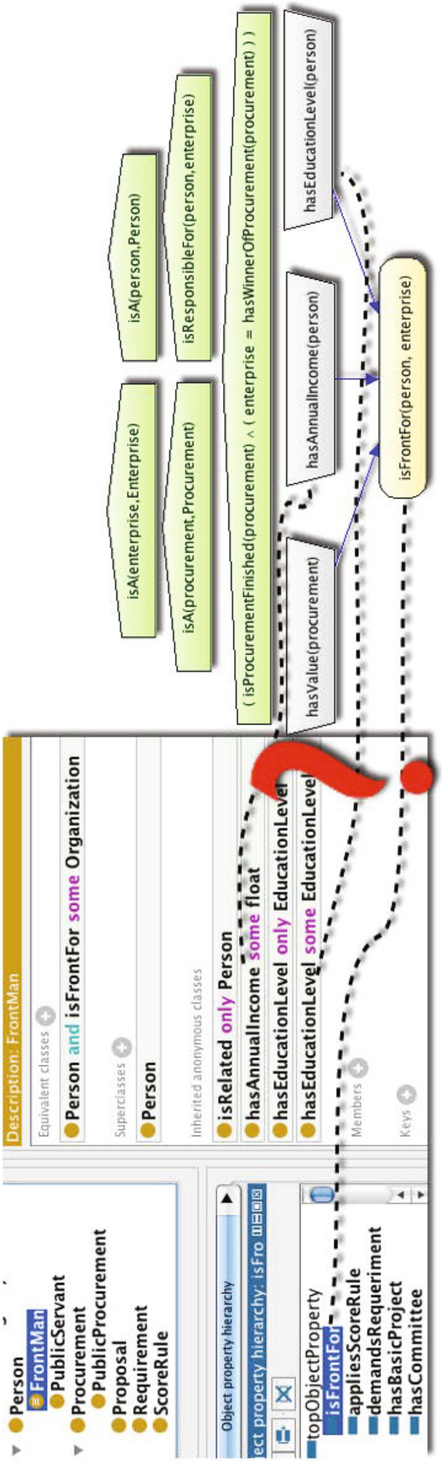

Fig. 8. Unknown mapping between PR-OWL 1.0 RVs and OWL properties

and an OWL property called `aspiresEducationLevel`, which is also a functional property with domain `Person` and range `EducationLevel`. As shown in Figure 9, in PR-OWL 1.0 it is not possible to distinguish whether the `hasEducationLevel_RV` is defining the uncertainty of the OWL property `hasEducationLevel` or `aspiresEducationLevel`. To clarify this problem, imagine that John Doe has only middle school (`John_Doe hasEducationLevel middleSchool`), but he aspires to have a graduate degree (`John_Doe aspiresEducationLevel graduate`). If we do not explicitly say which OWL property should be used to instantiate the `hasEducationLevel_RV`, we might end up saying that `hasEducationLevel(John_Doe) = graduate`, instead of saying that `hasEducationLevel(John_Doe) = middleSchool`, which is the intended semantics.

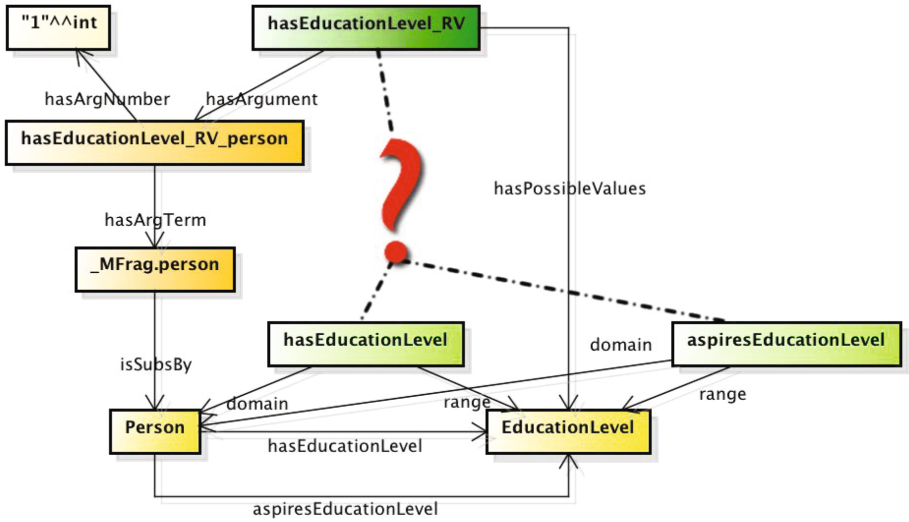

Fig. 9. PR-OWL 1.0 lack of mapping from RVs to OWL properties

A simple solution is to add a relation between a PR-OWL RV and the OWL property that this RV defines the uncertainty of, as suggested by Poole *et al.* [15]. In PR-OWL 2.0 this relation is called `definesUncertaintyOf` [1][3]. However, this is not enough to have a complete mapping between RVs and OWL properties. Another problem appears when we try to define n-ary RVs. This mapping is not as straight forward as the previous one because OWL only supports binary properties (for details on suggested work arounds to define n-ary relations in OWL see [7]).

[3] We make the distinction between property and property_RV with the `definesUncertaintyOf` relation, instead of just using the property as the RV, in order to stay within OWL DL. For more information see Section 4.4.2 in [1].

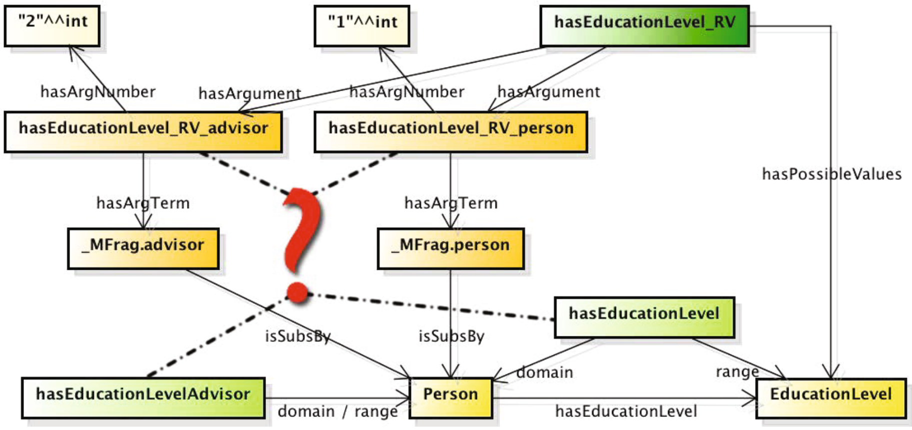

Fig. 10. PR-OWL 1.0 lack of mapping from arguments to OWL properties

Imagine we now want to represent not only the education level a person has, but also who was the advisor when this person attained that education level. So now, besides having the property `hasEducationLevel`, we also have the property `hasEducationLevelAdvisor`, which has `Person` as both domain and range. Thus, our RV now is `hasEducationLevel(person,advisor)`. With this new scenario, we can see that a similar problem occurs with the mapping of arguments. As it can be seen in Figure 10, there is nothing in PR-OWL 1.0 that tells which argument is associated with which property. To clarify the problem, imagine that Richard Roe has graduate education level (`Richard_Roe hasEducationLevel graduate`) and that his advisor was J. Pearl (`Richard_Roe hasEducationLevelAdvisor J_Pearl`). When instatiating the `hasEducationLevel(person,advisor)` RV, machines would not know who is the student and who is the advisor. Although this mapping is obvious for a human being, without an explicit mapping of the arguments, machines could end up using Richard Doe as the advisor and J. Pearl as the student (`hasEducationLevel(J_Pearl,Richard_Roe)`), instead of using J. Pearl as the advisor and Richard Doe as the student (`hasEducationLevel(Richard_Roe,J_Pearl)`).

As expected, to a similar problem we apply a similar solution. In PR-OWL 2 we have a relation between an argument to a RV and the OWL property it refers to. However, unlike the RV mapping, the argument mapping refers to either the domain or the range of a property, not to the property itself. For instance, in the `hasEducationLevel(person,advisor)` RV, the `person` argument refers to the domain of the OWL property `hasEducationLevel`, which is a `Person`. The `advisor` argument, on the other hand, refers to the range of the OWL property `hasEducationLevelAdvisor`, which is also a `Person`, but a different one (a person cannot be his/her own advisor). Therefore, in order to differentiate when the argument refers to the domain or to the range of a property, we add to PR-OWL 2.0 the relations `isSubjectIn` and `isObjectIn`. More examples of

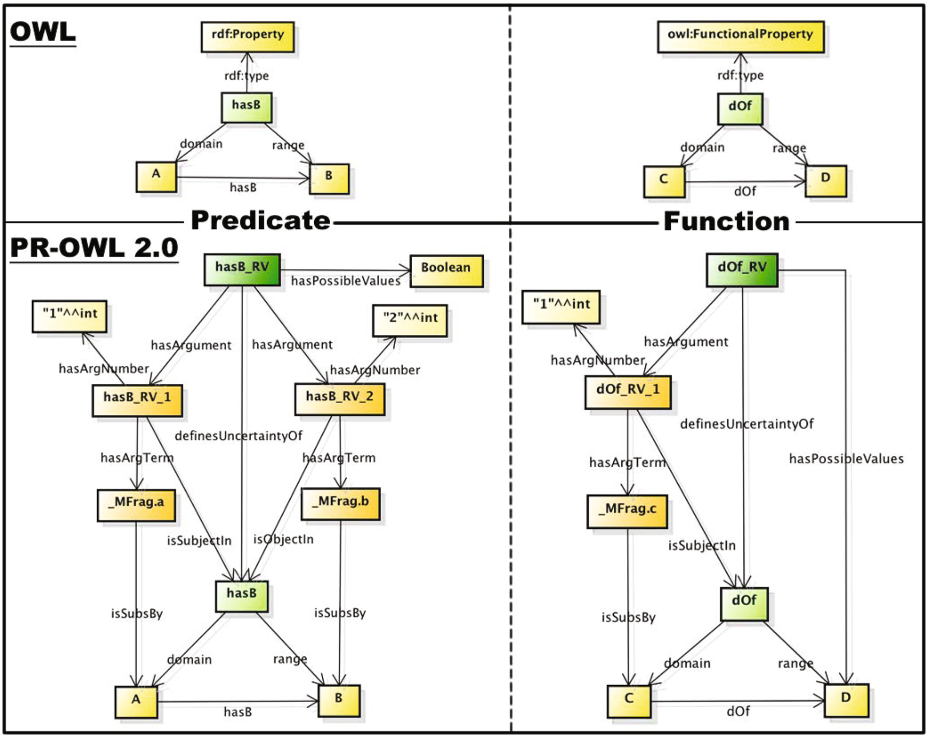

Fig. 11. The bridge joining OWL and PR-OWL

random variables in this new format can be found in [1, 3]. Here, a scheme is given in Figure 11 for the 2-way mapping between triples and random variables. Functions and predicates are considered as separate cases.

If a property (`hasB` or `dOf`) is defined in OWL, then its domain and range are already represented (`A` and `B`; `C` and `D`, respectively). The first thing to be done is to create the corresponding RV in PR-OWL (`hasB_RV` and `dOf_RV`, respectively) and link it to this OWL property through the property `definesUncertaintyOf`.

For binary relations, the domain of the property (`A` and `C`, respectively) will usually be the type (`isSubsBy`) of the variable (`_MFrag.a` and `_MFrag.c`, respectively) used in the first argument (`hasB_RV_1` and `dOf_RV_1`, respectively) of the RV. For n-ary relations see example given earlier in this Section on the RV `hasEducationalLevel(person,advisor)` and also [1, 3].

If the property is non-functional (`hasB`), then it represents a predicate that may be true or false. Thus, instead of having the possible values of the RV in PR-OWL (`hasB_RV`) being the range of the OWL property (`B`), it must be `Boolean`. So, its range (`B`) has to be mapped to the second argument (`hasB_RV_2`) of the RV, the same way the domain (`A`) was mapped to the first argument (`hasB_RV_1`) of the RV. On the other hand, if the the property is functional (`dOf`), the possible values of its RV (`dOf_RV`) must be the same as its range (`D`).

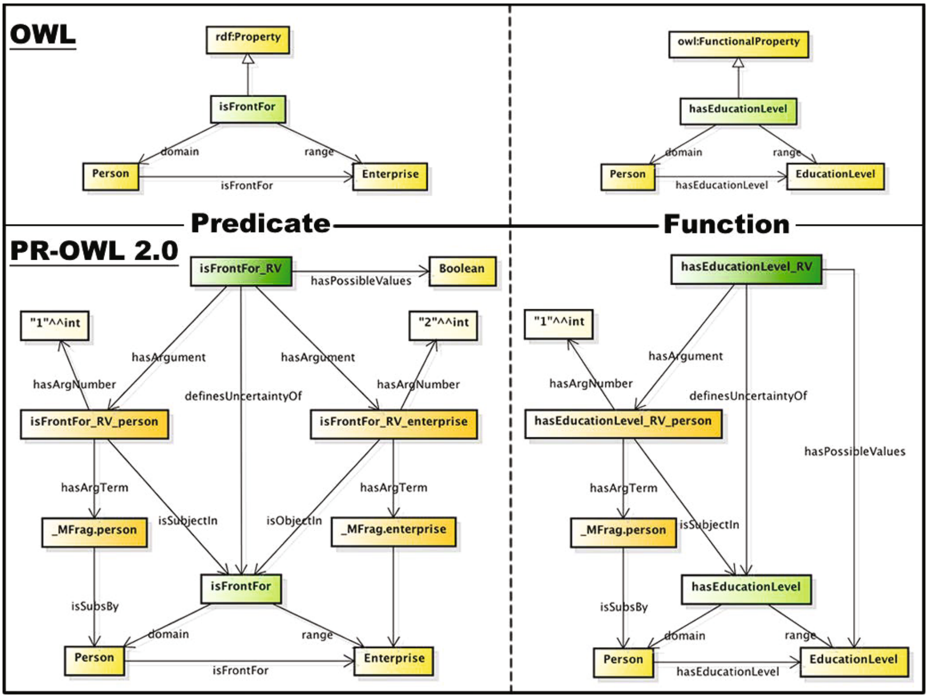

Fig. 12. Example of binary RVs mapping to OWL properties for both predicate and function

It is important to note that not only is the RV linked to the OWL property by `definesUncertaintyOf`, but also its arguments are linked to their respective OWL properties by either `isSubjectIn` or `isObjectIn`, depending on what they refer to (domain or range of the OWL property, respectively). This feature is especially important when dealing with n-ary relations, where each variable will be associated with a different OWL property (see explanation of Figure 10 earlier in this Section for details).

Finally, if the RV is already defined in PR-OWL with all its arguments and its possible values, the only thing that needs to be done is to create the corresponding OWL property, link the RV to it using `definesUncertaintyOf`, create the OWL properties for the arguments, if necessary, link them using either `isSubjectIn` or `isObjectIn`, depending on what they refer to (domain or range of the OWL property, respectively), and make sure that the domain and range of the property matches the RV definition, as explained previously.

Figure 12 presents examples of instantiations of the scheme just presented. In it we have the mapping of the RV `isFrontFor(person,enterprise)` to the OWL property `isFront`, which is a predicate, and the mapping of the RV `hasEducationLevel(person)` to the OWL property `hasEducationLevel`, which is a function.

The mapping described in this Section provides the basis for a formal definition of consistency between a PR-OWL probabilistic ontology and an OWL ontology, in which rules in the OWL ontology correspond to probability one assertions in the PR-OWL ontology. A formal notion of consistency can lead to development of consistency checking algorithms. For details on PR-OWL 2.0 abstract syntax and semantics see Carvalho [1].

5 Conclusion

With this mapping it is possible to not only resuse existing OWL semantics, but also automatically retrive available information from the mapped OWL ontology to use as evidence for probabilistic reasoning. This was not possible in PR-OWL 1.0.

Moreover, a scheme was given for how to do the mapping back and forth between PR-OWL random variables and OWL triples (both predicates and functions). Besides providing the scheme, a few examples were presented to illustrate how it works.

For full description of PR-OWL 2.0 abstract syntax and semantics, see Carvalho [1]. In it Carvalho also addresses other issues with PR-OWL 1.0 presented in [2].

Acknowledgments. The authors would like to thank the Brazilian Office of the Comptroller General (CGU) for their active support since 2008 and for providing the human resources necessary to conduct this research.

References

1. Carvalho, R.N.: Probabilistic Ontology: Representation and Modeling Methodology. PhD, George Mason University, Fairfax, VA, USA (2011)
2. Carvalho, R.N., Laskey, K.B., Costa, P.C.G.: Compatibility formalization between PR-OWL and OWL. In: Proceedings of the First International Workshop on Uncertainty in Description Logics (UniDL) on Federated Logic Conference (FLoC) 2010, Edinburgh, UK (July 2010)
3. Carvalho, R.N., Laskey, K.B., Costa, P.C.G.: PR-OWL 2.0 - Bridging the Gap to OWL Semantics. In: Bobillo, F., Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2008-2010/UniDL 2010. LNCS (LNAI), vol. 7123, pp. 1–18. Springer, Heidelberg (2013)
4. Carvalho, R.N., Matsumoto, S., Laskey, K.B., Costa, P.C.G., Ladeira, M., Santos, L.L.: Probabilistic Ontology and Knowledge Fusion for Procurement Fraud Detection in Brazil. In: Bobillo, F., Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2008-2010/UniDL 2010. LNCS (LNAI), vol. 7123, pp. 19–40. Springer, Heidelberg (2013)
5. Costa, P.C.G.: Bayesian Semantics for the Semantic Web. PhD, George Mason University, Fairfax, VA, USA (July 2005)
6. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty Modeling in Semantic Web Ontologies. In: Ma, Z. (ed.) Soft Computing in Ontologies and Semantic Web. STUDFUZZ, vol. 204, pp. 3–29. Springer, Heidelberg (2006)

7. Hayes, P., Rector, A.: Defining n-ary relations on the semantic web (2006), http://www.w3.org/TR/swbp-n-aryRelations/
8. Heinsohn, J.: Probabilistic description logics. In: Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence, UAI 1994, Seattle, Washington, USA, pp. 311–318. Morgan Kaufmann (1994)
9. Koller, D., Levy, A., Pfeffer, A.: P-CLASSIC: a tractable probabilistic description logic. In: Proceedings of AAAI 1997, pp. 390–397 (1997)
10. Laskey, K.B.: MEBN: a language for First-Order Bayesian knowledge bases. Artificial Intelligence 172(2-3), 140–178 (2008)
11. Laskey, K., Laskey, K.B.: Uncertainty reasoning for the World Wide Web: Report on the URW3-XG incubator group. URW3-XG, W3C (2008)
12. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence 172(6-7), 852–883 (2008)
13. Milch, B., Russell, S.: First-Order Probabilistic Languages: Into the Unknown. In: Muggleton, S.H., Otero, R., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 10–24. Springer, Heidelberg (2007)
14. Pan, J.Z., Stoilos, G., Stamou, G., Tzouvaras, V., Horrocks, I.: f-SWRL: A Fuzzy Extension of SWRL. In: Spaccapietra, S., Aberer, K., Cudré-Mauroux, P. (eds.) Journal on Data Semantics VI. LNCS, vol. 4090, pp. 28–46. Springer, Heidelberg (2006)
15. Poole, D., Smyth, C., Sharma, R.: Semantic Science: Ontologies, Data and Probabilistic Theories. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005-2007. LNCS (LNAI), vol. 5327, pp. 26–40. Springer, Heidelberg (2008)
16. Predoiu, L., Stuckenschmidt, H.: Probabilistic extensions of semantic web languages - a survey. In: The Semantic Web for Knowledge and Data Management: Technologies and Practices. Idea Group Inc. (2008)
17. Straccia, U.: A fuzzy description logic for the semantic web. In: Fuzzy Logic and the Semantic Web, Capturing Intelligence, pp. 167–181. Elsevier (2005)
18. Tao, J., Wen, Z., Hanpin, W., Lifu, W.: PrDLs: A New Kind of Probabilistic Description Logics About Belief. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 644–654. Springer, Heidelberg (2007)