RIF Use Cases and Requirements (Second Edition)

1. Introduction

Rule languages and rule-based systems have played seminal roles in the history of computer science and the evolution of information technology. From expert systems to deductive databases, the theory and practice of automating inference based on symbolic representations has had a long history and continues to be a key technology driver.

Due to the interactions made possible by the Internet, the World Wide Web, and, most recently, the Semantic Web, there is now even greater opportunity for growth in this sector. While some of these opportunities may require advances in research, others can be addressed by evolving existing rule-based technologies to interpret according to standards-based methodologies and processes. The primary goal of the Rule Interchange Format (RIF) Working Group has been to devise such standards and make sure that they are not

The purpose of this RIF-UCR document is to describe the use cases that guided the design of RIF and to document the requirements that were derived from both the use cases and from the basic goals of RIF. RIF-UCR also delivers a structured context for formulating future technical specifications of further RIF dialects. Each dialect targets a cluster of similar rule technologies and allows rules to be translated between rule languages and thus transferred between rule systems.

The RIF-UCR document is also available in these non-normative formats: PDF version.

Copyright © 2013 W3C (this document). All Rights Reserved. W3C Technical Reports index at
RIF can provide efficient process management through reuse, sharing, and the ability to provide unified views across disparate platforms. Finally, RIF can promote the growth of knowledge by enabling reasoning with merged sets of rules originating from disparate knowledge sources.

The RIF-UCR document is structured as follows: Section 2 formulates the overall goals of RIF. Section 3 summarizes the released RIF dialects and the structure of RIF. Section 4 discusses several important requirements for RIF dialects. Section 5 presents a set of use cases that illustrate the utility of the current RIF dialects. Finally, the functionality specified in the use cases, together with the requirements discussed in the previous section, serves as input for both the technical specification for future RIF dialects and the implementation of various variants of these semantics by RIF-compliant applications and systems.

4.1 Implementability
RIF must be implementable using well-understood techniques, and should not require new research in, e.g., algorithms or semantics in order to implement translators.

4.2 Semantic precision
RIF-Core must have a clear and precise syntax and semantics. Each standard RIF dialect must have a clear and precise syntax and semantics that extend RIF-Core.

4.3 Extensible format
It must be possible to create new RIF dialects that extend existing dialects (thus providing backward compatibility) and are handled gracefully by systems which support existing dialects (thus providing forward compatibility).

4.4 Translators
For every standard RIF dialect it must be possible to implement translators between RIF languages covered by that dialect and RIF without changing the RIF language.

4.5 Standard components
RIF implementations must be able to use standard support technologies such as XML parsers and other parser generators, and should not require special purpose implementations when reuse is possible.

4.6 Rule language coverage
Because of the great diversity of rule languages, rule interchange language is likely to be able to bridge between all. Instead, RIF provides dialects each of which targets a cluster of similar rule languages. RIF must allow intra-dialect interoperability, i.e., interoperability between semantically similar rule languages (via interchange of RIF rules) within a dialect, and should support inter-dialect interoperability, i.e., interoperability between dialects with maximum overlap.

4.7 Compliance model
The RIF specifications must provide clear conformance criteria that define what is and what is not a conformant RIF implementation.

4.8 Default behavior
RIF must specify the at the appropriate level of detail the default behavior that is expected from a RIF compliant application that does not have the capability to process all or part of the rules described in a RIF document, or it must provide a way to specify such default behavior.

4.9 Different semantics
RIF must cover rule languages having different semantics.

4.10 Limited number of dialects
RIF must have a standard core, a limited number of standard dialects based upon that core.

4.11 Embedded comments
RIF must be able to pass comments.

4.12 Embedded metadata
RIF must support metadata such as author and rule name.
How does a device know that no priority user is currently using a band it wants to use? The answer will depend on the environment. The ability of a device to absorb the rules defining the policies of a region, or the operational protocols for a spectrum, is contingent upon those rules being in a form that the device can use, as well as their being tailored to work with devices in the same class having different capabilities.

The set of use cases was developed over several years. Nearly all use cases documenting the need for a RIF were originally submitted by RIF working group members. These were grouped into general categories and then synthesized as much as possible. The goal was to come up with a set of use cases that would address application domains and industrial sectors, and that would cover a broad range of possible requirements. Following the presentation of each use case, we briefly discuss whether it can be covered by the existing RIF dialects (Core, BLD, and PRD). All use cases except for 5.2 and 5.6 can be covered at least in part by at least one of the dialects. The reader will note that several cases which require negation are covered only by PRD; negation is not expressible in either BLD or Core.

5.1 Negotiating E-business Contracts Across Rule Platforms

This use case illustrates a fundamental use of RIF to supply a vendor-neutral representation of rules. This enables rule-system developers and stakeholders to do their work for example, to make product investments without concern for vendor lock-in, and without concern that a business partner does not have the same vendor technology. It also illustrates the fact that RIF can be used to foster collaborative work. Each developer and stakeholder can make a contribution to the joint effort without being forced to adopt the tools or platforms of the other contributor.

John is negotiating an electronic business contract regarding the supply of various types of items that John's company is manufacturing. Jane and John interchange the contract-related data and rules involved in the negotiation in electronic form so that they can run simulations. Both agree on a standard Business Object Model [data model (i.e., vocabulary and ontology)] for the goods and services involved. In this case an XML schema and a appropriate text RIF documents are exchanged with their rules. Since Janes and Janes run applications based on different vendors' rule engines and rule languages, they interchange the rules using RIF. Both vendors can interpret the XML schema and data, and produce the results as an annotated XML document. John's company defines its purchase orders in terms of an XML description of goods, packaging, delivery location, and date of delivery and payment rules. A rule proposed by John might be the following:

If an order is perishable and is delivered to John more than 38 days after the scheduled delivery date, then he will reject the item.

Jane replies with suggested rule changes:

If an item is perishable and it is delivered to John more than 7 days after the scheduled delivery date but less than 14 days after the scheduled delivery date then a discount of 10.7% will be applied to the delivered item.

John considers this and agrees with Jane. Both organizations utilize these rules in their operational systems using disparate rule representations internally to compute prices for this order and determine contract compliance.

5.2 Negotiating eCommerce Transactions Through Disclosure of Buyer and Seller Policies and Preferences

This use case describes the ability of vendor to present formatted information or procedures, e.g. credit and background verification of signature or access of private medical records, to express and protect their interests within a policy governed framework. The goal is to broadly encode the preferences, priorities, and responses of the parties in such a way that the overall policy can be used while still providing opportunity for automatic negotiation of terms as it is allowed by the policy. Utilizing RIF in this use case would extend this capability to automate the negotiation process, as well as including use and sharing of preferences through interchange.

Alice wants to buy a device at an online site called "eShop." Alice employs software called "Emptor" that functions as automated negotiating agent for buyers. eShop employs software called "Venditor" as automated negotiating agent for sellers. Alice's and eShops policies describe whom they trust and for what purposes. The negotiation is based both on their policies, which are specified as rules, and on Emptor's and Venditor's credentials. These policies and credentials are disclosed (interchanged) in order to assure a successful transaction as well as to provide for automatic negotiation.

Policies are themselves subject to access control. Thus, rule interchange is only done during negotiation and in general depends on the current level of mutual trust. Since Emptor and Venditor might use different rule languages and/or engines for evaluating their own as well as imported rules, a standard rule interchange format needs to be employed for enabling the use interchange between the two systems.

Alice and John click on the "Buy" button at eShops' web site. Emptor takes over and sends a request to eShops site: vendor receives the request and sends parts of its policy (i.e. a set of rules) back to Emptor. Among other things, the policy states that:

In order to grant access a buyer must provide valid credit card information together with delivery information (address, postal code, city, and country).

Rules compactly express possible ways in which a resource can be accessed; by exchanging them negotiations are shorter and faster.

To determine whether Vendor's request for information is consistent with Alice's policy, Emptor takes its rules into account. One of these rules states:

Disclose Alice's credit card information only to those shops belonging to the Better Business Bureau.

By disclosing (interchanging) the above rule, Emptor asks Vendor to provide credentials saying that it belongs to the Better Business Bureau. Alice's most trusted source of information on online shops is Eshop. If Eshop has these credentials, however, before Emptor can disclose Alice's credit card information to Vendor, it must still check whether disclosing it is requested by Alice.

Alice's denial constraints. Alice has stated the following constraint:

For purposes of anonymity, never provide both the person's birth date and postal code.

Since this is a factual constraint, both birth date and postal code are necessary. Hence, Alice's constraints are violated. Alice's credit card information is not disclosed.

Companies that provide software such as Emptor and Venditor can make use of RIF in several ways. First, the rules expressing Alice's and/or eShops policies could be expressed in different rule languages but still work with a common software, using RIF-based interchange. Second, RIF would enable Emptor and Venditor to work together in real time, even if these automated agents are products of different companies, and different internal rule languages. When these two systems would need to exchange policy or preference information of their respective clients, they would use RIF to enable the interchange in real time. When vendor sends its initial policy information to Emptor, it would use RIF; likewise, Emptor would take that policy and translate it from RIF to its internal representation in order to determine what it needs to do.

Not currently supported by existing RIF dialects (Core, BLD, PRD)

Current RIF dialects do not specify explicit constructs for integrity constraints; thus, this use case is not adequately supported by any of these dialects.

5.3 Collaborative Policy Development for Dynamic Spectrum Access

Current RIF dialects do not specify explicit constructs for integrity constraints; thus, this use case is not adequately supported by any of these dialects.
A second type of device may give information from a central console that lets you know whether the desired band is being used by a priority user. That is, it might employ the rule:

If I control signal indicates use of a desired band by a priority user is detected then assume the band is available.

So every type of device will need to employ different "interpretations" or "operating semantics" of the policy in question.

Now assume that there are 10 manufacturers of these 2 different types of wireless devices. Suppose that each of these manufacturers uses a distinct rule-based platform in designing its devices. Each manufacturer needs to write 2 interpretations of the policy (one of the two types of device). That is, 20 different versions of the policy must be written, tested, and maintained.

This is a very significant cost. This is a strong group that is trying to standardize the radio policy. If it succeeds, however, it will have a cost savings because it will have reduced the number of different interpretations from 20 to 1. A cost model is often used to estimate the cost savings. If the policy is adopted by the industry, the cost savings is estimated to be $1 billion over the lifetime of the product.

Note that there are 10 manufacturers of these 2 different types of wireless devices. They all have to design their own interpretation of the policy. This is a very significant cost. If it succeeds, however, it will have reduced the number of different interpretations from 20 to 1. A cost model is often used to estimate the cost savings. If the policy is adopted by the industry, the cost savings is estimated to be $1 billion over the lifetime of the product.

Partly supported by: RIF

The energy-detection function specified in this use case would require an external call to a device that would sense the level of energy. Such procedural attachments cannot be specified in the current RIF dialects. Reaction rules support (complex) event detection capabilities, e.g., event streams over sensor data.

This use case is not supported by all RIF-BLD, and by extension, by RIF-Core. These dialects do not support reaction, and use the case makes explicit mention of reaction, e.g., for representing a "to the extent that user priority is currently being used by the bands."
The credit evaluation rules are themselves grouped into three rule sets that are executed sequentially. Rules in the first rule set apply thresholds to several “red flag” quantities in the credit report, such as:

- Original credit score
- Number of inquiry dates
- Number of bankruptcies

Because the rule engine is part of a Web service, existing BPEL diagramming and execution facilities can be used to implement the rule engine. If Bob’s credit evaluation service is operating as a BPEL service and a decision service containing a rules engine. BPEL first passes the credit request document to the decision service containing the rules engine. The credit evaluation model can be changed easily using GUI tools to customize rules. Using RIF would enable rules to be changed in place directly in the Web service and a decision service containing a rules engine. This is potentially problematic, since GlucosePoison has been known to cause acute renal failure.

An existing commercial credit approval service deployed as a Web service takes an applicant credit request document as input. Its input is the credit request. The output is a decision (accept or denial (with reason). It is implemented as a BPEL orchestration of two Web services — a credit history providing service and a decision service containing a rules engine. The decision service contains a set of RIF rules that implement the credit evaluation model.

Supported by RIF-BLD, RID-PRD, and RIF-Core
This gives him a uniform view that links from business processes through to the IT and finance data. Vlad publishes these rules so that other people across the company can reuse them to construct similar views.

He then creates rules that combine the data across his now simplified data sources, e.g.

- The bank database models the world in terms of book-keeping data such as accounts and clients. It is too fine-grained.
- The patient databases, which gives the patient record, including the medications a patient is currently taking. This has been significantly simplified.
- The finance database models the world in terms of physical assets such as racks, which is too coarse-grained.

Each of these sources is in a different form but can be mapped into RDF to simplify access. However, they all have some common vocabulary that needs to be captured.

The use case requires the formalization of changeable states (fluents) which can be represented by e.g. an event calculus axiomatization [3]. However, to formalize the classical event calculus as a set of horn clause rules so that it could be run as a Prolog program, resolution is needed, which is not supported by RIF-BLD or Core. The use case can be represented in RIF-PDR which supports changeable states (modifications) and assertions and retractions of knowledge.

5.7 Interchanging Rule Extensions to OWL
Rules are often used in conjunction with other declarative knowledge representation formalisms, such as ontology languages (e.g. RDFS and OWL). In order to provide greater expressive power than is provided by other formalism alone, Ontology languages, for example, typically have a mechanism for describing classes (so-called classes) and constraints on classes, and may also support higher-predicate operators.

Rich domain models combining both rules and ontologies are often needed in domains such as medicine, biology, a Science, and Web services. In such domain, several actors and also agents need to interoperate — the heterogeneous medical professionals and agents in a hospital environment. The challenge is to develop a standard for describing such models in an application service-oriented framework in RDF. In this case, an OWL ontology is used to capture knowledge about the most important brain cortex anatomical structures, and rule base is used to capture knowledge about neurological and spatial dependences between properties.

For example, a rule is used to express the dependency between the ontology properties isMAEConnectedTo and isMAEAdjacentTo, in particular a simplified form of the knowledge that two Material Anatomical Entities having a shared boundary are connected:

$$\text{If } MAE_X \text{ is bounded by } Z \text{ and } MAE_Y \text{ is also bounded by } Z \text{ then } X \text{ is connected to } Y.$$  

Benefits of interoperate via RIF include the ability to collaboratively and share variable knowledge, the ability to integrate anatomical images, possibly from distributed image sources, and the ability to use large-scale federated systems for statistical analysis of brain images of major brain pathologies.

Supported by RIF-Core, RIF-BLD and RIF-PDR
The use case uses RIF-FG to combine RIF Rules with OWL ontologies.

5.8 Vocabulary Mapping for Data Integration
This use case concerns the integration of information from multiple data sources. The Semantic Web provides a common data representation and query language, which greatly simplifies access to diverse sources but does not directly address the problem of integrating information, which may have different domain models.

Rules are an effective way to express mappings between such information models. However, rules locked within local proprietary systems cannot be reused. With a common rule representation, mappings can be published across the Semantic Web, enabling an extension of the information space. A common rule representation would help develop a unified knowledge broker for RDF ontologies, which details the protocol used for different medical procedures.

In this case, RIF-Core uses all the following up the pushing follow-up information:

- Metformin is contraindicated with contrast dye.
- Metformin is in the generic form of Glucophage.
- Glucophage is contraindicated with contrast dye.
- The contrast MRI requires as one of its steps injecting the patient with contrast dye.

RIF-Core therefore determines that Bob should not be taking the contrast MRI at the time with Metformin for the secondary medication.

For RIF to work, the rules from these different sources must be mapped to a uniform interchange format.

Supported by RIF-PRD
The use case describes the formalization of changeable states (fluents) which can be represented by e.g. an event calculus axiomatization [3]. However, to formalize the classical event calculus as a set of horn clause rules so that it could be run as a Prolog program, resolution is needed, which is not supported by RIF-BLD or Core. The use case can be represented in RIF-PDR which supports changeable states (modifications) and assertions and retractions of knowledge.

Use Case 3: AutoDoc
The AutoDoc system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

- If applicant’s income is under 20000 then add 10.
- If applicant rents then add 30.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.

This system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying what these results mean, and a pharmaceutical knowledge base giving guidelines for the use of particular drugs. Automated reasoning with rules from these combined sources is possible if the rules are first mapped into RIF. Here are two specific examples of such synergistic effects.
This appendix summarizes the main changes to this document since its publication as a Working Draft.

Changes since the draft of December 18th, 2010:

- The section describing the structure of RIF has been updated with newer RIF documents;
- AllMajor examples have been removed from this document;
- A new Major example has been added for the use case by the current RIF dialects Core, BLD, and PRD. In addition, a brief discussion of the features of the use case that may affect a dialect’s support for that use case is sometimes included;
- All information on change log has been added;
- The distinction between general and basic requirements has been removed, thereby refining the list of requirements.

7 References


8 Appendix: Change Log (Informat...