W3C Working Group Note

RIF Use Cases and Requirements (Second Edition)

W3C Working Group Note 5 February 2013

This version:
http://www.w3.org/TR/2013/NOTE-rif-ucr-20130205/
Latest version:
http://www.w3.org/TR/rif-ucr/
Previous version:
http://www.w3.org/TR/2008/WD-rif-ucr-20081218/
Editors:
Adrian Paschke, Freie Universitaet Berlin
Leora Morgenstern, SAIC
David Hirtle, National Research Council Canada
Allen Ginsberg, Mitre
Paula-Lavinia Patranjan, REWERSE
Frank McCabe, Fujitsu

A color-coded version of this document showing changes made since the previous version is also available.

This document is also available in these non-normative formats: PDF version.

Abstract

This document, developed by the Rule Interchange Format (RIF) Working Group, specifies use cases and requirements for the W3C Rule Interchange Format, a family of rule interchange dialects that allows rules to be translated between rule languages and thus transferred between rule systems.

Status of this Document

May Be Superseded

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

Set of Documents

This document is being published as one of a set of 13 documents:

1. RIF Overview (Second Edition)
2. RIF Use Cases and Requirements (Second Edition) (this document)
3. RIF Core Dialect (Second Edition)
4. RIF Basic Logic Dialect (Second Edition)
5. RIF Production Rule Dialect (Second Edition)
6. RIF Framework for Logic Dialects (Second Edition)
7. RIF Datatypes and Built-Ins 1.0 (Second Edition)
8. RIF RDF and OWL Compatibility (Second Edition)
9. OWL 2 RL in RIF (Second Edition)
10. RIF Combination with XML data (Second Edition)
11. RIF In RDF (Second Edition)
12. RIF Test Cases (Second Edition)
13. RIF Primer (Second Edition)

Summary of Changes

The document was reorganized, sections were re-written, and an analysis was added of how the use cases are addressed by the current RIF specifications.

Please Send Comments

Please send any comments to public-rif-comments@w3.org (public archive). Although work on this document by the Rule Interchange Format (RIF) Working Group is complete, comments may be addressed in the errata or in future revisions. Open discussion among developers is welcome at public-rif-dev@w3.org (public archive).

No Endorsement

Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Patents

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

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 rich history and continues to be a key technology driver.

Due to the innovations 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 enabling existing rule-based technologies to interoperate 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 only useful in the current environment, but are easily extensible in order to deal with the evolution of rule technology and other enabling technologies. RIF's mission is part of W3C's larger goal of enabling the sharing of information in forms suited to machine processing:

1. Rules themselves represent a valuable form of information for which there is not yet a standard interchange format, although significant progress has been made within the RuleML Initiative and elsewhere.
2. Rules provide a powerful business logic representation, as business rules, in many modern information systems.
3. Rules are often the technology of choice for creating maintainable adapters between information systems.
4. As part of the Semantic Web architecture, rules can extend or complement the OWL Web Ontology Language to more thoroughly cover a broader set of applications, with knowledge being encoded in OWL or rules or both.

The purpose of this RIF-UCR document is to describe the use cases that guided the design of RIF and to document the design 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 languages and enables platform-independent interoperation between them (via interchange of RIF rules). The use cases presented illustrate some of the principal ways in which RIF can provide benefits. RIF can promote innovation and development by fostering collaborative work and providing new opportunities for third-party services. RIF can promote e-commerce by providing interoperability across vendor platforms.

W3C Working Group Note

RIF can promote 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 are representative of the types of application scenarios that RIF is intended to support. The use cases illustrate the utilization of the current RIF dialects. More importantly, 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 scenarios by RIF-compliant applications and systems.

Although in this document the discussion of requirements precedes the discussion of use cases, the development of the use cases preceded, and in fact motivated, the development of the requirements. For the most part, requirements were not established unless there was at least one use case, or straightforward modification of a use case, from which that requirement could be derived. There are some exceptions to this rule: for example, some requirements were already defined as constraints in the working group charter and were not necessarily derived from any specific use case. The fulfillment of such requirements is discussed with respect to the existing RIF dialects.

Note that in fact not all use cases can be represented in a straightforward manner within all RIF dialects. In particular, several use cases cannot be represented within the RIF dialect BLD . This is due first, to the fact that BLD is strictly less expressive than first-order logic (since negation is not expressible), and second, to the fact that some use cases are most naturally represented in languages more powerful than first-order logic, such as modal logics. Some use cases that cannot be represented within BLD can be represented within the RIF dialect PRD, which includes the construct of negation as failure. Following the presentation of each use case in Section 5 is a note indicating in which of the current RIF dialects a use case can be represented naturally, sometimes accompanied by discussion.

It should be noted that even for use cases which cannot be represented in a straightforward manner within some RIF dialect, there may be other approaches that would enable representation. For example, although epistemic operators such as Know and `Believe` are most straightforwardly represented within a modal logic, it is generally possible to represent such operators in a first-order logic using a possible-worlds semantics. For certain use cases which cannot be naturally represented within some dialect, the extent to which this can be done will be discussed.

2 Goals

RIF is primarily intended to be an effective means of exchanging rules that has the potential to be widely adopted in industry and that is consistent with existing W3C technologies and specifications.

2.1 Exchange of Rules

The primary goal of RIF is to facilitate the exchange of rules.

2.2 Consistency with W3C specifications

RIF is intended to be a W3C specification that builds on and further develops the existing range of specifications that have been developed by the W3C. This implies that existing W3C technologies should fit well with RIF.

2.3 Widescale Adoption

It is an explicit goal of the W3C that RIF will have maximal potential for widescale adoption. The wider the adoption of the specification, the more effective rule interchange becomes. This is known as the "network effect."

These goals, along with the use cases in Section 5, motivate the requirements in Section 4.

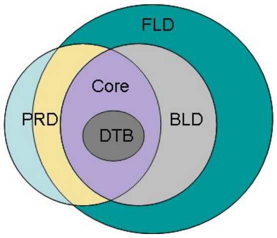
3 Structure of RIF

RIF is described by a set of documents, each fulfilling a different purpose and catering to possibly different audiences. The three RIF dialects that are part of the W3C recommendation and that can be used to represent RIF use cases are:

- RIF Core Dialect, which provides a standard, base level of functionality for interchange.
- Basic Logic Dialect and Production Rule Dialect provide extended functionality matching two common classes of rule engines, namely derivation rules and production rules.

Along with these three dialects, RIF includes related documents: RIF Framework for Logic Dialects, RIF Datatypes and Built-Ins 1.0, RIF Test Cases, RIF RDF and OWL Compatibility, RIF Use Cases and Requirements, OWL 2 RL in RIF, rdf:PlainLiteral, RIF Combination with XML data, RIF Primer, and RIF In RDF. For an overview of RIF, see the RIF Overview.

RIF is designed as a family of RIF dialects as shown in the following Venn diagram:



Each dialect is a collection of components that work together, forming an interlingua. New dialects are needed for those cases in which no existing dialect provides the required rule-language features for interchange.

The RIF Framework for Logic-based Dialects (RIF-FLD) describes mechanisms for specifying the syntax and semantics of logic-based RIF dialects through a number of generic concepts. In general, a logic-based RIF dialect should specialize these general mechanisms; justifications should be provided for cases where this is not done. This specialization may include leaving out some elements of RIF-FLD in order to produce a dialect's concrete syntax and model-theoretic semantics. The RIF dialects that are part of the W3C RIF recommendation are RIF Basic Logic Dialect (RIF-BLD), which is fully specified using RIF-FLD, and RIF Production Rules Dialect (RIF-PRD), which is partially specified using RIF-FLD and supplemented by an operational semantics.

RIF-BLD (Basic Logic Dialect) is a specialization of RIF-FLD capable of representing definite Horn rules with equality and enhanced with a number of syntactic extensions to support expressive features such as objects and frames, internationalized resource identifiers (IRIs) as identifiers for concepts, and XML Schema data types.

RIF-PRD (Production Rules Dialect) specifies a production rules dialect to enable the interchange of production rules. The condition language of RIF-PRD is defined in Core as a common subset of RIF-BLD and RIF-PRD.

RIF-Core (Core Dialect) specifies a common subset of RIF-BLD and RIF-PRD that includes RIF-DTB.

The normative syntax for RIF dialects is a concrete XML syntax. A non-normative presentation syntax is additionally specified for each dialect to allow a more easily readable and compact presentation of language fragments (such as examples).

4 Requirements

The goals and use cases motivate a number of requirements for a Rule Interchange Format. The Working Group has approved the following requirements:

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 rule languages covered by that dialect and RIF without changing the rule 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, no single 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 one 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 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, and 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.

4.13 OWL data

RIF must cover OWL knowledge bases as data where compatible with RIF semantics.

4.14 RDF data

RIF must cover RDF triples as data where compatible with RIF semantics.

4.15 Dialect Identification

The semantics of a RIF document must be uniquely determined by the content of the document, without out-of-band data.

4.16 XML syntax

RIF must have an XML syntax as its primary normative syntax.

4.17 XML types

RIF must support an appropriate set of scalar datatypes and associated operations as defined in XML Schema part 2 and associated specifications. See the charter on Datatype support.

4.18 Merge Rule Sets

RIF must support the ability to merge rule sets.

4.19 Identify Rule Sets

RIF must support the identification of rule sets.

4.20 XML data

RIF should be able to accept XML elements as data.

4.21 Internationalized text

RIF must support internationalized text — that is, text that conveys additional information in terms of a language tag.

5 Use Cases

A use case for RIF consists of a description of a problem together with a description of a solution. Most of these solutions often can be implemented using one of the existing RIF dialects in the W3C recommendation (Core, BLD, or PRD); others require the specification of a new RIF dialect.

The set of use cases is intended to do the following:

- motivate the need for a rule interchange format in a broad range of application domains and industrial sectors;
- provide scenarios that explain the benefits of a rule interchange format in such scenarios and motivate the design of RIF;
- motivate the working group's set of requirements for a rule interchange format; and
- guide users to RIF's specified dialects.

The set of use cases was developed over several years. Nearly fifty 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 relatively small set of use cases that would refer to popular 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 three RIF dialects (Core, BLD, and PRD). All use cases except for 5.2 and 5.4 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 eBusiness 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 contributors.

John is negotiating an electronic business contract regarding the supply of various types of items that Jane'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 / ontology) for the goods and services involved. In this case an XML schema and appropriate test XML documents are interchanged with their rules. Since John and Jane run applications based on different vendors' rule engines and rule languages, they interchange the rules using RIF; both vendors used can interpret the XML schema and data, and produce the results as an amended XML document. John's company defines its purchase orders in terms of an XML description of goods, packaging, delivery location, and date with delivery and payment rules. A rule proposed by John might be the following:

```
If an item is perishable and is delivered to John more than 10 days after the scheduled delivery date, then he will reject the item.
```

Jane replies with some 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 18.7% 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.

Future requests for the supply of items by John's company are defined on their purchasing web site, as the appropriate XML schema and a RIF rule set (or rule sets). This allows Jane's company and its competitors to respond electronically with XML cost sheets. Suppliers respond with multiple cost sheets with different variations on the RIF rules proposed by John's company, allowing John's company to review the alternative rules with their associated costs to determine whether they, as a business, would benefit by relaxing or adding new rules as proposed by suppliers.

Supported by: RIF-Core, RIF-BLD, RIF-PRD

The rules of this use case can be adequately represented in RIF-Core (and likewise, in RIF-BLD and RIF-PRD) by using (ordered) relations, as e.g. in standard Logic Programming, or by using unordered relations and named arguments, or in a more object-oriented encoding using frames.

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

This use case concerns the ability of parties involved in formal transactions or procedures, e.g., credit card authorization of a purchase or access of private medical records, to express and protect their interests within a policy-governed framework. The goal is to formally encode the preferences, priorities, and responses of the parties in such a way that the overall policy can work as intended while still providing opportunity for automatic negotiation of terms as long as it is allowed by the policy. Utilizing RIF in this use case would extend the scope of this technology, affording a higher degree of interoperability, as well as enabling re-use and sharing of preferences through interchange. The detailed scenario below shows how this would work.

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 eShop's 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 automatically establish trust with the goal of successfully completing the transaction.

Policies are themselves subject to access control. Thus, rule interchange is necessarily 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 rule interchange between the two systems.

When Alice clicks on a "buy it" button at eShop's Web site, Emptor takes over and sends a request to eShop's site. Venditor 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 privacy protection is improved. In this example, Venditor reveals part of its policy in the form of rules to enable Emptor to choose how to answer, i.e., to decide which credentials and required information to disclose.

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

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

By disclosing (interchanging) the above rule, Emptor asks Venditor to provide credentials saying that it belongs to the Better Business Bureau, Alice's most trusted source of information on online shops. eShop has these credentials. Moreover, its policy contains a rule authorizing release of this credential to any potential purchaser. Hence, Venditor passes the credential to Emptor. However, before Emptor can disclose Alice's credit card information to Venditor, it must still check whether disclosing all requested information breaks 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 for this purchase, Alice birth date is not necessary, Alice's constraints are respected. Emptor therefore provides Alice's credit card information to Venditor.

Companies that provide software such as Venditor and Emptor can make use of RIF in several ways. First, the rules expressing Alice's and/or eShop's policies could be expressed in different rule languages but still work with a common software, using RIF-based interchanges. Second, RIF would enable Venditor and Emptor to work together in real time, even if these automated agents are products of different companies using 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 Venditor 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

This use case demonstrates how RIF leads to increased flexibility in matching the goals of end-users of a service/device, with the goals of providers and regulators of such services/devices. RIF can do that because it enables deployment of third party systems that can generate various suitable interpretations and/or translations of the sanctioned rules governing a service/device.

This use case concerns *Dynamic Spectrum Access for wireless communication devices*. Recent technological and regulatory trends are converging toward a more flexible architecture in which reconfigurable devices may operate legally in various regulatory and service environments. The ability of a device to absorb the rules defining the policies of a region, or the operational protocols required to dynamically access available 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.

In this use-case we suppose a region adopts a policy that allows certain wireless devices to opportunistically use frequency bands that are normally reserved for certain high-priority users. The decision by the European Union to allow "Dynamic Frequency Selection" (DFS) use of the 5 GHz frequency band by wireless systems, a band intermittently used by military and weather radar, is a recent example.)

Suppose the policy states:

```
A wireless device can transmit on a 5 GHz band if no priority user is currently using that band.
```

How does a device know that no priority user is currently using a band it wants to use? The answer will depend on the specific capabilities of the device. One type of device may answer this question by sensing the amount of energy it is receiving on that band. That is, it might employ the rule:

```
If no energy is detected on a desired band then assume no other device is using the band.
```

A second type of device may get information from a control channel that lets it know whether the desired band is being used by a priority user. That is, it might employ the rule:

```
If no control signal indicating use of a desired band by a priority user is detected then assume the band is available.
```

So each type of device will need to employ different "interpretations" or "operational definitions" 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 (for each of the two types of device). That means that 20 different versions of the policy must be written, tested, and maintained.

Enter RIF. The 10 manufacturers form a consortium. This is a third-party group that is responsible for translating regional policies into RIF. When it does so, however, it provides different versions corresponding to the possible interpretations (operational definitions) of the policy. So in this case, 2 RIF versions of the DFS policy are provided for the 2 types of device mentioned above. Each of these RIF specifications can be automatically translated into the appropriate rule-platform provided that a RIF-Compiler for the target device architecture exists. Clearly it will be in the interest of each device manufacturer to develop such compilers. That is because the manufacturer only needs to develop such a compiler *once* for every architecture it owns. Contrast that investment with having to produce, test, and maintain different versions of various policies over the lifetime of a product.

This arrangement also allows the overall process to be organized in a fashion that maintains the natural division of labor in the corresponding division of artifacts produced by that labor. The policy and its various interpretations are written and maintained in platform-independent artifacts (RIF); knowledge about how to translate from RIF to a particular device architecture is maintained in the compilers. A change in policy is inserted at the top level in the policy artifact hierarchy where it should be; possible operational interpretations of that change are inserted at the next level down; and the implementation implications for the various device architectures is generated automatically at the lowest level.

Partly supported by: RIF-PRD

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 on, e.g., event streams over sensor data.

This use case is not supported at all by RIF-BLD, and by extension, by RIF-Core. These dialects do not support negation, and the use case makes explicit mention of negation, e.g. for representing that "no priority user is currently using the band."

5.4 Access to Business Rules of Supply Chain Partners

A business process (BP) designer designs processes that can span multiple departments in the same business as well as other business partners. A classic example of this is the integration of supply chain business processes which typically involve multiple partners. Supply chain integration involves exposing a certain amount of business logic between partners as well as integrating processes across partners. In such activities it is therefore often necessary to access or invoke rules that originate in other ownership domains.

A key part of a business process is the logic used to make decisions within the process. Such logic is often coded in rules because rule languages are easier for BP designers to understand and manipulate than procedural code (as in Java) -- although both forms of business logic are prevalent. When business logic is represented in different rule languages, designing an integrated process presents a significant burden to the BP designer.

Two primary integration modalities are possible: importing the different rule sets into a single engine and processing them in a uniform manner, or accessing the rule sets by querying remote engines and processing the results. Each modality has its uses and contra-indications. Merging rule sets of partners may not be permitted in cases where there are strong ownership boundaries.

For example, in an insurance adjustment process, the inspection of a damaged vehicle is often performed by independent inspectors. The critical factor in determining how an insurance claim will proceed is whether the damage results in a total loss or whether a repair is feasible:

```
If an inspector believes a vehicle is repairable then process the claim as a repair; otherwise process the claim  as a total loss.
```

Determining whether a vehicle is repairable depends on the processes executed by the inspector and therefore cannot be directly integrated into the insurance company's own adjustment process. The insurance company effectively queries the inspector's logic rather than incorporating the inspector's logic into its rule set. Moreover, within the adjustment process, the overall flow will be quite different for repairable claims and total loss claims.

Even in the case of a single company, which is nominally under a common ownership domain, information and business logic are often controlled by multiple stakeholders. For example, a large company will often be organized into semi-independent profit centers (often known as business units). Each unit will be motivated differently, will have different ontologies and business logic, and may use different rule languages to represent their logic. This is particularly the case when one company acquires another company.

RIF could be used to permit the BP designer a unified view of the different partners' business rules in designing the process, while at the same time permitting the partners to continue to leverage their own business rules and their own technologies.

Realizing this sort of unified view of business rules in a deployed BP will depend on both technical and non-technical factors. Even when all parties are required to use a common rule language, there may be compelling ownership issues that mitigate against a simple merge of rule sets. When merging rule sets is not possible, a query-style access to partners' business rules may be used. In this way, RIF permits a unified dynamic view of the business rule logic independent of the rules' original form.

For this to be viable from a business perspective the semantics of the rules and query exchange must be completely predictable and should be loss-less.

Not supported, in a straightforward manner, by the current RIF dialects

The example business rule in this use case makes explicit reference to belief, which as the discussion below indicates, is not supported by either BLD or PRD (and thus by implication not supported by Core). The reader may note that this business rule could be changed to use another term such as "determine": e.g.,

```
If an inspector determines that a vehicle is repairable then process the claim as a repair; otherwise process the claim  as a total loss.
```

Using another term that has a similar semantics --- i.e., indicating in some way the inspector's thought process and mental state --- does not alleviate the problem.

BLD and PRD do not support modal operators or predicates on quoted sentences; therefore standard representations of knowledge, belief, and/or uncertainty cannot be specified in BLD and PRD. Even without such constructs, it can be possible to represent knowledge or belief using the semantics of possible worlds [Moore1980]. That is, one can get the effect of saying that an agent A believes proposition P by saying that P holds in all worlds that are belief-accessible to P. For example, to get the effect of saying

```
Believe(john,overCreditLimit(bill,t))
```

one says instead

```
holds(overCreditLimit(bill),t1) :- beliefAccessible(john,t,t1)
```

The translation between sentences involve operators such as Know and Believe and sentences that make direct use of possible worlds relies on the possible-worlds semantics, which posits properties on the knowledge-accessibility or belief-accessibility relation. These relations correspond to axioms on the Know and Believe operators.

Such a strategy could work, to a certain extent, in RIF-PRD, but would not work in RIF-BLD. This is because expressing the standard axioms on belief or knowledge would require the use of negation, which is not supported by BLD.

For both PRD and BLD there is an additional consideration, which is that the notion of reification is in general not supported by the dialects of RIF.

</p>

5.5 Managing Inter-Organizational Business Policies and Practices

This use case concerns organizations that acquire rule sets from external sources and have to integrate these new rule sets into their existing rule bases. Such rule sets may be acquired in the following ways:

- An organization may buy rule sets from expert sources
- An organization may use rule sets from shared interest groups such as trade associations
- A component of a distributed organization may acquire rules when a rule set is distributed across a distributed organization. In such case, there may be different localization requirements in different regions and locations, entailing a variety of integration challenges in these various locations and component organizations.

The following scenario examines these different methods of acquisition and the various types of integration and management issues that may arise.

This scenario uses the (fictitious) car rental company, EU-Rent, used as the case study in the Semantics of Business Vocabulary and Business Rules Specification. The EU legislation discussed is also fictitious, as are the consulting companies CarWise and AutoLaw.

EU-Rent's corporate HQ deals with CarWise, a consulting company with expertise in managing fleets of vehicles. One service CarWise offers to its clients is negotiating with EU regulators to clarify regulations.

An EU regulator issues a directive dealing with insurance for vehicles owned by corporations. CarWise agrees with the regulator on an acceptable interpretation, and provides EU-Rent (and its other car rental clients) with two sets of rules:

- A business policy, stating that every car rental must be insured for damages to third parties.
- A supporting rule set, addressing levels of required coverage, tax calculation in different EU countries, liabilities in rentals that span multiple countries, and reporting of compliance with this business policy.

EU-Rent decides that it will maintain its compliance documentation electronically. CarWise then provides EU-Rent with an additional rule set for electronic compliance documentation, including such rules as:

```
Each tax schedule must have electronic signatures from two EU-Rent employees who are at least at the level of manager.
```

Before it can use the two general rule sets, EU-Rent needs to connect them to the relevant data sets in its IT systems, e.g. relate the EU country-specific taxation rules to the relevant record types in its databases.

EU-Rent corporate HQ subsequently decides that the cost of third-party insurance will be built into the basic cost of each rental and not shown as a separate item on the rental contract; this will ensure that it can never be omitted from rentals or disputed by renters. It then sends three rule sets to its operating companies in the EU:

- The rule set for car rental insurance (from CarWise), including the basic policy and the supporting rule set.
- The rule set for electronic compliance documentation (also from CarWise).
- Its own rule set for building insurance into the basic rental cost.

The operating companies then have to localize the rule sets for their countries of operation. For example, in the UK, another consulting company, AutoLaw, advises EU-Rent of rules for placing aggregate insurance for large fleets with more than one insurer in order to spread the risk:

```
For fleets of more than 200 vehicles, fleet insurance policies must be placed with at least 3 insurers, each of whom covers at least 25% of the risk.
```

A timing issue makes it difficult for EU-Rent UK to strictly comply with this directive. EU-Rent UK has some existing insurance policies in place, which provide third-party insurance as an explicit item, and it cannot get refunds on early termination. It therefore asks corporate HQ for a temporary dispensation: that it can continue its existing insurance until it expires, and then switch to the new rules.

EU-Rent HQ permits this, not just for the UK but for any of its operating companies that have similar insurance arrangements. To ensure that this dispensation is temporary, it adds a new rule:

```
Insurance policies that provide separate third-party coverage must not be renewed.
```

EU-Rent HQ is also concerned about meeting deadlines for electronic filing. It introduces a new rule that it distributes to operating companies:

```
Each electronic compliance document must have its required electronic signatures 48 hours before its filing deadline.
```

This rule is meant to be implemented as follows: If '48 hours before its filing deadline' passes, and the electronic signatures are not present, then the operating company's rules system must report the out-of-compliance situation, and subsequently wait for the responsible managers to provide the signatures.

Partially supported by RIF-PRD

The business rules in this use case express norms which requires the representation of deontic operators for obligations and permissions. Deontic operators are generally represented by modal operators, as in [Horty2001]. As was discussed above for knowledge and belief, it is not possible to express modal operators directly within RIF dialects. However, one can use the technique discussed above, of reasoning directly within possible worlds. This is possible within PRD though not within BLD or Core.

Moreover, in a meta programming approach it is possible to express deontic reasoning rules with RIF-PRD. Using this approach, too, one would not be able to use BLD or Core, since some business rules express negated deontic norms.

5.6 Rule set Integration for Medical Decision Support

Decision support systems aid in the process of human decision making, especially decision making that relies on expertise. Reasoning with rules is an important part of this expert decision making. For complex decision support systems, it is expected that rules will be furnished by a variety of different sources, including ontologies, knowledge bases, and other expert systems. This use case illustrates how RIF makes it possible to merge rule sets from diverse sources in diverse formats into one rule-based system, thereby enabling inferences that might otherwise have remained implicit.

Medical decision support systems, such as the ones discussed below, might use rules from pharmaceutical knowledge bases, laboratory knowledge bases, patient databases, and medical ontologies. For example, a large amount of information on therapeutic medications (drug taxonomies, indications, contraindications, and clearance times) and diseases (disease taxonomies, etiologies, and symptoms) is contained in existing ontologies such as SNOMED Clinical Terms®. Rules can be used to express therapeutic recommendations, to formulate queries about relevant prescriptions for a patient, and to assess the effectiveness of a treatment.

The following scenario illustrates how rule-interchange would be used in various medical decision support systems to support the following functionalities:

- Improving situation assessment, e.g., determining when a patient needs to be put on medication, or have his medication switched.
- Prescribing a course of action, e.g., determining which drug is best for a patient in a particular circumstance.
- Improving event planning, e.g., determining whether a patient can be scheduled for a procedure given the medication that he is currently taking.

Bob, 62 years old and reasonably healthy, has been going to his internist, Dr. Rosen, for several years for control of his Type II diabetes. Dr. Rosen has been using the `AutoDoc` system to help him decide when to switch to medications and which drugs to prescribe. The `AutoDoc` system uses two sources when making its recommendations: a laboratory knowledge base giving particular results for patients and specifying when these results are out of normal range, and a pharmaceutical knowledge base giving guidelines for the use of medications. Automated reasoning with rules from these combined sources is possible if the rules are first mapped to RIF. Here are two specific examples of such synergistic effects.

This scenario discusses the fictitious expert systems `AutoDoc` and MEDIC. In the interest of readability and brevity, the information and rules presented in the following scenario may not precisely capture the current state of medical knowledge and best practices in this field, but may be somewhat simplified.

Originally Bob's diabetes was controlled through diet and moderate exercise. In time, however, Bob's blood glucose level began to rise, even under this regimen. Due to Bob's elevated `HbA1c` level (which indicates one's average blood sugar level over the last several months), Dr. Rosen prescribed oral medication for Bob. He was forced to change Bob's medication a number of times over the course of a year. He first prescribed Precose, an oral alpha-glucosidase inhibitor, but had to discontinue this medication due to undesired side effects. He then prescribed several sulfonylurea drugs, Micronase and Glucotrol, to no avail. Bob's lab results still indicated an elevated `HbA1c` level. The following rule from the *laboratory knowledge base* suggests that Bob's treatment at that time was not effective:

```
If a Type II diabetes patient's current level of HbA1c is high, then the patient's current treatment is considered to be ineffective.
```

To deal with this problem, Dr. Rosen was about to prescribe Glucophage (metformin, one of the biguanides) 850 mg, 3 times a day, when as usual, he double checked his prescription with the `AutoDoc` system. The system, based on the following guidelines from the *pharmaceutical knowledge base*, informed Dr. Rosen that he should have prescribed an oral bitherapy (two medications, each of which controls blood sugar levels) instead of a monotherapy.

```
If an oral monotherapy at recommended doses of a sulfonylurea or biguanide, combined with lifestyle changes, is ineffective, then the monotherapy should be replaced by an oral bitherapy.
```

Based on the recommendation from `AutoDoc`, Dr. Rosen switched Bob's prescription to Glucophage and Avandia (rosiglitazone, one of the thiazolidinediones).

Bob recently suffered a concussion and has become increasingly forgetful. He went to see a neurologist, Dr. Cervello, who prescribed a contrast MRI (Magnetic Resonance Imaging). When asked about current medication, Bob told Dr. Cervello that he was taking Glucotrol to control his diabetes, forgetting that he had been switched to Glucophage. This was potentially problematic, since Glucophage should not be taken close to the administration of a contrast injection.

Fortunately, when Bob went to the lab to schedule his MRI, the medical receptionist pulled up MEDIC (Medical Event and Drug Interaction Consultant), the hospital's new automated system, which checks for incompatible medical events and/or drugs (e.g., liposuction scheduled during pregnancy, blood thinners prescribed before surgery, etc.).

MEDIC uses a variety of sources in its reasoning, including:

- the pharmaceutical knowledge base, described above
- the patient databases, which gives the patient record, including the medications a patient is currently taking
- the hospital medical event protocol knowledge base, which details the protocol used for different medical procedures

In this case, MEDIC uses all three sources and pulls up the following information:

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

MEDIC therefore determines that Bob should not be taking the contrast MRI at this time.

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

Supported by RIF-PRD

The use case requires the formalization of changeable states (fluents) which can be represented by e.g. an event calculus axiomatization [KS86]. However, to formulate the classical event calculus as a set of Horn clause rules so that it could be run as a Prolog program, negation as failure is needed, which is not supported by RIF-BLD or Core. The use case can be represented in RIF-PRD 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. RDF and OWL), in order to provide greater expressive power than is provided by either formalism alone. Ontology languages, for example, typically provide a richer language for describing classes (unary predicates). Rules, on the other hand, typically provide a richer language for describing dependencies between properties (binary predicates), and may also support higher-arity predicates.

Rich domain models combining both rules and ontologies are often needed in domains such as medicine, biology, e-Science, and Web services. In such domains, several actors and/or agents may need to interchange the data, ontologies, and rules that they work with. An example is the use of such a domain model in an application that aims at assisting the labeling of brain cortex structures in MRI images. In this case, an OWL ontology is used to capture knowledge about the most important brain cortex anatomical structures, and a rule base is used to capture knowledge about mereological and spatial dependencies between properties.

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

```
If MAE X is bounded by Z and MAE Y is also bounded by Z then X is connected to Y.
```

Benefits of interchange via RIF include the ability to collaboratively develop and share valuable 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-PRD

The use case uses RIF-SWC 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 that independent data sources may have rather divergent information 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, such mappings can be published across the Semantic Web, enabling an enterprise or community to progressively build up a rich network of mappings unifying the information models.

Information mapping and integration problems like this arise in many diverse domains including health care, travel planning, IT management, and customer information management. The following scenario comes from the world of IT systems management.

Vlad has been given the job of analyzing how exposed his division's business processes are to changes in their IT maintenance contracts. He has three sources of information to combine:

- a report on application services and associated servers, databases, and networks (from the IT department)
- a maintenance contracts database (from the finance department)
- a registry indicating which business processes use which IT services (from the business planning group)

Each of these sources is in a different form but can be mapped into RDF to simplify access. However, they all have different information models. The IT report is too fine-grained: it talks about routers and interface cards while Vlad only needs to identify servers and pick out some generic dependency relations. On the other hand, the finance database models the world in terms of physical assets such as racks, which is too coarse-grained.

First, Vlad creates simple mapping rules to create a uniform, simplified view of the data in terms of a small number of concepts -- Server, BusinessProcess, and Dependency. This involves rules such as:

```
If x is a ComputeNode in Rack r
   and Rack r is in Cage c
   and mc is a MaintenanceContract for Cage c
      then x is a Server with MaintenanceContract mc

If x is a ComputeNode with a NetworkInterface with Port p
   and app is an Application running on Port p
      then x is a Server that hosts app

If bp is a BusinessProcess that uses Application app
      then bp has a Dependency on app
```

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

```
If bp is a BusinessProcess that has a Dependency on Application app
  and x is a Server with MaintenanceContract mc that hosts Application app
     then bp has a Dependency on mc
```

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.

Supported by RIF-BLD, RID-PRD, and RIF-Core

5.9 BPEL Orchestration of Rule-Based Web Services

Rule-based Web services depend on the use of XML data for their request and response format. The involved rules must be able to access and compare XML data in their conditions and modify and generate XML data in their actions.

An existing commercial credit approval service deployed as a Web service takes an applicant credit request document as input and returns an approval 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. BPEL first passes the credit request document to the decision service to determine, using rules, whether enough information (Social Security number, mother's maiden name, etc.) is available to request a credit history. If so, BPEL then requests a credit history from the history providing service and passes the credit history document to the decision service to be evaluated. Based on the evaluation, credit is approved or denied.

Because the rule engine is part of a Web service, existing BPEL diagramming and execution facilities can be used to integrate rules into this credit approval service. The credit evaluation model can be changed easily using GUI tools to customize rules. Using RIF would further improve the situation. First, the credit history vendor could supply a default set of rules for evaluating its histories. Second, there would be several rule editing and customization tools from different RIF-compatible vendors to tailor the rules to meet specific business objectives.

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:

- number of times a payment was 60 days late
- debt-to-income ratio
- number of foreclosures or repossessions
- number of garnishments
- number of liens
- bankruptcy

A red flag above the threshold results in denial of credit.

Rules in the second rule set increment a credit score variable. For example:

```
If applicant owns residence then add 40.
If applicant rents then add 30.
If applicant has lived at current address 2 to 4 years then add 20.
If applicant's income is under 20000 then add 10.
If applicant's income is between 40000 and 50000 then add 40.
```

The third and final rule set compares the applicant's credit score and income to threshold values, and makes the final decision to approve or deny credit to the applicant.

The decision and supporting rationale is returned from the decision service as an XML document. This decision document is then used to construct the reply to the original credit approval request.

Supported by RIF-PRD

W3C Working Group Note

This use case requires a goal-driven solution which makes use of assert and retract to update a global score value. Assert and retract are not supported by BLD or Core; indeed, there is no such similar notion within BLD or Core.

5.10 Publishing Rules for Interlinked Metadata

The Semantic Web includes technologies (e.g., RDF) that allow metadata to be published in machine-readable form. Currently, this information is mostly enumerated as a set of facts. It is often desirable, however, to supplement such facts with rules that *capture implicit knowledge*. To maximize the usefulness of such published rules, a standard rule format such as RIF is necessary.

One case involves extending current standards for metadata publication with rules in order to express implicit knowledge. Suppose that the International Movie Database (IMD) publishes its metadata and rules in a machine readable format at http://imd.example.org. Besides the ground facts, which can be expressed in RDF, the metadata might also have general rules like the following:

```
Every science fiction movie is a movie.
Every movie produced before 1930 is black and white.
```

Such rules allow data to be published more concisely by expressing knowledge that, without these rules, is implicit. This can greatly simplify the maintenance of data, guard against inadvertently introduced inconsistencies, and reduce storage requirements.

Published rules also allow combining data from different sources to exploit this knowledge. Consider an alternative database of movies published at http://altmd.example.org. In addition to metadata, it again publishes its own rules:

```
All movies listed at http://altmd.example.org but not listed at http://imd.example.org are independent movies.
All movies with budgets below 5 million USD are low-budget movies.
```

Publication of rules with explicit references to other rule sets allows the definition of knowledge dependent on explicitly specified remote sources. Such explicitly specified *scope* is important in the Web environment, since it can reduce the danger of unintended interference from rules published at other remote sources, which may be exporting their own predicates.

Another example of such explicit referencing, which also illustrates implicit person-centric metadata, involves published rules being used to specify how to use other metadata, e.g. in the form of a widespread vocabulary such as FOAF or a standard exchange format like iCalendar. For example, FOAF user Charlie might choose to complement his normal FOAF profile with his preferences about which of his phone numbers should be used depending on his iCalendar schedule:

```
If Charlie is currently attending a public talk according to http://charlie.example.org/calender.ical
    then leave him a voicemail message

If Charlie is currently in a meeting according to http://charlie.example.org/calender.ical
  and the importance is high
    then call his cell number

If Charlie currently has no appointments according to http://charlie.example.org/calender.ical
    then call his office number
```

RIF should allow extending current standards for metadata publication by enabling such implicit knowledge to be captured via rules and allowing metadata and rules distributed over different sources to be interlinked. In a manner similar to how HTML links human-readable Web pages, RIF should permit linking metadata on the Web to support new kinds of "intelligent" crawling and search.

Supported by RIF-PRD

Since negation is not supported by RIF-BLD or RIF-Core it is not possible to express ... *but not listed at ...*.

6 Conclusion

The goal of the RIF working group is to provide representational interchange formats for processes based on the use of rules and rule-based systems. These formats act as interlingua to interchange rules and integrate with other languages, in particular (Semantic) Web mark-up languages.

As can be seen by studying the use-cases presented in this document, rules are used to perform a wide variety of tasks, and therefore rule-based systems are not monolithic. Rules have been used, for example, to perform and validate inference, perform calculations, direct the flow of information, enforce integrity constraints on databases, represent and enforce policies, control devices and processes in real time, and determine the need for human intervention.

In light of this diversity, rather than being a single all-encompassing format, RIF consists of several dialects, each dialect serving a particular set of related rule languages. The key idea is to attain the goal of interoperability (via interchange of RIF rules) *within* each dialect. This should allow the main benefits of RIF to be realized.

RIF has been designed in such a way that it is possible to create new dialects (extensibility) according to the overall goals and the general requirements of RIF, as well as to update existing dialects (upwardly compatible). This is in keeping with the working group charter's call for an extensible format.

7 References

[Horty2001]
Agency and Deontic Logic, John F. Horty, Oxford University Press, 2001.

[KS86]
Kowalski, R.A. and M.J. Sergot, *A logic-based calculus of events.* New Generation Computing, 1986. 4: p. 67-95.

[Moore80]
Moore, R.C., *Reasoning about Knowledge and Action,* SRI TR-191, 1980.

[OWL-Reference]
OWL Web Ontology Language Reference, M. Dean, G. Schreiber, Editors, W3C Recommendation, 10 February 2004. Latest version available at http://www.w3.org/TR/owl-ref/.

[RIF-BLD]
RIF Basic Logic Dialect (Second Edition) Harold Boley, Michael Kifer, eds. W3C Recommendation, 5 February 2013, http://www.w3.org/TR/2013/REC-rif-bld-20130205/. Latest version available at http://www.w3.org/TR/rif-bld/.

[RIF-Core]
RIF Core Dialect (Second Edition) Harold Boley, Gary Hallmark, Michael Kifer, Adrian Paschke, Axel Polleres, Dave Reynolds, eds. W3C Recommendation, 5 February 2013, http://www.w3.org/TR/2013/REC-rif-core-20130205/. Latest version available at http://www.w3.org/TR/rif-core/.

[RIF-PRD]
RIF Production Rule Dialect (Second Edition) Christian de Sainte Marie, Gary Hallmark, Adrian Paschke, eds. W3C Recommendation, 5 February 2013, http://www.w3.org/TR/2013/REC-rif-prd-20130205/. Latest version available at http://www.w3.org/TR/rif-prd/.

8 Appendix: Change Log (Informative)

This appendix summarizes the main changes to this document since its publication as a Working Draft.

Changes since the draft of December 18th, 2008:

- the section describing the structure of RIF has been updated with newer RIF documents;
- all code examples have been removed from the document;
- each use case concludes by indicating the support of 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;
- an informative change log section has been added;
- the distinction between general and basic requirements has been removed, thereby flattening the list of requirements.