

Automatic generation of Probabilistic Ontologies from UMP-ST model

Diego M. Azevedo¹, Marcelo Ladeira¹, Laécio L. Santos¹, and Rommel N. Carvalho^{1,2}

¹ Department of Computer Science
University of Brasília
Campus Universitário Darcy Ribeiro
Brasília, Distrito Federal, Brazil
diegomarques.azevedo@gmail.com
laecio@cic.unb.br, {mladeira, rommelnc}@unb.br
<http://www.cic.unb.br>

² Department of Research and Strategic Information
Brazilian Office of the Comptroller General (CGU)
SAS, Quadra 01, Bloco A, Edifício Darcy Ribeiro
Brasília, Distrito Federal, Brazil
rommel.carvalho@cgu.gov.br
<http://www.cgu.gov.br>

Abstract. The Uncertainty Modeling Process for Semantic Technologies (UMP-ST) is an incremental and iterative approach that covers the difficulty in maintaining and evolving existing POs [5]. It is a general methodology for the majority of the existing semantic technologies which support uncertainty. One of them is the Probabilistic OWL (PR-OWL), which is a language for representing Multi-Entity Bayesian Networks (MEBN). The modeling of a PO using UMP-ST methodology and MEBN/PR-OWL representation is supported by UnBBayes, a framework for building probabilistic graphical models and performing plausible reasoning. Although there is a guidance described by UMP-ST to model a PO, the implementation of a PO is painful and repetitive. Nowadays, the user needs to build the ontology from the zero in a specific technology, even if the user models the PO in UMP-ST. A proper integration that helps the user to implement the PO such as an intermediate structure makes implementation easier than build the PO from the zero and speeds up the modeling process. This paper proposes an algorithm to generate POs based on the mapping of elements of an UMP-ST model into a MEBN/PR-OWL representation. This approach is basically an extension of the UMP-ST to generate POs to an specific formalism, that is, the MEBN/PR-OWL language. The algorithm proposed is implemented as a Java plug-in for the UnBBayes.

Keywords: Automatic Generation, Probabilistic Ontology, Uncertainty Modeling Process, UMP-ST, POMC, Fraud Detection, MEBN, UnBBayes.

1 Introduction

The limitation of OWL in representing uncertainty are covered by many approaches based on different formalisms, one of them is the Probabilistic Web Ontology Language (PR-OWL) [2]. PR-OWL is based on Multi-Entity Bayesian Networks (MEBN), a formalism that joins the expressiveness power of First Order Logic to represent a domain and the ability to perform plausible reasoning of Bayesian Networks.

Although probabilistic reasoning is one of the most promising approaches to deal with uncertainty in ontologies, little support was giving because of the difficulty level to execute and replicate probabilistic ontologies. Carvalho [1], focusing on how to design this kind of probabilistic ontologies, proposes a methodology based on the Unified Process (UP) to make the working related to build this models less complex. The Uncertainty Reasoning Process for Semantic Technologies (URP-ST) is a framework that guides this working process through a series of steps as modeling, populating the knowledge base, and making inferences on the model.

The Uncertainty Modeling Process for Semantic Technologies (UMP-ST) is an incremental and iterative approach to model probabilistic ontologies through a set of disciplines. Carvalho defines Probabilistic Ontology Modeling Cycle (POMC) as a project lifecycle composed by four disciplines to handle the uncertainty modeling and reasoning processes. These disciplines are composed by Requirements, Analysis & Design, Implementation, and Test [1]. The URP-ST methodology is supported by UnBBayes [1, 6].

UnBBayes¹ is an open source framework with support to plug-ins that offers support to many methods with different formalisms based on Bayesian Networks (BN) [7]. One of them gives support to UMP-ST modeling process. Basically, the UMP-ST plug-in supports the Requirements and Analysis & Design disciplines [5]. The Implementation and Test disciplines as well as others steps of the URP-ST, such as populating the knowledge base and making inferences, can be done in UnBBayes using the MEBN/PR-OWL plug-in.

Although UnBBayes gives support to URP-ST, it does not have a link between modeling and implementing disciplines. This makes the developing process more painful because the effort needed to build the same model switching between the UMP-ST and in PR-OWL/MEBN plug-in, without a proper integration. This paper presents in detail how elements defined in UMP-ST are mapped to PR-OWL/MEBN elements and how is made the automatic generation of probabilistic ontology in PR-OWL from the UMP-ST plug-in extension.

This paper is organized as follows: Section 2 presents fundamental knowledge of URP-ST, UMP-ST, PR-OWL and MEBN. Section 3 describes UnBBayes and the plug-ins developed for PR-OWL and UMP-ST. Section 4 defines how it is possible to generate PR-OWL² ontologies with UMP-ST models and uses the Fraud Detection proposed by Carvalho [1] as a case study. Section 5 presents some concluding remarks.

¹ <http://sourceforge.net/projects/unbbayes/>

2 Fundamental Knowledge

This section presents basic concepts related to URP-ST, UMP-ST, PR-OWL/PR-OWL 2 and MEBN, providing an approach about these formalisms and their application in the PO context.

2.1 URP-ST

Uncertainty Reasoning Process for Semantic Technologies was proposed by Carvalho [1] as an approach for modeling a probabilistic ontology and using it for plausible reasoning in applications that use Semantic Technologies. It is divided into three steps: modeling the domain, populating the model with data, and using both the model and the data available, i.e Knowledge base, for reasoning. The model describes how different concepts in our ontology interact under uncertainty by knowing facts that will guide the construction process. Once the model is available, reasoning can be done when the model is populated with data, i.e entities and evidence. Then it is possible make inferences to answer queries from the users.

2.2 UMP-ST

Uncertainty Modeling Process for Semantics Technologies was proposed by Carvalho [1] as an iterative and incremental methodology for modeling POs. It describes the first phase of the URP-ST and it is compounded by four major disciplines: Requirements (functional and non-functional), Analysis & Design, Implementation, and Test. The modeling process described by UMP-ST is also consistent with the bayesian network modeling methodology [9][10].

Like in the UP, all disciplines in each iteration of the UMP-ST cover a set of requirements, building deliverable versions of the model. The idea behind every iteration is to take advantage of what has been learned during the modeling of earlier versions. It comes from discovering new requirements, rules, entities, and relations from previous versions.

Following the steps described by the UMP-ST, it is possible to build a Requirements Traceability Matrix (RTM) defined by traceability of work products of different disciplines. It can describe which requirement item is being implemented in each design object or implementation object. I.e, an entity and relationship defined in Analysis & Design discipline can be linked to goal described in the first discipline. Traceability promotes an easy identification of objects that should be analyzed or modified when implementing changes and also ensures that all requirements are covered.

The interaction process covered by the UMP-ST is depicted in Probabilistic Ontology Modeling Cycle (POMC), illustrated in Figure 1. This cycle defines the major outputs from each discipline and the natural order in which outputs are produced. Although there is a modeling order, it is not necessary to occur in a rigid and sequential way like the waterfall model. The interactions between the disciplines are not restricted to the arrows presented.

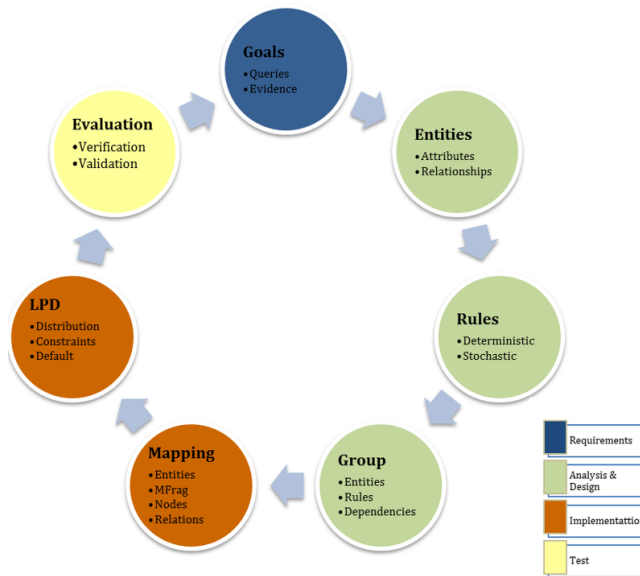


Fig. 1. POMC flow.

The Requirements discipline defines objectives that must be achieved by reasoning in PO. For each goal, it is necessary to define queries that will be answered with the aid of evidence.

Analysis & Design discipline defines classes of entities, their attributes, relationships, and rules. It will be grouped to isolate characteristics of the domain and to facilitate the visualization of dependencies that will guide implementation of the PO model in the next step.

In the Implementation discipline, entities, relationships, attributes, rules, and groups will be mapped to their corresponding concepts in a specific formalism that allows uncertainty representation. In our case study, the mapping is done to PR-OWL 2 format.

In the Test discipline, the user will validate and verify the model, in which it is observed whether the resulting ontology has an expected behavior and whether all requirements have been implemented.

2.3 MEBN and PR-OWL

Multi-Entity Bayesian Networks (MEBN) was proposed by Laskey [4] as a language for representing probabilistic knowledge. It extends Bayesian networks by incorporating the expressive power of first-order logic, and expands first-order logic by adding a way to specify probability distribution through its random variables. Thus, MEBN expands Bayesian network approach to a domain where the number of random variables (RVs) involved is unknown.

MEBN describes knowledge as a collection of MEBN Fragments (MFrag), organized into MEBN Theories (MTheories). Each MFrag represents a repeatable pattern of knowledge that can be instantiated as many times as needed to form a BN addressing a specific situation. Thus, it can be seen as a template for building fragments of a Bayesian network. MFrag are instantiated by binding their arguments to domain entity identifiers to create instances of RVs. Each MEBN fragment has three kinds of RV: context, resident, and input. Context RVs are conditions that guarantee the consistency of the distributions defined in a MFrag. Input RVs influence the distributions of other RVs in a MFrag, but their distribution are defined in their home MFrag. Resident RVs are defined by specifying their Local Probability Distribution (LPD) in their home MFrag. Each LDP of Resident RV is conditioned on the values of instances of its parents.

An MTheory is a set of MFrag that collectively satisfy conditions of consistency ensuring the existence of a unique joint probability distribution over instances of its random variables.

The instantiation of its RVs by bidding their arguments to domain entity identifiers results in a Situation-Specific Bayesian Network (SSBN), where the RVs of MFrag will become RVs in an ordinary Bayesian network. Then, these SSBNs can use regular BN inference engines to answer the query.

PR-OWL is a language for representing uncertainty knowledge in a principled, structured, and shareable way proposed by Costa [2]. It was developed as an extension of OWL, using MEBN to allow applications to perform plausible reasoning and requiring MEBN inference engine to process additional syntax based on Bayesian network probabilistic inference. This language consists of a set of classes and subclasses that collectively form a framework for building probabilistic ontologies [6].

PR-OWL 2 is an extension of PR-OWL proposed by Carvalho [1], providing a consistent mapping between OWL concepts and data types and PR-OWL. It facilitates the construction of hybrid ontologies with deterministic and probabilistic statements.

3 UnBBayes and the plug-in for UMP-ST

UnBBayes is an open-source JavaTM application distributed in GPL license. It has been developed since 2000 and was created for building probabilistic graphical models and performing plausible reasoning [7]. In mid-2010, UnBBayes had its architecture modified to give support to plug-in extension form. This facilitates its expansion into new formalisms. In core version, UnBBayes implements Bayesian network (BN) and Inference Diagrams (ID). Multiply-Sectioned Bayesian Network (MSBN), Object-Oriented Bayesian network (OOBN), Hybrid Bayesian Network (HBN), MEBN, PR-OWL, and PR-OWL 2 as others formalisms are supported by adding plug-ins extension.

MEBN implementation in UnBBayes is saved using the PR-OWL format. It was the first implementation of MEBN formalism in the world. In order to popularize PR-OWL language and MEBN formalism, UnBBayes provides a GUI for

editing ontologies making inference through the creation of SSBNs. PR-OWL 2 was implemented to allow the creation of probabilistic ontologies from deterministic ontologies in OWL. PR-OWL 2 plug-in was developed to allow the user to edit both probabilistic and deterministic parts of ontologies. It is possible because Protegé has been integrated into UnBBayes to design the deterministic part.

UMP-ST plug-in was implemented to solve three problems: the complexity in creating POs; the difficulty in maintenance and evolution of existing POs; and the lack of a centralized tool for documenting POs [5]. One of the key features in this plug-in is the implicitly creation of the traceability matrix by linking the objects created during the disciplines to the requirements they relate to. The plug-in allows the user to work with the Requirements and Analysis & Design disciplines. However, to allow the automatic generation of a PO from the UMP-ST model, it is necessary select a specific language and formalism that will be mapped in the Implementation discipline. MEBN and PR-OWL 2 were chosen because UnBBayes has support to both formalisms as well a plug-in for UMP-ST.

4 Automatic generation of probabilistic ontologies from UMP-ST models

After the modeler defines the elements in the Analysis & Design discipline, it is necessary to choose a specific language that allows uncertainty representation in semantic technologies to implement the model designed. This work describes how to generate a PO in PR-OWL 2 using UnBBayes from the model designed in UMP-ST plug-in. One of the main advantages is the traceability between the elements defined in the Requirements and Analysis & Design disciplines and the PR-OWL elements in the Implementation discipline.

To automatic generate a model in PR-OWL 2, it is necessary to start by mapping the entities, their attributes, and relations to the chosen language. Figure 2 presents how the elements created in UMP-ST can be mapped to PR-OWL 2, which essentially uses MEBN terms [1]. The designed model needs to be simplified, since UnBBayes has some limitations. E.g., the lack of support for a type hierarchy.

As represented in Figure 2, the elements of UMP-ST (in blue) at the left side are mapped to MEBN terms present at the right side by considering the limitations of UnBBayes. Thus, the entities in the model will be mapped to an entity in PR-OWL 2. The attributes and relationships of the model are characteristics that may be uncertain. In MEBN, it is represented by defining random variables (RVs), which can be a kind of context RV (in green), input RV (in gray) or resident RV (in yellow).

To define a RV in UnBBayes PR-OWL, first we need to define its home MFrag. However, it is not necessary with the PR-OWL 2 plug-in, because RVs are independent of MFrag and are defined globally. This makes the RVs definition easier compared with UnBBayes PR-OWL definition process [1].

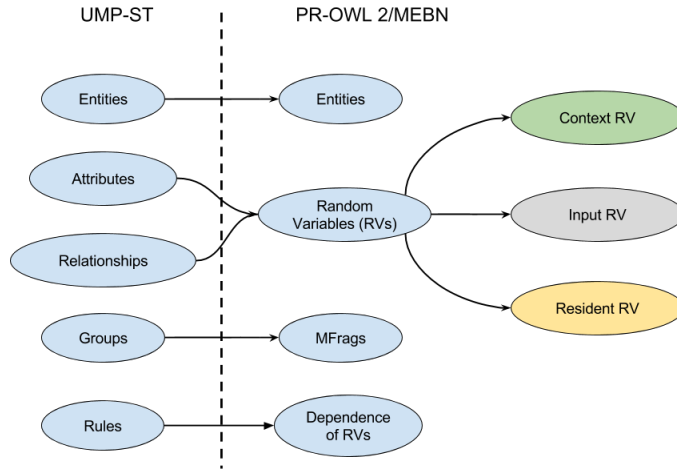


Fig. 2. Mapping of UMP-ST elements to PR-OWL 2.

In UMP-ST, the attributes and relationships as others elements defined in the Analysis & Design discipline are separated in different groups. Also in MEBN, each RV has a home MFrag into which it is defined the local probability distribution. So on, in Figure 2 the groups (on the left side) will be mapped as MFrag (on the right side).

As described in Figure 3, relationships in UMP-ST (on top) can be defined as a relation of one or more entities, being a predicate or a function, depending of their context, e.g. `livesAtSameAddress(Person, Person)`. In MEBN theories, a RV (in yellow) takes arguments that refer to entities of the domain of application [4]. These arguments in UnBBayes have `OrdinaryVariable` type and they are represented by a Context RV (in green) with Boolean range as the built-in RV `isA(resource, class)`. `resource`, in this type of RV, represents individuals of `OrdinaryVariable` and `class` represents entities related to the domain of application [8, 1].

Relationships defined in the UMP-ST will be mapped to (input or resident) RVs (in yellow) and for each entity present in relationship, context RVs (in green) with Boolean range will be created as the built-in RV `isA(resource, class)`. This will be necessary to guarantee the unique joint probability distribution over the RVs when the model is instantiated.

Even though UMP-ST is a generic modeling process for the majority of existing semantic technologies which support uncertainty representation, the chosen language does not offer a formal link between these terms. It is impossible for reasoners to identify that these terms are even linked. At best, it could only “guess” they are the same, since they have similar syntax [1].

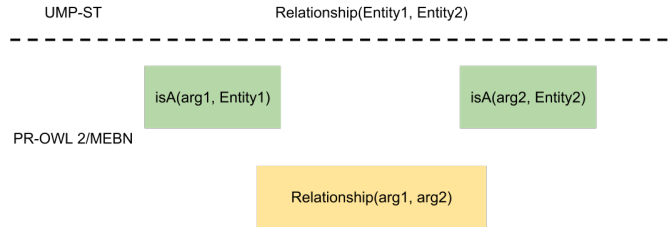


Fig. 3. Mapping a relationship in UMP-ST to a RV in PR-OWL 2.

Once all RVs are mapped, their relations and constraints can be defined by analyzing RVs dependencies described by the rules in the UMP-ST model [3, 1]. It is important to capture local consistence of the model by testing these rules even before implementing the model in a specific language by creating simple probabilistic model to evaluate whether it will behave as expected. However, rules in UMP-ST are basically described in textual form [3], which makes an automatic identification of the rule structure more complexity and subject to errors.

To make the rule structure understandable by computer and less subject to errors, an extension of the rule definition is proposed during the Implementation discipline. It is based on four steps. In the first and the second steps, the rule context (ordinary variables and necessary conditions) is specified. In the third and last steps, a casual relation to describe the dependence of RVs is defined. Figure 4 presents an example to illustrate this form of extension.

Considering the rule and the group description in the Analysis & Design discipline, each rule is related to only one group and each group can have one or more rules. Therefore, the relation of dependence over the RVs described by the rule is restricted to the group in which it is declared. This relation of dependence defined in a specific group will contribute to set the local probability distribution over the RVs by describing who it is cause and who it is effect.

The new rule definition extension allows mapping the ordinary variables and the necessary conditions to context RVs and the cause and the effect to (input or resident) RVs. For instance, in Figure 4 the ordinary variables `person` and `enterprise` are both related to the class `Person` and `Enterprise`. Thus, they are mapped in an MFrag to the `isA(person, Person)` and the `isA(enterprise, Enterprise)` context RVs. The necessary condition can be described by a first-order logic and also can be a RV as `isResponsibleFor(person, enterprise)` which is mapped to a context RV. This condition, in the MEBN fragment, will restrict the application of the probability distribution over its RVs to instances of `Person` that are responsible for the instances of `Enterprise`.

With all elements defined in the Analysis & Design already mapped to the PR-OWL 2 language, now it is necessary to distinguish input and resident RVs. An algorithm to select RVs based on their home MFragment and their relation of dependence is proposed. This algorithm is composed by three criterion of selections.

The **First criteria of selection** defines as resident RV the RV present only in its home MFragment and does not influence a probability distribution of other RVs as a possible input RV. E.g., `ownsSuspendedEnterprise(person)` in `Owns_Suspended_Enterprise` MFragment illustrated by Figure 5. This is done by the algorithm, by identifying the relationships that there are in just one group.

The **Second criteria of selection** defines as input RV in a MFragment the RV that also is defined as resident RV in another MFragment. E.g, in Figure 5, `isSuspended(enterprise)` has input type in `Owns_Suspended_Enterprise` MFragment, but first it is defined as resident RV in `Enterprise_Information` MFragment.

In the Second criteria of selection, the algorithm analyses the rules of each group and identify where the RVs are the cause and where they are the effect. If there is a RV that is cause and in other group it is defined as effect by a rule, then this RV which is cause will have input type and the other which is effect will have resident type. This is possible because in MEBN an input RV influences the local probability distribution of resident RV like a cause and effect relation, but first the RV which has input type needs to be a resident RV in its home MEBN fragment.

After analyzing the causal relation defined by the rules, it is possible to exist some RVs that still have undefined type. This kind of RVs can be identified by analyzing the RVs which are not defined by a rule but they still participate in a MFragment. E.g, in Figure 5 in `Enterprise_Information` MFragment, the RV `isResponsibleFor(person, enterprise)` is not defined by some rule because in the group related to its MFragment, it is not a cause or effect. In this case, the algorithm considers that this kind of MFragment is the home MFragment of the RV analyzed and it will be defined as resident RV. Nevertheless, if there are RVs that still do not have a type, they will be added in a list of doubts.

The **Third criteria of selection** analyzes whether after first and second criterion of selection may exist some RVs that still have undefined type. So on, if there are RVs in the list of doubts, then each RV in doubt will be presented to the user decides manually whether this RV has input or resident type. The type of RV decided by the user will be added to a list of hypothesis and it will be saved in UMP-ST plug-in file.

Carvalho [1] proposes the probabilistic ontology for procurement fraud and detection and prevention in Brazil. This case study is modeled in PR-OWL 2 with URP-ST and it will be used to illustrate the generation process using the algorithm proposed.

To explain the problem described by Carvalho in the fraud and detection model, it is necessary to have in mind the principles established by Brazil's Law among equality of bidders. This Law prohibits the procurement agent from discrimination among potential suppliers. Therefore, the procurement agent cannot

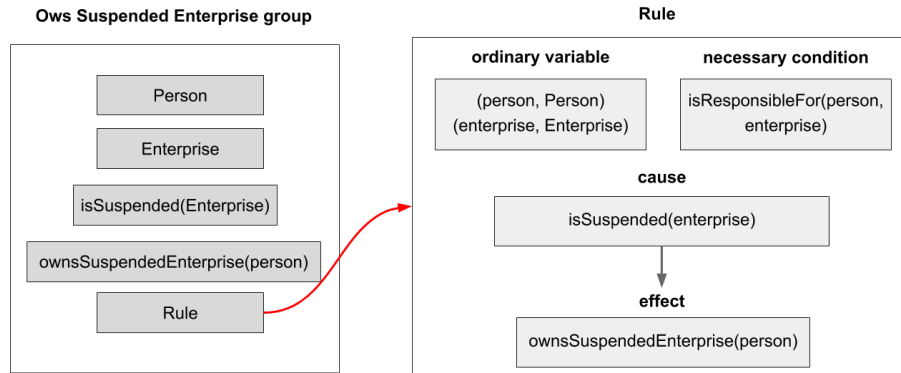


Fig. 4. Group and rule representation.

be related to the bidder or might feed information in order to favor the bidder's enterprise. Also, the committee for a procurement must be well prepared and must has a clean history with no criminal or administrative convictions. These constraints give rise to a set of goals, queries and evidences which will guide the modeling process.

During the Analysis & Design discipline, one of the goals covered by Carvalho identifies whether some person `ownsSuspendedEnterprise`. This relationship is defined as a relation between `Person` and `Enterprise` entities. However, to know whether some person `ownsSuspendedEnterprise`, it is necessary to known also the probability of the enterprise analyzed `isSuspended`. This rule only will be valid whether the person `isResponsibleFor` the enterprise. Figure 4 illustrates this rule in extension form.

This goal can be analyzed and designed in two groups. `isSuspended` and `isResponsibleFor` can be grouped as `Enterprise Information` which also will contain the entities that they relate to. The other group will contain the elements necessary to describe the rule defined before. So on, it needs to have `isResponsibleFor`, `isSuspended` and `ownsSuspendedEnterprise` as relationships and `Person` and `Enterprise` as entities.

After modeling the elements during Analysis & Design discipline, to generate the PO, first each element will be mapped as described in Figure 2. Then, the algorithm will run the sequence of steps presented below to define the RVs type based on rules and groups distribution.

- i** It is identified whether the model had already been generated by analyzing the list of hypothesis.
- ii** Case the list of hypothesis is not empty, then each RV listed will have the type defined by the user hypothesis.
- iii** First criterion of selection is executed and the RVs present in just one MFrag will be defined as resident.

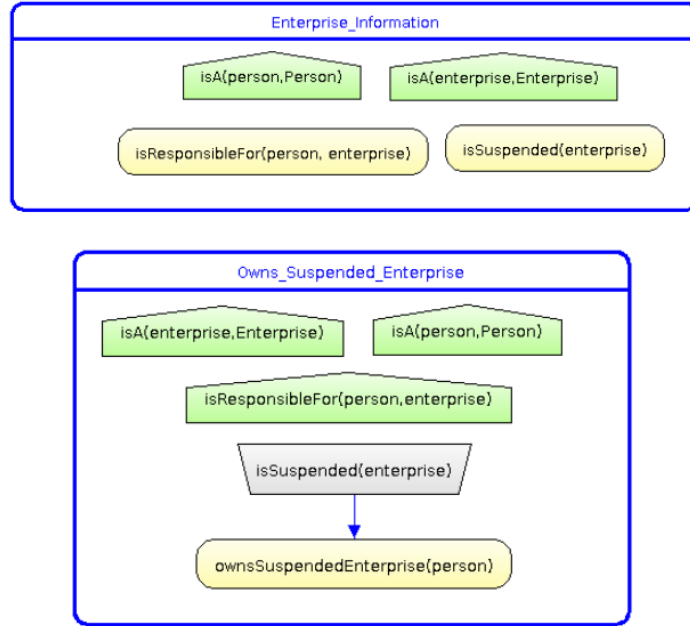


Fig. 5. MFragment generated after mapping elements of Analysis & Design stage.

- iv Second criterion of selection is executed. If the RV is defined as cause in a rule and in other group it is defined as effect, then it will have input type in MFragment that it was cause and in the other that it was effect, it will have resident type.
- v Case there are RVs that were not declared in any rule, but they were defined in a group, then they will have resident type.
- vi Third criterion of selection will define the type of the RVs that still are in doubt by presenting them to the user decides.

The PR-OWL 2 format generated by the algorithm does not define the local probability distribution presents in each resident RVs. Therefore, the model generated is not complete. The user needs to define the local probability distribution by editing the code generated in MEBN plug-in supported by UnBBayes. After that, the PO is validated and verified by analyzing whether the ontology has a expected behavior and whether all requirements have been implemented. Case one requirement has not been implemented, the user will edit the PO in the MEBN plug-in, but if it is necessary to change some rule, then it is possible to generate other PO in PR-OWL 2 format from the UMP-ST plug-in extension.

5 Conclusion

It is possible to generate automatically a PO in PR-OWL 2 format able to be read and edited by UnBBayes. Although there is not a formal link between the UMP-ST elements and the output format, the proposed mapping of the elements defined in Analysis & Design discipline to PR-OWL 2 format was based on the related concepts of each element of both representations.

The new definition of the rule during the generation process is necessary to make its format understandable by a computer. This is possible by defining the ordinary variables, the necessary conditions and the causal relation over its RVs. These elements will be mapped to MEBN/ PR-OWL 2 and, by executing the algorithm, the model defined by UMP-ST plug-in will generate a MEBN theory.

The generation of the PR-OWL 2 code by the extension of the UMP-ST plug-in speeds up and facilitates the creation of a MEBN model (PR-OWL 2 PO). The development of this extension is on implementing phase. It is developed in *JavaTM* and relies on Java Plugin Framework² (JPF) version 1.5.1.

References

1. R. N. Carvalho, "Probabilistic ontology: Representation and modeling methodology," PhD, George Mason University, Fairfax, VA, USA, 2011.
2. P. C. G. Costa, "Bayesian semantics for the semantic web," PhD, George Mason University, Fairfax, VA, USA, Jul. 2005.
3. R. M. de Souza, "UnBBayes plug-in for documenting probabilistic ontologies using UMP-ST," B.S., University of Brasília, Brasília, Brazil, 2011.
4. K. B. Laskey, "MEBN: a language for First-Order bayesian knowledge bases," *Artificial Intelligence*, vol. 172, no. 2-3, pp 140-178, 2008.
5. R. N. Carvalho, M. Ladeira, R. M. de Souza, S. Matsumoto, H. A. D. Rocha, and G. L. Mendes, "UMP-ST plug-in: A tool for documenting, maintaining, and evolving probabilistic ontologies." in URSW, ser. CEUR Workshop Proceedings, F. Bobillo, R. N. Carvalho, P. C. G. da Costa, C. d'Amato, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, T. Martin, M. Nickles, and M. Pool, Eds., vol. 1073. CEUR-WS.org, 2013, pp. 15–26.
6. R. N. Carvalho, K. B. Laskey, P. C. G. da Costa, M. Ladeira, L. L. Santos, and S. Matsumoto, "UnBBayes: modeling uncertainty for plausible reasoning in the semantic web," in *Semantic Web*, gang wu ed. INTECH, Jan. 2010, pp. 1–28.
7. S. Matsumoto, R. N. Carvalho, M. Ladeira, P. C. G. da Costa, L. L. Santos, D. Silva, M. Onishi, E. Machado, and K. Cai, "UnBBayes: a java framework for probabilistic models in AI," in *Java in Academia and Research*. iConcept Press, 2011.
8. S. Matsumoto, "Framework based in plug-ins for reasoning with probabilistic ontologies," M.Sc., University of Brasflia, Brasflia, Brazil, 2011.
9. K. B. Laskey and S. M. Mahoney, "Network engineering for agile belief network models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 487–498, 2000. [Online]. Available: <http://portal.acm.org/citation.cfm?id=628073>
10. K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*, 1st ed. & Hall/CRC, Sep. 2003. Chapman

² JPF homepage: <http://jpf.sourceforge.net/>