Rule Interchange Format: How will change the Web

George Alexiadis
giorgito_alex@yahoo.gr

Abstract. One of the important issues concerning the Web is the ability of rule exchange between Web applications and systems. This paper is a review of Rule Interchange Format, introduced by the World Wide Web Consortium RIF Working Group. Its main purpose is to shortly describe RIF and research the impact it has on the Web. RIF's uses cases show that RIF affects the Web in such a way, that this is only the beginning.

Keywords: Rule Interchange Format, RIF framework, RIF dialect, rule exchange

2010 AMS Subject Classification Code: 68N02

1 Introduction

The Web is changing in a dramatically rapid manner. One of its important issues regards rule exchange between systems and applications. For this reason, the World Wide Web Consortium introduced a Web standard for exchanging rules, the Rule Interchange Format (RIF).

The first section of the paper presents the idea of RIF, created by the W3C RIF Working Group. The RIF WG designed a family of languages, called dialects, which are uniform and extensible, with a view to become standards. The second section describes the structure of RIF as a set of documents that serve different purposes. The last section focuses on some use cases of RIF and presents a more extensive example of RIF's usefulness and value, in order to show how RIF changes the Web.

2 The Idea of Rule Interchange Format (RIF)

Rules play a significant role in a variety of Semantic Web applications. The Web, as a universal medium for publishing information, is envisioned to become the place for publishing, distributing and exchanging rule-based knowledge [1]. W3C realized the importance and the promise of this vision, and created the Rule Interchange Format Working Group (RIF WG) as an attempt to develop an interchange format for rules in alignment with the existing standards in the Semantic Web layered architecture (Figure 1). Figure 1 shows the latest version, proposed by Tim Berners-Lee, of Semantic Web structure [2].

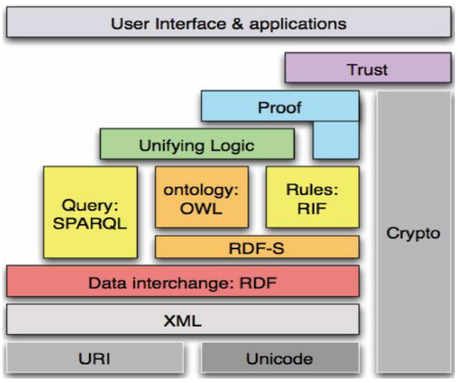

Figure 1: Semantic Web layered architecture [2]

The first attempts for rules on the Web came through XML [33]. XML provided a set of rules and syntax for structured documents, but it didn't support semantic constraints on documents' meaning. The next step was to create a Web standard that it could be for semantics, what XML was for syntax. This was accomplished by the introduction of RDF [3]. This was not enough though. There was a need to find a way to develop domain specific vocabularies. For this reason, OWL [4] was introduced as a language for creating ontologies describing and representing areas of knowledge. Moreover, SPARQL [5] provided a ground for querying in RDF.

Contrary to the above Semantic Web standards, the main deal was not to develop a single one-fits-all rule language, but to focus on the exchange of rules between different units. This is the goal of RIF, not at all simple, if we take into account the fact that the existing rule systems have a wide variety of features and differ a lot in syntax and moreover in semantics. Therefore, the question is how can interoperability be achieved?

The RIF WG designed a family of languages, called dialects, each of which having strict and specific syntax and semantics. These dialects should have two main characteristics. A RIF dialect must be uniform, in order to share as much as possible of the existing syntactic and semantic schemes. It also needs to be extensible, meaning that anyone expert should be able to define new RIF dialects as syntactic extensions to existing RIF dialects. The purpose of extending RIF dialects is that the new RIF dialects might eventually become Web standards [6].

3 The Structure of RIF

The structure of RIF is show in Figure 2. RIF consists of a set of documents, each of which serving a different purpose and referring to a different audience. Some of these documents may have common lines, while others may extend some previously created. The Venn diagram of Figure 2 depicts this idea [7].

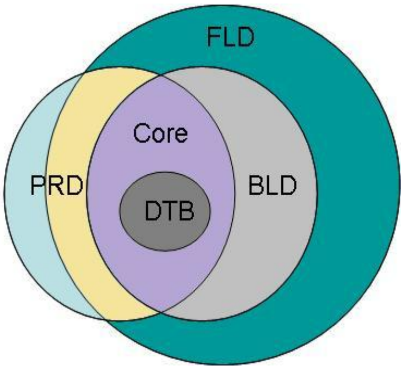

Figure 2: The structure of RIF [7]

3.1 RIF Dialects

RIF's name imposes the concept of a format, but RIF actually is something more than just a format. It enables the exchange of rules between different rule systems, using XML as its base. The idea is that different systems could provide syntactic mappings from their native languages to RIF dialects and opposite. Moreover, these mappings need to be semantics-preserving, so that the dialect of communication can be supported by both sides [8].

Boley and Kifer [9] report the aforesaid procedure of mapping: "*A RIF dialect is a rule-based language with an XML syntax and a well-defined semantics. A dialect D_1 can extend a dialect D_2, if the syntax of D_1 is a superset of the syntax of D_2, and the dialects are semantically compatible. A rule system A, a family of RIF dialects, can interchange its native ruleset R with a rule system B, if there is a RIF dialect D, such that A can map R to a ruleset in D, R^D, in a semantics-preserving manner, and B can map R^D to its native ruleset S, preserving the semantics too. The key point of this interchange is that both the syntax and the semantics of a RIF dialect, such as D, will be standardized, and the interchanging rule systems, such as A and B, must implement one or more dialects (i.e. D), in order to be RIF-compliant. A rule system implements a dialect, if the native language of the system is a syntactic variant of the language of the dialect with possible extensions. Therefore, if A and B implement D, then there are semantics-preserving mappings from D onto some subsets of the languages of A and B, and vice versa. If the interchanging rules fall into these subsets, the aforesaid mappings enable the interchange of the rules between the two systems.*" This procedure is shown in Figure 3.

The RIF WG takes into account two kinds of dialects, logic-based dialects and dialects for rules with actions. The first one includes languages that employ some kind of logic, such as first-order logic or non-first-order logics underlying the various logic programming languages (i.e. logic programming under the well-founded [10] or stable semantics [11]). The second one includes production systems, such as Jess [12], Drools [13] and JRules [14], and furthermore reactive rules, such as Reaction RuleML [15] and XChange [16].

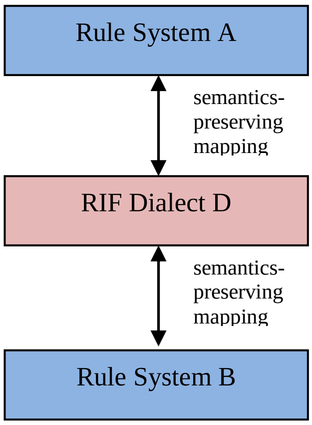

Figure 3: The process of rule exchange between two rule systems

Until now, the RIF WG has defined two logic dialects, the Basic Logic Dialect (RIF-BLD) and the RIF Core Dialect, and one rule-with-actions dialect, the Production Rule Dialect (RIF-PRD). Figure 4 shows the current state of RIF dialects.

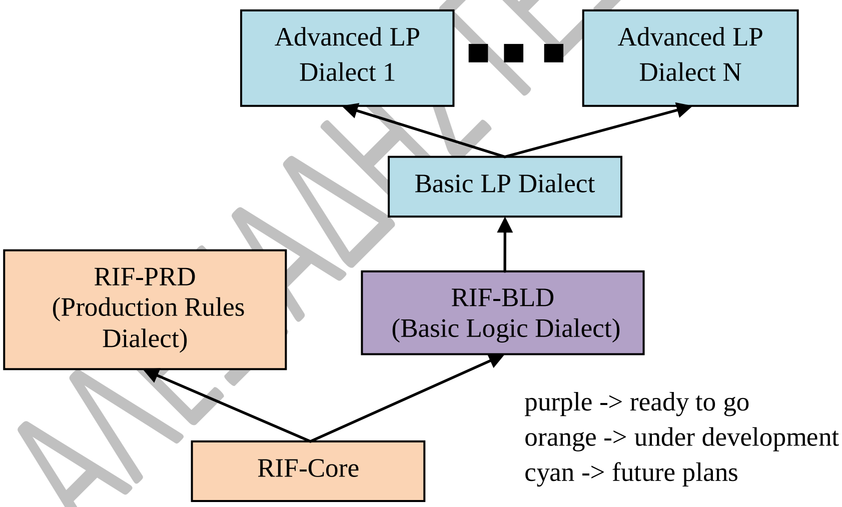

Figure 4: The current state of RIF dialects

3.2 RIF Framework for Logic Dialects (RIF-FLD)

Defining a dialect from scratch is a hard and time-consuming procedure. For this reason, the RIF WG developed an extensibility framework, called the Framework for Logic Dialects (RIF-FLD). This work proved to be feasible due to the fact that the logical theories beyond different logic rule systems share much of the same syntactic and semantic machinery, and moreover the ways to combine the different pieces of that machinery in order to create those logic systems are well studied [17].

RIF-FLD consists of three main components [9]:

- Syntactic framework. It defines the mechanisms for specifying the formal presentation syntax of RIF's logic dialects. The presentation syntax is not just a concrete syntax, but it defines the semantics of dialects and illustrates the main issues by providing examples. The only concrete syntax used by RIF dialects is through XML.
- Semantic framework. It describes the mechanisms used for specifying the models of RIF logic-based dialects.
- XML serialization framework. It defines the general principles that RIF logic-based dialects should use in order to specify their concrete XML-based syntaxes.

RIF-FLD is very general and captures most of the popular logic rule languages found in Databases, Logic Programming, and on the Semantic Web. However, the need for future dialects will require the further evolution of RIF-FLD. Future extensions might include a logic rendering of actions as found in production and reactive rule languages. This would support Semantic Web services languages such as SWSL-Rules [18] and WSML-Rules [19].

3.3 RIF Basic Logic Dialect (RIF-BLD)

RIF-BLD is the only fully specified dialect of RIF until now. It is a dialect corresponding to the language of definite Horn rules with equality and a standard first-order semantics [6]. RIF-BLD has a number of extensions that support features such as objects and frames as in F-logic [20], internationalized resource identifiers (or IRIs, defined by [21]) as identifiers for concepts, and XML Schema datatypes [22]. Considering the significance of such features, RIF-BLD may eventually become a Web-aware language.

RIF-BLD is defined in two normative ways [23]:

- as a direct specification, independently of RIF-FLD, so that, anyone who desires a direct path to RIF-BLD (i.e. prospective implementers) and is not interested in extensibility issues, can benefit from it, and
- as a specialization of RIF-FLD, so that, anyone who is familiar with RIF-FLD and does not need to follow the long path of the direct specification of RIF-BLD, can benefit from this specialization.

3.4 RIF Production Rules Dialect (RIF-PRD)

The RIF WG developed RIF-PRD with a view to capture the main aspects of various production rule systems [24]. There is a serious industrial, and not only, interest in production rule technology, therefore RIF could help to this direction. RIF-PRD is not a part of RIF logical dialects and can stand alone, apart from them. This is explained if we consider production rules practiced in main-stream systems like Jess [12] or

JRules [14], in which these rules are defined using ad hoc computational mechanisms, not based on a logic. Nevertheless, the RIF WG tried to make RIF-PRD capable of sharing much with the other dialects, by developing the RIF Core Dialect.

3.5 RIF Core Dialect (RIF-Core)

RIF-Core is a dialect with two main characteristics: it corresponds to the language of definite Horn rules without function symbols and with a standard first-order semantics, and it is also a language of production rules where conclusions are interpreted as assert actions. The first characteristic implies that RIF-Core is actually a subset of RIF-BLD, while the second one implies that it is also a subset of RIF-PRD. The main role of RIF-Core is to enable limited rule exchange between logic rule dialects and production rules [25].

3.6 RIF Datatypes and Built-ins (RIF-DTB)

RIF-DTB is a document enabling semantics-preserving exchange of rules that contain datatypes (i.e. strings, integers) and built-ins (i.e. arithmetics, string manipulation). It is currently on version 1.0 and helps in the interoperation of RIF with other semantic Web formalisms by providing a general infrastructure of datatypes and built-ins [26].

3.7 Other RIF Documents

The RIF WG has created some other than then aforesaid documents, as an effort to capture the purpose of RIF. These documents are the following:

- RIF RDF and OWL Compatibility (RIF-RDF+OWL): This document describes how can interoperability between RIF and RDF or OWL be achieved [27].
- RIF Combination with XML Data (RIF+XML-Data): This document specifies how RIF can be combined with XML data sources [28].
- OWL 2 RL in RIF (RIF-OWLRL): This document shows that OWL 2 RL, an OWL 2 subset, can be used as the basis for a rule-based implementation using RIF [29].
- RIF Use Cases and Requirements (RIF-UCR): This document presents the classes of applications that the RIF suite of dialects is able to address, according to RIF WG's research. This helped to derive requirements to RIF, which consequently contributed to the design of RIF dialects [7].
- RIF Test Cases (RIF-Test): This document includes the description of test cases (positive and negative) that can be used in order to give an indication of whether a particular implementation of a RIF dialect is compliant with the specifications. It is mainly targeted for RIF implementers [30].

Given the overviewed structure of RIF, described in this section, it is obvious that RIF changes the Web. The question is how and to which degree this change is feasible. A part of the answer to this question is based on the aforesaid overview. But, in order to realize the affect RIF has on the Web, there is a need for examples. These examples are given by the following section, which presents some of RIF's uses cases on the Web together with a more extensive example showing the value RIF.

4 RIF and the Web

Until now, the RIF WG has submitted nearly fifty use cases documenting the need of a RIF by providing scenarios motivating the current design of RIF and explaining its benefits. Some of these uses cases are the following one:

- Negotiating eBusiness Contracts Across Rule Platforms: This use case supplies a vendor-neutral representation of rules. It implies that rule-system developers and stakeholders can work and make product investments without being concerned about vendor technology. RIF's use in this case can bring and improve collaborative work [7].
- Negotiating eCommerce Transactions Through Disclosure of Buyer and Seller Policies and Preferences: This use case concerns a policy-governed framework established for members involved in formal transactions or procedures (i.e. credit card authorization of a purchase, access of private medical records). It enables them to express their interests and priorities, and protect their preferences and responses within this framework [7].
- Rule-based Email Manipulation: In this use case a user of an email system can define his/her own rules about processing incoming and outgoing messages automatically. When the user switches to another email system, RIF can help to interchange the rules between these systems [31].
- Organizing a Vacation with Friends: In this use case some friends use a Web-based travel service in order to get recommendations and arrange their vacation. A critical issue is how can, in such a case, rules be interchanged between different systems (the travel service and other more specialized services – i.e. hotel booking services) and help combine the different profiles in order to find a trip fitting most of friends' expectations. This is achieved by RIF [32].

These uses cases of RIF give a view of how important the RIF WG's idea of creating a rule interchange format is. To better understand RIF's impact with regard to the Web, let us consider the following example.

The current technological and regulatory trends converge to a flexible architecture, under which reconfigurable devices may operate legally in various regulatory and service environments [34]. Suppose there is a policy stating that a wireless device can transmit on a 5 GHz band, only if no priority user is currently using this band. Suppose also two devices that use different rules for the use of band. The rule of the first device states "if no energy is detected on a desired band, then assume no other device is using the band", while the rule of the second device states "if no control signal indicating use of a desired band by a priority user is detected, then assume the band is available". It is obvious that both devices will need to employ different interpretations or operational definitions of the questioned policy.

Consider now ten manufacturers of these two different types of wireless devices, each of which using a distinct rule-based platform. For each of these two different types, each manufacturer requires two interpretations. As a consequence, twenty different versions of the policy must be written, tested and maintained. This procedure can be automated adopting RIF for the interchange of rules and automating the translation process. The use of RIF in this example is ideal.

Conclusion

This paper has presented the idea of Rule Interchange Format, its structure and some of RIF's uses cases concerning the Web. Rules play an important role in many applications regarding or not the Web. RIF came as an effort to create a framework for rule exchange with a view of this framework becoming general and extensible. RIF-FLD is still work in progress: some details may change and additions to the framework should be expected. A dialect for production rule systems is under development. Moreover, other dialects are being planned, such as logic programming dialects, dialects that support higher-order extensions and dialects extending RIF-BLD with full F-logic support. The development of the RIF standard is an open process and feedback from experts and users is welcome.

What is done so far is significant and it is just the beginning. The use cases of RIF, some of which presented in this paper, are examples of RIF's impact on the Web. Figure 1 depicts the position of RIF in the latest version of Semantic Web architecture. What remains in this area is to examine the role of rules and RIF within the upper layers of this architecture.

References

[1] Boley, H., Kifer, M., Patranjan, P.-L., Polleres, A. 2007. *Rule Interchange on the Web*.

[2] Berners-Lee, T., Hall, W., Hendler, J.A., O'Hara, K., Shadbolt, N., Weitzner, D.J. 2006. *A framework for Web Science*.

[3] Klyne, G., Carrol, J.J., W3C Recommendation 2004. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. <URL: http://www.w3.org/TR/rdf-concepts/>. Accessed 05 February 2010.

[4] Dean, M., Schreiber, G., W3C Recommendation 2004. *OWL Web Ontology Language Reference*. <URL: http://www.w3.org/TR/owl-ref/>. Accessed 05 February 2010.

[5] Prud'hommeaux, E., Seaborne, A., W3C Recommendation 2008. *SPARQL Query Language for RDF*. <URL: http://www.w3.org/TR/rdf-sparql-query/>. Accessed 05 February 2010.

[6] Kifer, M. 2008. Rule Interchange Format: The Framework.

[7] Paschke, A., Hirtle, D., Ginsberg, A., Patranjan, P.-L., McCabe, F., W3C RIF WG Wiki 2008. *RIF Use Cases and Requirements*. <URL: http://www.w3.org/2005/rules/wiki/UCR>. Accessed 05 February 2010.

[8] Boley, H., Kifer, M., W3C RIF WG Wiki 2009. *RIF Overview*. <URL: http://www.w3.org/2005/rules/wiki/Guide>. Accessed 05 February 2010.

[9] Boley, H., Kifer, M. 2007. The Rule Interchange Format: An Interim Report.

[10] Van Gelder, A., Ross, K.A., Schlipf, J.S. 1991. *The Well-Founded Semantics for General Logic Programs*. Journal of ACM 38:620-650.

[11] Gelfond, M., Lifschitz, V. 1988. *The Stable Model Semantics for Logic Programming*. Pages 1070-1080. Proceedings of the Fifth Conference and Symposium.

[12] *Jess, the Rule Engine for the Java Platform* 2008. <URL: http://www.jessrules.com/>. Accessed 05 February 2010:.

[13] *Drools, Business Logic integration Platform*. <URL: http://jboss.org/drools/>. Accessed 05 February 2010.

[14] *WebSphere ILOG JRules*. <URL: http://www.ilog.com/products/jrules>. Accessed 05 February 2010.

[15] *Reaction RuleML* 2007. <URL: http://reaction.ruleml.org>. Accessed 05 February 2010.

[16] *XChange: A rule-based language for programming reactive behavior on the Web*. <URL: http://reactiveweb.org/xchange/>. Accessed 05 February 2010.

[17] Boley, H., Kifer, M., W3C RIF WG Wiki 2009. *RIF Framework for Logic Dialects*. <URL: http://www.w3.org/2005/rules/wiki/FLD>. Accessed 06 February 2010.

[18] Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D., Su, J., Tabet, S., W3C Member Submission 2005. *Semantic Web Services Language (SWSL)*. <URL: http://www.w3.org/Submission/SWSF-SWSL/>. Accessed 06 February 2010.

[19] De Bruijn, J., Fensel, D., Keller, U., Kifer, M., Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., W3C Member Submission 2005. *Web Service Modeling Language (WSML)*. <URL: http://www.w3.org/Submission/WSML/>. Accessed 06 February 2010.

[20] Kifer, M., Lausen, G., Wu, J. 1995. *Logical foundations of object-oriented and frame-based languages*. Journal of ACM:741-843.

[21] Duerst, M., Suignard, M. 2005. *Internationalized Resource Identifiers (IRIs)*. <URL: http://www.ietf.org/rfc/rfc3987.txt>. Accessed 06 February 2010.

[22] Peterson, D., Gao, S., Malhotra, A., Sperberg-McQueen, S.M., Thompson, H.S., W3C Working Draft 2009. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. <URL: http://www.w3.org/TR/xmlschema11-2/>. Accessed 06 February 2010.

[23] Boley, H., Kifer, M., W3C RIF WG Wiki 2009. *RIF Basic Logic Dialect*. <URL: http://www.w3.org/2005/rules/wiki/BLD>. Accessed 06 February 2010.

[24] De Sainte Marie, C., Paschke, A., Hallmark, G., W3C RIF WG Wiki 2009. *RIF Production Rule Dialect*. <URL: http://www.w3.org/2005/rules/wiki/PRD>. Accessed 06 February 2010.

[25] Boley, H., Hallmark, A., Kifer, M., Paschke, A., Polleres, A., Reynolds, D., W3C RIF WG Wiki 2009. *RIF Core Dialect*. <URL: http://www.w3.org/2005/rules/wiki/Core>. Accessed 06 February 2010.

[26] Polleres, A., Boley, H., Kifer, M., W3C RIF WG Wiki 2009. *RIF Datatypes and Built-Ins 1.0*. <URL: http://www.w3.org/2005/rules/wiki/DTB>. Accessed 07 February 2010.

[27] De Bruijn, J., W3C RIF WG Wiki 2009. *RIF RDF and OWL Compatibility*. <URL: http://www.w3.org/2005/rules/wiki/SWC>. Accessed 07 February 2010.

[28] De Sainte Marie, C., W3C RIF WG Wiki 2009. *RIF Combination with XML data*. <URL: http://www.w3.org/2005/rules/wiki/RIF%2BXML_data-schema>. Accessed 07 February 2010.

[29] Reynolds, D., W3C RIF WG Wiki 2009. *OWL 2 RL in RIF*. <URL: http://www.w3.org/2005/rules/wiki/OWLRL>. Accessed 07 February 2010.

[30] Mitchell, S., Morgenstern, L., Paschke, A., W3C RIF WG Wiki 2008. *RIF Test Cases*. <URL: http://www.w3.org/2005/rules/wiki/Test>. Accessed 07 February 2010.

[31] Patranjan, P.-L., W3C RIF WG Wiki 2006. *Rule-Based Email Manipulation*. <URL: http://www.w3.org/2005/rules/wg/wiki/Rule-Based_Email_Manipulation>. Accessed 07 February 2010.

[32] Bry, F., W3C RIF WG Wiki 2006. *Organizing a Vacation with Friends*. <URL: http://www.w3.org/2005/rules/wg/wiki/Organizing_a_Vacation_with_Friends>. Accessed 07 February 2010.

[33] Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F., W3C Recommendation 2008. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. <URL: http://www.w3.org/TR/REC-xml/>. Accessed 05 February 2010.

[34] Fensel, D., Facca, F., Semantic Web Course WS2009/10. STI Universitat of Innsbruck. <URL: http://www.sti-innsbruck.at/teaching/courses/ws200910/details/?title=semantic-web>. Accessed 26 February 2010.