Krzysztof Kaczor*, Grzegorz J. Nalepa*

A Critical Overview of Methods for Rules Interoperability. Position Paper**

1. Introduction

Rule-Based Systems [15, 7] constitute a mature technology in the field of Artificial Intelligence [23]. Over the years, they were applied in many domains like medicine, engineering [9], decision support [16]. Even NASA created its own language for building Rule-Based Systems called C Language Integrated Production System (CLIPS). It became one of the most commonly known and used language for example image processing, image recognition. Currently its newer implementation called Jess [6] gained a lot of attention. Nowadays, these systems do not fit into directions and trends of the scientific research in their classical form. However, many ideas, algorithms and solutions that are applied in new technologies such as Business Rules, Semantic Web are derived from Rule-Based Systems.

Business Rules (BR) [22] are one of the latest application field of the classical rules. They allow for defining logical aspects of business, and are used by tools such as Drools or OpenRules. BR semantics is defined by Semantics of Business Vocabulary and Business Rules (SBVR) standard [21]. The goal of SBVR is to provide the basis for formal and detailed natural language declarative specifications of business entities (vocabulary) and policies (rules). The formal representation is based on several logics including first order logic, alethic modal logic and deontic logic. Furthermore, it adapts model theoretic interpretations for semantic formulations.

The features of the SBVR are provided by very complex and vague meta-model. It consists of four submodels which define the representation and meaning of the business and business rules vocabulary. Due to its complexity, SBVR is hard to use in practical applications. What is more, the support for SBVR have to involve application of the dedicated parsers of controlled english.

The Semantic Web [2] initiative aims at adding meaning to the data in order to make it machine processable. This allows for using more advanced searching and planning

---

* AGH University of Science and Technology, Krakow, Poland

mechanisms, which work in an automated way. Hence, the Semantic Web mainly focuses on ontology modeling. Ontologies are designed using the OWL language [8] which provides an XML-based syntax and semantics described by Description Logics (DL) [3].

One of the main problems in Semantic Web is efficient reasoning and application of rules. One of the approach to using rules in Semantic Web is the Semantic Web Rule Language (SWRL) [10]. This language combines the OWL sublanguages (OWL DL and OWL Lite) with the RuleML [26] (Datalog). It includes an abstract syntax for Horn rules in the mentioned sublanguages of OWL. There are three approaches to implement inference engines for SWRL: Hoolet, Bossam and Pellet. From the mentioned approaches, the Pellet[1] reasoner provides the best support for SWRL. Nevertheless, this support is still limited to DL-Safe subset of SWRL. The new OWL 2 RL Profile is a rule subset of OWL 2 that allows for using simple rule semantics with OWL 2. However, the current support for this feature in DL reasoners as well as Semantic Web using them is very limited [1].

The Semantic Knowledge Engineering (SKE) [18] approach provides a formalized model for classic rules representation [20]. It is based on the Attributive Logic with Set of Values over Finite Domains (ALSV(FD)) logic [17]. The SKE design methodology is supported by a set of dedicated tools allowing for visual rules modeling, quality analyzing and automated implementation [12]. The underlying logic provides support for complex types and gener-alized attributes which can take set as a value. This makes the logic more expressive than other rule base representations [19]. The SKE methodology provides logic-based rule language called Extended Tabular Trees ver. 2 (XTT2) [20]. This language is supported by the dedicated tool HQEd [11] and allows for visual and agile rule base modeling.

The XTT2 is based on the ALSV(FD) logic, however it does not use it in efficient way. It provides a several limitations: it does not support complex types, it allows for using only constant expressions within conditional part of rule. Moreover, SKE assumes that the under-lying ALSV(FD) logic is dedicated for the XTT2 rule language. It does not consider if the provided formalism can be used for rule interchange purposes.

Apart from the nowadays approaches to knowledge base design, there were several attempts done in the past. Historically, an important approach that aimed at full formalization of the knowledge representation including basic inference tasks was KADS. In fact, in this area an important effort was done, see [5, 24]. However, KADS had a very broad perspective on the knowledge representation, with rules being one of several methods. Due to this com-plexity, KADS-oriented research did not result in any practical tools for Rule-Based Systems.

2. State of the Art in Rule Interchange

Currently, there are several solutions that are being proposed for knowledge inter-change: Rule Interchange Format (RIF), Rule Markup Language (RuleML), REVERSE Rule Markup Language (R2ML), and Knowledge Interchange Format (KIF).

---

[1] See: http://pellet.owldl.com

RIF [14] is the extensible rule interchange format for the SemanticWeb. It is indented to be an extensible interchange format for all rule languages. The architecture of RIF consists of several dialects, which are the XML-based rule languages with well-defined semantics. Some of the dialects, especially RIF Core, are at early stage of development and can be superseded.

The practical application of RIF is limited because of their complexity and generic nature. What is more, the RIF specification leaves many questions open e.g. how to implement a transformation from a source rule language into RIF. Currently, RIF support is developed in a number of tools e.g. RIFLE[2], SILK[3], fuxi[4], Eye[5] and many others. Nevertheless, none of these tools provide complete support for RIF.

RuleML [4] is an XML-based language for the representation of rules. It provides a way for expressing Business Rules in modular stand-alone units and allows the deployment, execution, and exchange of rules between different systems and tools. Each unit provides support and semantics for a specific rule language and application. Thanks to the modular units, RuleML seems to be very flexible and extensible.

However, RuleML does not provide any mechanisms for semantics evaluation of the rule language elements. This is why, the rule interchange can be inconsistent at the semantic level. Furthermore, the provided units for specific rule languages lead to emergence of dialects. Thus, RuleML and RIF suffer from the same problem of complex dialects. The implementation of tool that supports all the dialects becomes very hard. Hence, the practical support for this language is very weak. Moreover, the issue of interchange between rule languages with complex data types and vocabularies (Drools) is not well clarified in RuleML.

R2ML [25] is a visual rule markup, which purpose is to capture rule formalized in different languages and interchange them between rule formats and tools. It also allows for enriching ontologies by rules. It has a XML based concrete syntax validated by an XML Schema allowing for different semantics for rules. Rule concepts are defined with the help of MOF/UML, a subset of UML class modeling language. Later, the MOF/UML representation is mapped to the concrete markup syntax.

In general, R2ML does not provide any specific semantics. This is why the semantics that accommodate the semantics of the target rule languages have to defined every time. What is more, R2ML do not assure that defined interchange method for target rule languages is lossless.

KIF[6] is a computer-oriented language for the interchange of knowledge among different applications. It is logically comprehensive (i.e. it provides the expression of arbitrary sentences

---

2   See: http://sourceforge.net/apps/mediawiki/rifle/index.php?title=Main_Page
3   See: http://silk.semwebcentral.org
4   See: http://code.google.com/p/fuxi
5   See: http://eulersharp.sourceforge.net/README\#eye
6   See: http://www.upv.es/sma/teoria/sma/kqml_kif/kif.pdf

in the first-order predicate calculus) and provides declarative semantics. It is possible to understand the meaning of the expressions without usage of interpreter. In this way, KIF differs from other languages that are based on specific interpreters such as Emycin and Prolog. The KIF language was intended as highly specialized not efficient knowledge representation language. So, it should not be used as an internal knowledge representation for applications.

KIF and their successors (SUO-KIF, LKIF) suffer from the same reasons as SBVR. The specification of KIF is very complicated and vague. Similarly to SBVR, KIF provides very complex meta-model consisting of large number of classes. Hence, from the practical point of view, complexity of the meta-model causes that there is a lack of tools supporting KIF.

3. Summary and Conclusions

This position paper concerns knolwedge interoperability in rule bases. It points that the existing interchange methods are mostly abstract and general. They provide complex metamodels and vague formalisms, which cause the practical application hard. This is why they suffer from the lack of the supportive tools as well as lack of the formalism that can be used in practical applications. In fact, some rule interchange languages provide logical foundation, however in order to assure the semantically coherent rule interoperability, they can be reasonably applied only to rule languages that provide formalized semantics. In general, the following problems related to knowledge interoperability can be noticed:

1. Lack of the logical interpretation of rule languages.
2. Lack of formalized way for knowledge exchanging between rule formats.

These problems are significant, because they decreases the maintenence and management capabilities. The efficient method for rule interoperability may reduce the number of knowledge repositories what in turn may reduce the risk of logical anomalies like redundancy or inconsistencies.

The methods for rule interchange are desirable and there are several approaches to solve this problem. In general, all of them suffer from their general nature, which causes that the practical support is very weak. The goal of our future work is to develop a formal method for knowledge interchange which would be supported by tools. We also assume that our method would provide a logic-based, unified knowledge representation model that can be used for providing a logical interpretation of the rule languages and for clarification of the rules semantics. Thanks to that, the proposed rule interchange method will allow for semantically coherent knowledge interchange supported by tools.

The results of our preliminary research show that the rule interchange between SKE and Drools rule languages is possible and semantically coherent. The translation was performed using no formal algorithm, but it was based on the identification of the knowledge

base elements that have a similar meaning. Moreover, it has been performed only in one direction: from SKE to Drools rule language. The details of our preliminary research are described in [13].

References

[1] Adrian W.T., Nalepa G.J., Kaczor K., Noga M., *Overview of selected approaches to rule representation on the Semantic Web.* Technical Report CSLTR 2/2010, AGH University of Science and Technology 2010.

[2] Antoniou G., Harmelen F., *A Semantic Web Primer*. The MIT Press 2008.

[3] Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. (Eds), *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press 2003.

[4] Boley H., Tabet S., Wagner G., *Design rationale for ruleml: A markup language for semantic web rules.* [in:] Isabel F. Cruz, Stefan Decker, Jérôme Euzenat, and Deborah L. McGuinness (Eds), *SWWS*, 2001, pp. 381–401.

[5] Fensel D., Angele J., Landes D., *KARL: A knowledge acquisition and representation language.* [in:] J.C. Rault (Ed.), Proceedings of the 11th International Conference Expert Systems and their Applications, vol. 1 (Tools, Techniques & Methods), pp. 513–528, Avignon, 1991. EC2, Nanterre.

[6] Friedman-Hill E., *Jess in Action, Rule Based Systems in Java*. Manning 2003.

[7] Giurca A., Gašević D., Taveter K. (Eds), *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*. Information Science Reference, Hershey, New York, May 2009.

[8] Hitzler P., Krötzsch M., Parsia B., Patel-Schneider P.F., Sebastian Rudolph, *OWL 2 Web Ontology Language – primer.* W3C recommendation, W3C, October 2009.

[9] Hopgood A.A., *Intelligent Systems for Engineers and Scientists*. CRC Press, Boca Raton London New York Washington, D.C., 2nd ed., 2001.

[10] Horrocks I., Patel-Schneider P.F., Boley H., Tabet S., Grosof B., Dean M., *SWRL: A semantic web rule language combining OWL and RuleML, W3C member submission 21 may 2004.* Technical report, W3C, 2004.

[11] Kaczor K., Nalepa G.J., *Extensible design and verification enviroment for XTT rule bases.* [in:] Ryszard Tadeusiewicz, Antoni Ligęza, Wojciech Mitkowski, and Maciej Szymkat (Eds), CMS'09: Computer Methods and Systems: 7th conference, 26–27 November 2009, Kraków, Poland, pp. 99–104, Cracow, 2009. AGH University of Science and Technology, Cracow, Oprogramowanie Naukowo-Techniczne.

[12] Kaczor K., Nalepa G.J., *HaDEs – presentation of the HeKatE design environment.* [in:] Joachim Baumeister and Grzegorz J. Nalepa (Eds), 5th Workshop on Knowledge Engineering and Software Engineering (KESE2009) at the 32nd German conference on Artificial Intelligence: September 15, 2009, Paderborn, Germany, pp. 57–62, Paderborn, Germany, 2009.

[13] Kaczor K., Nalepa G.J., Lysik L., Kluza K., *Visual design of Drools rule bases using the XTT2 method.* [in:] Radoslaw Katarzyniak, Tzu-Fu Chiu, Chao-Fu Hong, and Ngoc Nguyen (Eds), Semantic Methods for Knowledge Management and Communication, vol. 381 of Studies in Computational Intelligence, pp. 57–66. Springer-Verlag, 2011. DOI: 10.1007/978-3-642-23418-7.

[14] Kifer M., Boley H., *RIF overview.* W3C working draft, W3C, October 2009. http://www.w3.org/TR/rif-overview.

[15] Liebowitz J. (Ed.), *The Handbook of Applied Expert Systems*. CRC Press, Boca Raton, 1998.

[16] Ligęza A., *Expert systems approach to decision support.* European Journal of Operational Research, 37(1), 1988, pp. 100–110.

[17] Ligęza A., *Logical Foundations for Rule-Based Systems*. Springer-Verlag, Berlin, Heidelberg, 2006.

[18] Nalepa G.J., *Semantic Knowledge Engineering. A Rule-Based Approach*. Wydawnictwa AGH, Kraków 2011.

[19] Nalepa G.J., Ligęza A., *Special section: Semantic knowledge engineering.* International Journal of Applied Mathematics and Computer Science, 21(1), 2011.

[20] Nalepa G.J., Ligęza A., Kaczor K., *Formalization and modeling of rules using the XTT2 method.* International Journal on Artificial Intelligence Tools, 20(6), 2011, pp. 1107–1125.

[21] OMG. *Semantics of Business Vocabulary and Business Rules (SBVR).* Technical Report dtc/06-03-02, Object Management Group, 2006.

[22] Ross R.G., *The RuleSpeak Business Rule Notation.* Business Rules Journal, 7(4), April 2006.

[23] Russell S., Norvig P., *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 3rd ed., 2009.

[24] Harmelen F., Balder J., *(ML)2: A formal language for KADS models.* [in:] ECAI, 1992, pp. 582–586.

[25] Wagner G., Giurca A., Lukichev S., *R2ml: A general approach for marking up rules.* [in:] F. Bry, F. Fages, M. Marchiori, and H. Ohlbach (Eds), Principles and Practices of Semantic Web Reasoning, Dagstuhl Seminar Proceedings 05371, 2005.

[26] Wagner G., *How to design a general rule markup language.* [in:] XML Technology for the SemanticWeb (XSW2002), Lecture Notes in Informatics, HU Berlin, 2002, pp. 19–37.