



RIF Use Cases and Requirements

W3C Editor's Draft 29 September 2009

This version:

<http://www.w3.org/2005/rules/wg/draft/ED-rif-ucr-20090929/>

Latest editor's draft:

<http://www.w3.org/2005/rules/wg/draft/rif-ucr/>

Previous version:

<http://www.w3.org/2005/rules/wg/draft/ED-rif-ucr-20090904/> ([color-coded diff](#))

Editors:

Adrian Paschke, Freie Universitaet Berlin
David Hirtle, National Research Council Canada
Allen Ginsberg, Mitre
Paula-Lavinia Patranjan, REWERSE
Frank McCabe, Fujitsu

This document is also available in these non-normative formats: [PDF version](#).

Copyright © 2009 W3C[®] (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

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May Be Superseded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 11 documents:

1. [A Guide to RIF](#)
2. [RIF Core Dialect](#)
3. [RIF Basic Logic Dialect](#)
4. [RIF Framework for Logic Dialects](#)
5. [RIF RDF and OWL Compatibility](#)
6. [RIF Datatypes and Built-Ins 1.0](#)
7. [RIF Production Rule Dialect](#)
8. [RIF Use Cases and Requirements](#) (this document)
9. [RIF Test Cases](#)
10. [RIF Combination with XML data](#)
11. [OWL 2 RL in RIF](#)

Summary of Changes

@@@UPDATE

@@@ Include-in-this-round?

Please Comment By 27 October 2009

The [Rule Interchange Format \(RIF\) Working Group](#) seeks public feedback on this Editor's Draft. Please send your comments to public-rif-comments@w3.org ([public archive](#)). If possible, please offer specific changes to the text that would address your concern. You may also wish to check the [Wiki Version](#) of this document and see if the relevant text has already been updated.

No Endorsement

Publication as a Editor's Draft 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](#). This document is informative only. 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](#)
- [2 Goals](#)
 - [2.1 Exchange of Rules](#)
 - [2.2 Consistency with W3C specifications](#)
 - [2.3 Widescale Adoption](#)
- [3 Structure of RIF](#)
- [4 Use Cases](#)
 - [4.1 BLD Use Cases](#)
 - [4.1.1 Negotiating eBusiness Contracts Across Rule Platforms](#)
 - [4.1.2 Negotiating eCommerce Transactions Through Disclosure of Buyer and Seller Policies and Preferences](#)
 - [4.1.3 Interchanging Rule Extensions to OWL](#)
 - [4.1.4 Vocabulary Mapping for Data Integration](#)
 - [4.1.5 BPEL Orchestration of Rule-Based Web Services](#)
 - [4.2 PRD Use Cases](#)
 - [4.2.1 Medical Reasoning Across Integrated Production Systems](#)
 - [4.2.2 Integrating Production Systems in Cases of Business Mergers](#)
 - [4.2.3 Collaborative Policy Development for Dynamic Spectrum Access](#)
 - [4.2.4 Managing Inter-Organizational Business Policies and Practices](#)
 - [4.3 FLD Use Cases](#)
 - [4.3.1 Access to Business Rules of Supply Chain Partners](#)
 - [4.3.2 Ruleset Integration for Medical Decision Support](#)
 - [4.3.3 Publishing Rules for Interlinked Metadata](#)
 - [4.4 Out-of-scope Use Cases](#)
- [5 Requirements](#)
 - [5.1 General](#)
 - [5.1.1 Implementability](#)
 - [5.1.2 Semantic precision](#)
 - [5.1.3 Extensible Format](#)

- [5.1.4 Translators](#)
- [5.1.5 Standard components](#)
- [5.1.6 Rule language coverage](#)
- [5.2 Basic Requirements](#)
 - [5.2.1 Compliance model](#)
 - [5.2.2 Default behavior](#)
 - [5.2.3 Different semantics](#)
 - [5.2.4 Limited number of dialects](#)
 - [5.2.5 Embedded comments](#)
 - [5.2.6 Embedded metadata](#)
 - [5.2.7 OWL data](#)
 - [5.2.8 RDF data](#)
 - [5.2.9 Dialect Identification](#)
 - [5.2.10 XML syntax](#)
 - [5.2.11 XML types](#)
 - [5.2.12 Merge Rule Sets](#)
 - [5.2.13 Identify Rule Sets](#)
 - [5.2.14 XML data](#)
 - [5.2.15 Internationalized text](#)
- [6 Conclusion](#)
- [7 References](#)

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 basic goal of the Rule Interchange Format (RIF) Working Group is 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. This mission of RIF 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 provide a reference to the design of RIF and a guide for users and implementers to the current technical specifications of RIF dialects. RIF-UCR also delivers a structured context for formulating future technical specifications of further RIF dialects. Each dialect targets at a cluster of similar rule languages and enables platform-independent interoperation between them (via interchange of RIF rules). The presented use cases 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. RIF can promote efficient process management through reuse, sharing, and the ability to provide unified views across disparate platforms. Last, but not least, 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 and several accordant critical success factors for RIF. Section 3 summarizes the released RIF dialects and the current structure of RIF. Section 4 presents a set of use cases that are representative of the types of application scenarios that RIF is intended to support. Besides illustrating the utilization of the current RIF dialects, the functionality specified in the use cases, together with the inferred requirements, acts as input for the technical specification of future RIF dialects and for the implementation of various variants of these scenarios by applications or systems that incorporate the existing or newly developed RIF technical specifications. In section 5 several important requirements for RIF are inferred from the goals and use cases. In the main all requirements should have a use case or derivation of a use case from which they are derived. In exceptional circumstances requirements may not be derived from a use case, e.g. when they are already defined as constraints in the working group charter. Their fulfillment is discussed with respect to the existing RIF dialects.

2 Goals

The primary goal of RIF is 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 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 the Rules Interchange Format will have the maximum potential for widescale adoption. Rules interchange becomes more effective the wider adoption there is of the specification -- the so-called "network effect".

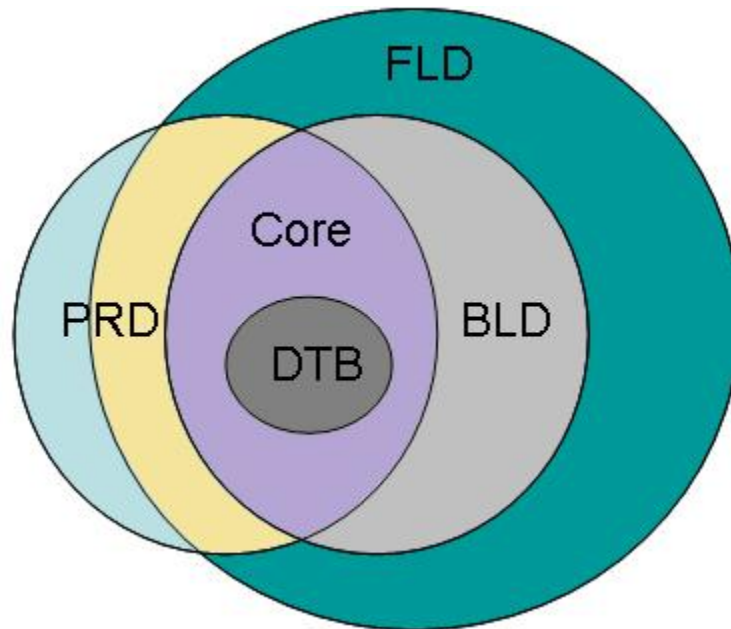
Along with the use cases in the next section, these goals motivate the requirements in Section 5.

3 Structure of RIF

RIF is described by a set of documents, each fulfilling a different purpose, and catering to a different audience. Currently the following set of documents has been released:

- The [RIF-FLD](#) (Framework of Logic Dialects) document describes a framework of mechanisms for specifying the syntax and semantics of logic-based RIF dialects through a number of generic concepts.
- The [RIF-RDF+OWL](#) (RIF RDF and OWL Compatibility) document specifies the interoperation between RIF and the data and ontology languages RDF, RDFS, and OWL.
- The [RIF-DTB](#) (Data Types and Builtins) document describes RIF data types and built-in functions and predicates
- The [RIF-Core](#) (Core Dialect) document specifies a common subset of RIF-BLD and RIF-PRD including RIF-DTB
- The [RIF-BLD](#) (Basic Logic Dialect) document specifies a basic interchange format that allows logic rules (definite Horn rules with equality) to be exchanged
- The [RIF-PRD](#) (Production Rules Dialect) document specifies the RIF production rules dialect to enable the interchange of production rules
- The [RIF-UCR](#) (Use Cases and Requirements) document describes use cases and requirements for RIF
- The [RIF-Test](#) (Test Cases) document describes the test cases developed by the RIF working group

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



Each dialect is a collection of components that works together, forming an interlingua. New dialects are needed when 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. Every logic-based RIF dialect should specialize these general mechanisms or justify why it does not. This specialization may include leaving out some elements of RIF-FLD, to produce its concrete syntax and model-theoretic semantics. Currently, the first two existing RIF dialects are the RIF Basic Logic Dialect (RIF-BLD) and the RIF Production Rules Dialect (RIF-PRD) which is a partial specialization of FLD.

[RIF-BLD](#) (Basic Logic Dialect) is a specialization of RIF-FLD capable of representing definite Horn rules with equality 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 which 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 Use Cases

A use case for RIF consists of a description of a problem together with a solution that either uses an existing RIF dialect or requires the specification of a new RIF dialect.

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

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

The set of use cases were developed over several years. Nearly [fifty use cases](#) documenting the need for a RIF were originally submitted by the 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 cover a broad range of possible requirements. In addition, it was desired that the use cases refer to popular application domains and industrial sectors.

Subsequent to the development of the RIF dialects Core, BLD, PRD, and FLD, the set of use cases were sorted according to the (weakest) dialect needed to express the problem statement and solution of the use case. In addition, several use cases were added in order to highlight features of these dialects.

Below, we give concrete illustrations of how these RIF dialects address various aspects of the use cases.

The button below can be used to show or hide the RIF examples.

Show Examples

Editor's Note: the given examples and used presentation syntax in this version of the UCR working draft are still under development

In order to enhance readability and avoid the appearance of syntactic prejudice, we have deliberately avoided the use of formal notation in representing rules in these use cases. Instead, we will use the RIF presentation syntax (of the RIF dialects).

4.1 BLD Use Cases

4.1.1 Negotiating eBusiness Contracts Across Rule Platforms

This use case illustrates a fundamental use of RIF: to supply a vendor-neutral representation of rules, so that rule-system developers and stakeholders can do their work and make product investments without concern about vendor lock-in, and in particular 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 it is delivered to John more than 10 days after
```

Jane replies with some suggested rule changes:

```
If an item is perishable and it is delivered to John more than 7 days after
```

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 ruleset (or rulesets). 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.

4.1.2 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, access of private medical records, etc., to express and protect their interests within a policy-governed framework. The goal is to formally encode the preferences, priorities, responses, etc., of the parties in such a way that the overall policy can work as intended while providing opportunity for automatic negotiation of terms when allowed by the policy. Utilization of 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, etc., 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 who they trust and for what purposes. The negotiation is based on the policies, which are specified as rules, and credentials Emptor and Venditor have. These policies and credentials are disclosed (interchanged) so as 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 trust that the systems have on each other. Since Emptor and Venditor might use different rule languages and/or engines for evaluating (own and imported) rules, a (standard) rule interchange format (RIF) needs to be employed for enabling the rule interchange between the two systems.

When Alice clicks on a "buy it" button at the 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

Rules express compactly possible ways in which a resource can be accessed; by exchanging them negotiations are shorter and privacy protection is improved. In the example, Venditor reveals part of its policy in form of rules to enable Emptor to choose how to answer, i.e. to decide which credentials and required information to disclose.

For determining whether Venditor's request for information is consistent with Alice's policy, Emptor takes its rules into account, which state for example:

Disclose Alice's credit card information only to online shops belonging to

By disclosing (interchanging) the above given 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 such a credential and its policy contains a rule stating to release it to any potential purchaser. Hence, Venditor passes the credential to Emptor. Emptor is now ready to disclose Alice's credit card information to Venditor but it still must check whether disclosing all the information does not break Alice's denial constraints. Alice has stated two constraints stating:

For anonymity reasons, never provide both her birth date and postal code.

For this purchase, Alice birthdate is no issue. Thus, Alice's constraints are respected. Emptor therefore provides Alice's credit card information to Venditor.

Companies that provide software such as Venditor and Emptor would make use of RIF in a number of ways. The rules expressing Alice's and/or eShop's policies could be expressed in different rule languages but still work with the software, using RIF-based interchanges. Secondly, assuming Venditor and Emptor are products of different companies using different internal rule languages, it would still be possible for them to work together in real-time. When these two systems 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 uses RIF. Emptor takes that policy and translates it from RIF to its internal representation in order to determine what it needs to do.

4.1.3 Interchanging Rule Extensions to OWL

Editor's Note: Example needs to be translated into BLD. Also style needs to be made consistent with other use cases

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 are involved that have 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
```

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.

4.1.4 Vocabulary Mapping for Data Integration

Editor's Note: Example needs to be translated into BLD.

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 whereas 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.

4.1.5 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 (SSN, 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. Use of RIF would improve the situation further. 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 rulesets that are executed sequentially. Rules in the first ruleset 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 ruleset 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 ruleset 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.

4.2 PRD Use Cases

4.2.1 Medical Reasoning Across Integrated Production Systems

4.2.2 Integrating Production Systems in Cases of Business Mergers

4.2.3 Collaborative Policy Development for Dynamic Spectrum Access

Editor's Note: Incomplete: this use case needs to be written in PRD.

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

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

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

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 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; the implementation implications for the various device architectures is generated automatically at the lowest level.