Improving Probabilistic Rules Compilation using PRM

Gaspard Ducamp[1,2], Philippe Bonnard[2], Christian De Sainte Marie[2], Christophe Gonzales[1], and Pierre-Henri Wuillemin[1]

[1] LIP6 (UMR 7606), Sorbonne Université, 4 place Jussieu, 75005 Paris, France
prenom.nom@lip6.fr
https://www.lip6.fr/

[2] IBM France Lab, 9 rue de Verdun, 94250 Gentilly, France
philippe.bonnard@fr.ibm.com, csma@fr.ibm.com, gaspard.ducamp@ibm.com

Abstract. Widely adopted for more than 20 years in industrial fields, business rules offer the opportunity to non-IT users to define decision-making policies in a simple and intuitive way. To facilitate their use, systems known as Business Rule Management Systems have been developed, separating the business logic from the application one. While suitable for processing structured and complete data, BRMS face difficulties when those are incomplete or uncertain.

This study proposes a new approach for the integration of probabilistic reasoning in IBM Operational Decision Manager (ODM), IBMs BRMS, especially through the introduction of a notion of risk, making the compilation phase more complex but increasing the expressiveness of business rules.

Keywords: Business Rule · Business Rule Management System · Decision Making · Bayesian Networks · Bayesian Inference · Object Model · Probabilistic Relational Models.

1 General presentation

Business Rules Management Systems (BRMS), such as *IBM Operational Decision Manager* (ODM), have been introduced in the 90's to facilitate editing, authoring, deploying and executing business policies by domain users, in the form of conditions/actions rules. Syntactically close to the business language, these ease the translation of decision-making and business strategies, making them accessible to users with no programming experience. Besides the rule set, an object data model describes the different objects concerned by the rules, they are dynamically instantiated in a working memory during the execution of the program. The activation/execution of rules is managed by inference algorithms such as RETE [7, 5, 13].

The PhD thesis presented here builds upon work already carried out within IBM France Lab and the LIP6. It is based on the hybridization of business rules with probabilistic graphical models such as Bayesian networks [6] or as Probabilistic Relational Models [1].

A Bayesian network is a compact representation of a joint probability distribution over a set of random variables. These appear in the form of nodes in a direct acyclic graph (DAG) where the absence of arcs represent conditional independences. This type of structure is used as a decision-making tool in many expert systems and applications [10, 14].

Probabilistic Relational Models (PRM), on the other hand, are combining notions from Bayesian networks and from the paradigm of object-oriented languages [12]. The notions of random variables and conditional probabilities are added to those of classes, relations, interface, inheritance and instantiations. Such expressiveness makes it possible to answer the problems of reusability and scalability of graphical models [11].

Several studies including a thesis [4, 1] showed that a loose coupling between IBM's BRMS, on the one hand, and probabilistic graphical models (Bayesian networks initially, then PRM), on the other hand, allowed reasoning about uncertain data by introducing the notion of *probabilistic production rules*.

Figure 1 shows how those works fit into the ODM toolchain. Several semantic trees are generated after analyzing and checking the syntax of the user file describing the rules and the object model (*ARL*). In addition to producing the one describing the rules (*SemRuleset*) these studies proposed to use the object model to generate a probabilistic graphical model semantic tree (*SemPGM*). The probabilistic expression of rules are then rewritten to link ODM to an inference engine (aGrUM[1], jSMILE[2], ...) using the generated file (*PGM file*). Once the rules are rewritten, the models are compiled before being translated into bytecode in an archive (*JAR*), facilitating deployment and execution on the target machines.

[1] aGrUM (A GRaphical Universal Modeler) [15] is an open source C ++ library for manipulating (learning, modeling, inferring) graphical models and implementing O3PRM: http: //agrum.gitlab .io

[2] SMILE is a reasoning and learning/causal discovery engine for graphical models https://www.bayesfusion.com/smile-engine

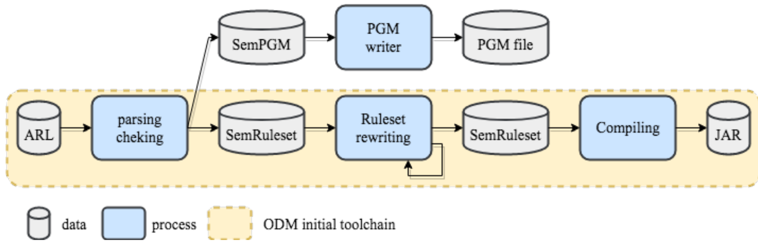

Fig. 1. ODM toolchain using probabilistic handling modules

A working example used in those studies is that of a fraud detection application, an organization being responsible for managing the reimbursement requests made by its clients according to their nature (type and cost of reimbursement). For this purpose, a set of healthcare professionals are in charge of validating the reimbursement requests made by customers.

Figure 2 shows an example of relation schema for PRM classes of this problem. A HealthcareProfessional class describes healthcare professionals based on several characteristics, including their age, location, gender, and the list of clients they manage (Subscriber), themselves characterized by an age and a list of reimbursements requests (Reimbursement).

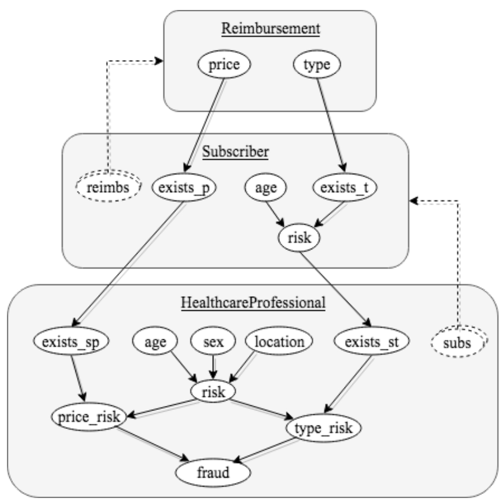

Fig. 2. Class dependency schema for the insurance example

Generated during compilation, this model allows to answer a first set of probabilistic rules. In the example below, we will trigger the action part when we detect a trusted healthcare professional based in Paris who has a client with a high risk of fraud.

Ex. 1. An example of probabilistic rule

```
rule detectInvoiceAnomaly{
    when{
      hp: HealthcareProvider(hp.location==Paris && probability(hp.risk==low)>.7);
      sub: Subscriber (probability(sub.risk==high)>.9) in hp.subs;
    } then { … }
}
```

The previous works raised two major issues:

- **Business user friendliness**: such rules can be difficult to define and to understand by a business user, expressing a probability on particular conditions requiring a deep level of knowledge of the probabilistic models used.
- **Performance**: ODM provides the ability for users to define different types of conditions (for example, using filters, aggregators, and nested conditions). Neither these constructions, more complex, nor their impacts on the performances have been studied.

Incidentally, a new inference algorithm [3] has been studied during the previous thesis but:

- the interest of the use of this algorithm comes from the assumption that the incremental modifications of the working memory were not going to have big impacts on the structure of the junction tree used for the inferences, important savings of calculations could therefore be realized. It is, consequently, especially adapted to large connected structures;
- the algorithm works with PRM, but retranscribed in the form of Bayesian networks (so-called "grounded"). The interest of PRMs is greatly reduced since the structural redundancies of these grounded networks, which are encoded in the PRMs that generated them, are not exploited, which reduces the efficiency of the inferences;

We notice that the graphical model built during the compilation of ODM takes into account a probabilized model of the relations between the objects of the working memory but it does not adapt itself to the probabilistic queries defined in the rules.

2 A new definition of probabilistic production rules

To address the business user friendliness issue raised above, we need to redefine the treatment of uncertainty in the expression of rules by replacing the probability thresholds attached to single variables by a notion of acceptable risk on the evaluation of the conditions of the rule as a whole. The action part of a rule will therefore be executed only if the set of conditions is verified with a probability greater than the defined acceptable risk. This will allow our probabilistic rules to be both more complex and intuitive, but it requires a redefinition of the rules compilation phase to redistribute the overall risk to each individual condition.

In the rule below, a rewriting of the previous one, the **then** part will only be triggered if the probability that conditions c1 and c2 are true is greater than 80%. Using such kind of formulation will ease the use of probabilistic rules by business users.

Ex. 2. A new form of probabilistic rule

```
rule detectInvoiceAnomaly {
   when {
      hp: HealthcareProvider (hp.location==Paris && hp.risk==low);        (c1)
       sub: Subscriber (sub.risk==high) in hp.subs;                       (c2)
   } [with probability >.8] then {…}
}
```

To achieve this, it is necessary to generate the PRM not only from the object models but also from the set of rules (see Figure 3). All the predicates over attributes of a probabilistic object, such as *hp.risk==low*, will be added in our graphical model in an extension of the concerned class (red classes in Figure 3). This requires studying the different forms that conditions can take (aggregations, filters, nesting, ...) and their predicates. Hence, a feasibility study must be carried out to define the theoretical and practical limits of such a method.

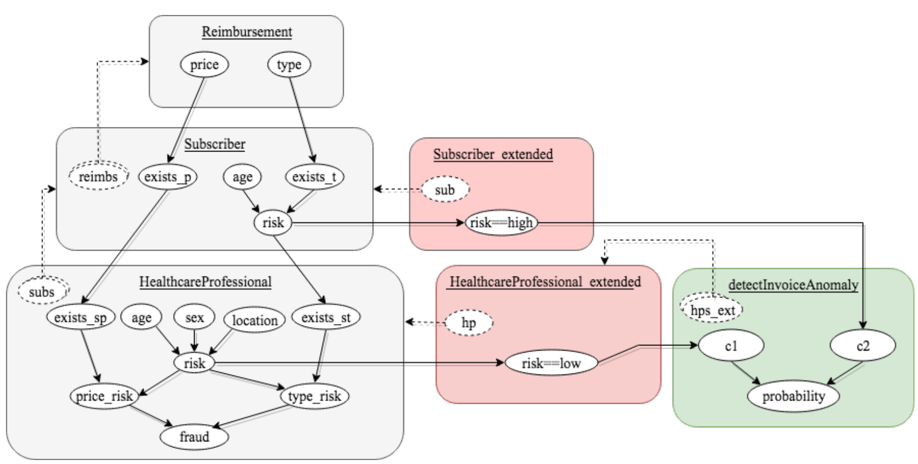

Fig. 3. Class dependency schema after enhancement based on example 2

The rules will then be rewritten to take into account our new probabilistic queries (on the *probability* attribute of the *detectInvoiceAnomaly* class, for example). Such manipulations could, however, change the efficiency of the pattern matching performed by ODM for the selection of rules to be performed, therefore their impact should be studied.

On the PRM side, the dynamic instantiation of classes representing rules (in green in Figure 3) as well as the simplification of the model generated during compilation will be at the center of the project (by taking into account that some of the data sources will be deterministic). In our example the location of a healthcare professional is considered certain, thus we don't need to include it in our new PRM.

Due to the highly deterministic aspect of the user models, we will assume that the instantiation of our PRMs will yield a large number of small disjoint subgraphs (one per health professional, for example), which would require working on a new inference algorithm taking advantage of this structural redundancy.

This development work is based on what has been done in the Bayesian Insight Service (BIS) plugin developed as part of the previous thesis. The figure 4 illustrates how our new module, PRIME (*Probabilistic Reasoning Insight ModulE*), fits into the ODM compilation chain. It intervenes directly during the process of rewriting the semantic tree describing the rules (*SemRuleset*) but, unlike BIS, extends the definition and optimization of the graphical model from the rules before rewriting them (via the *PRM enhancement* process).

Once the rules are rewritten, the graphical model is serialized in order to be usable by a probabilistic engine in parallel with ODM.

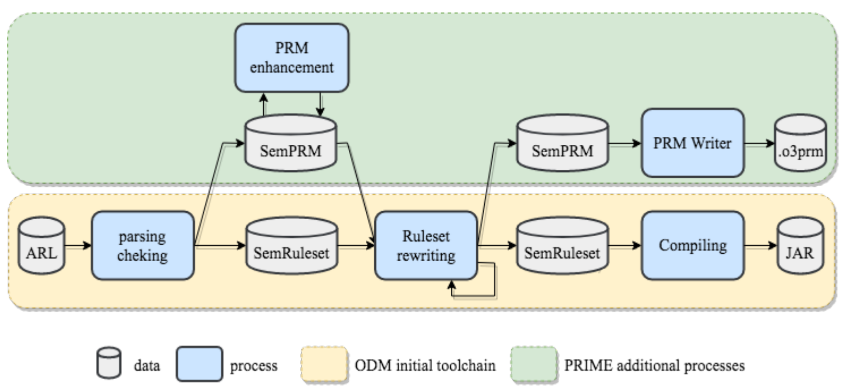

Fig. 4. ODM toolchain using PRIME

As part of our work, ODM uses aGrUM for probabilistic calculations and manipulation of PRMs. The complexity of the compilation, as well as performance problems during the execution, might require working on an extension of this library (advanced aggregations, functional nodes, parametric nodes, ...)

3 Application and extensions

This work will provide an extension of the ODM business rule definition language to probabilistic concepts. Syntactically and semantically close to natural language, it will allow non-computer scientists to define probabilistic business rules in an non-ambiguous and intuitive way.

A study on the use of temporal models via an extension of the PRM modeling capabilities in O3PRM[3] could be done in a second step, which would allow a similar reasoning but in the context of the treatment of complex events (Complex Event Processing, CEP [9]). The use of rules with temporal expressions is at the core of the reasoning capabilities of IBM Decision Server Insights, a CEP application developed by IBM. This study would therefore be useful at the theoretical as well as the technical and industrial levels.

Finally, one of the advantages of the expressiveness of PRM is that it is possible to take into account the structural uncertainty of a model [8]. In doing so, a study on a predictive extension of ODM could be carried out. The system would then be able to reason beyond the working memory, allowing to define new types of rules (anticipation ones, for example).

References

1. Agli, H.: Uncertain Reasoning for Business Rules. PhD thesis. Sorbonne University (2017)
2. Agli, H., Bonnard, P., Gonzales, C., Wuillemin, P.-H.: Business Rules Uncertainty Management with Probabilistic Relational Models. RuleML16, Stony Brook, New York, US (2016). https://doi.org/10.1007/978-3-319-42019-6_4
3. Agli, H., Bonnard, P., Gonzales, C., Wuillemin, P.-H.: Incremental Junction Tree Inference. IPMU16, Eindhoven, Netherlands (2016). https://doi.org/10.1007/978-3-319-40596-4_28
4. Ait-Kaci, H. and Bonnard, P.: Probabilistic Production Rules. Technical report. IBM. (2011)
5. Berstel - Da Silva, B.: Verification of Business Rules Program. PhD thesis. University of Freiburg (2012)
6. El Ghali, A. and Bonnard, P. and El Ghali, K. and Hromada, D. and Ait-Kaci, H.: Règles de production et réseaux bayésiens pour lextraction de mots clés. Technical Report. (2012)
7. Forgy, C. L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. In: Artificial Intelligence, vol 19, pp.17–37. (1982) https://doi.org/10.1016/0004-3702(82)90020-0
8. Getoor, L. and Koller, D. and Taskar, B. and Friedman, N.: Learning Probabilistic Relational Models with structural uncertainty. In: Proc. of the AAAI-2000 Workshop on Learning Statistical Models from Relational Data,Technical Report WS-0006, pp. 13–20. AAAI Press, Menlo Park, CA (200)
9. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison Wesley (2002)

[3] O3PRM is a framework developed in the framework of aGrUM allowing the manipulation of PRM: http://o3prm.gitlab.io

10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufman (1988).
11. Medina-Oliva, G., Weber, P., Levrat, E., Iung, B.: Use of probabilistic relational model (PRM) for dependability analysis of complex systems. In: IFAC Symposium on Large Scale Systems: Theory and Applications, LSS 2010. Villeneuve d'Ascq, France. (2010)
12. Torti, L., Wuillemin, P.-H., Gonzales, C.: Reinforcing the Object-Oriented Aspect of Probabilistic Relational Model. In: PGM 2010 - The Fifth European Workshop on Probabilistic Graphical Models, pp. 273–280. Helsinki, Finland. (2010)
13. Reteplus, the ODM rule execution mode based on the Rete Algorithm. https://www.ibm.com/support/knowledgecenter/SSQP76_8.9.0/com.ibm.odm.dserver.rules.designer.run/optimizing_topics/tpc_opt_reteplusalgo.html
14. Weber, P. and Medina-Oliva, G. and Simon, C. and Iung, B.: Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. In: Engineering Applications of Artificial Intelligence, vol. 25, n. 4, pp. 671 – 682. (2012) https://doi.org/10.1016/j.engappai.2010.06.002
15. Gonzales, C. and Torti, L. and Wuillemin, P.-H.: aGrUM: a Graphical Universal Model framework. In: Proceedings of the 30th International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems. International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems, Arras, France. (2017)