

Linked Rules: Principles for Rule Reuse on the Web ^{*}

Ankesh Khandelwal¹, Ian Jacobi², and Lalana Kagal²

¹ Rensselaer Polytechnic Institute {ankesh}@cs.rpi.edu

² Massachusetts Institute of Technology {jacobi, lkagal}@csail.mit.edu

Abstract. Ontologies are information models which provide vocabulary terms or terminologies and associated meanings to allow the modeling of a domain. They are *shared* conceptualizations; this has never been more true, because in recent years they have been developed by community efforts, often including experts from academia as well as industry. Those efforts have been complemented by the standardization of formats and languages, such as RDF, OWL, and SPARQL, for representing and (re)using ontologies and data on the (Semantic) Web. Rules, on the other hand, are (seldom) used for knowledge representation (i.e. to define the semantics or integrity constraints). Rules are also used for other intelligent reasoning tasks such as for defining business logic and policies. With the prevalence of shared information models, it is possible and may be necessary to share and reuse rules. Furthermore, with the advent of the Rule Interchange Format (RIF), rules can be shared across many rule systems. We propose a set of basic principles and features by which rules can be represented and shared over the web so that they may be effectively reused and demonstrate several methods of rule reuse. Finally, we discuss how some of these features work in practice in the N3-based AIR web rules language.

Keywords: linked rules, online rules, web rules, rule reuse, AIR

1 Introduction

The rise of Semantic Web technologies has enabled the publication, processing and analysis of structured data on the web through the use of standard vocabularies or terminologies (also known as ontologies) which are developed and shared by user communities. Ontologies are used to describe the semantics which facilitate machine readability and data exchange. Although the term ontology may be used to refer to both the terminological knowledge as well as the assertional knowledge (or instance data), we will use *ontology* to refer only to terminologies,

^{*} This paper is based upon work supported by the National Science Foundation under Award No. CNS-0831442, the Air Force Office of Scientific Research under Award No. FA9550-09-1-0152, the Information Technology Alliance in Network and Information Science by U.S. ARL and the U.K. MoD under Agreement No. W911NF-06-3-0001.

and refer to assertional knowledge as *data*, unless specified otherwise. Various languages, such as RDFS [1] and OWL [2], may be used to define ontologies.

Rules may also be used to represent implicit knowledge and integrity constraints that cannot be described by ontology languages such as OWL. Furthermore, rules may be used to declaratively define many intelligent reasoning tasks and computations such as ontology mapping, business logic, and policies. For the remainder of the paper, we will use the phrase *intelligent reasoning tasks* to refer to all possible uses of rules, including the aforementioned uses.

Thus far, rules have not been shared or reused in the same manner as ontologies have over the Semantic Web. Reusing ontologies is encouraged for many reasons, including reducing duplication of effort, sharing common semantics with other ontologies, and reducing maintenance costs by minimizing the number of terms defined for a particular need. We believe that the fundamental reason behind the lack of rule reuse has been a lack of shared information models and terminologies. The wide diversity of the languages supported by rule systems in terms of syntax, features, and semantics may also have contributed. We believe both issues have recently been overcome due to the development of well-defined domain ontologies and the standardization of expressive ontology languages such as RDFS and OWL. Furthermore, the development of RuleML [3] and RIF [4], a W3C standard, has driven the development of interoperability between rule systems/languages.

Much of the growth in Semantic Web data today is based on the principles of Linked Data. These principles [5] describe the recommended best practices for exposing, sharing, and connecting knowledge on the Semantic Web through the use of URIs and RDF. We build upon those principles and propose the following principles for what we call “Linked Rules”:

1. Individual rules and rule-bases should be named using URIs.
2. These URIs should be able to be “dereferenced” to obtain a description in the RIF format.
3. Rules should operate over data expressed in RDF.
4. Rules must be linked.

We will use *linked data* and *linked rules* to refer to data and rules that abide by the principles of Linked Data and those we proposed for Linked Rules, respectively. We also use the word *rules* to refer generally to individual rules, groups of rules, rule-sets, and rule-bases.

We believe that employment of these Linked Rules principles will allow rules to be shared, re-used, combined, and extended on the web in a manner similar to how Linked Data principles have allowed ontologies and data to be shared and reused. We have come to these conclusions as a result of our experience in developing and using the web rules language AIR [6, 7]. Many rules, such as laws, security policies, business rules, or work-flow plans, are rarely defined by a single entity or exist in a single document. They usually comprise a number of rule-bases that are defined and maintained by different entities in a distributed fashion. For example, a university hiring policy may depend on concepts and rules defined by the HR department, the different schools, as well as labs within

each school. Additionally, rules often reference other rules, including those of other organizations. A hospital might want to know with whom a pharmaceutical company would share patient records it provides and might want to reason over the company's rules before transferring patient information. Linked Rules may also be useful in systems that are inherently distributed, such as distributed work-flow systems, supply chain management systems, and collaborations between multiple organizations, as well as in systems that require expertise from several domains such as tax and auditing systems and policy management systems for large organizations.

The paper is structured as follows. In Section 2, we discuss the research and technologies on which Linked Rules depend. The Linked Rules principles are then described in detail in Section 3. In Section 4, we discuss additional metadata and features which may be needed to make Linked Rules practical and useful. In Section 5, we discuss our experience with AIR as a (partial) Linked Rules system. We conclude with a summary in Section 6.

2 Background

A number of recent developments have contributed to the development of Linked Rules principles, the most important of which is perhaps the rise of standardized rule interchange formats (or markup languages).

There are many rule languages possessing widely varying features and expressivities. The recently standardized Rule Interchange Format (RIF) has attempted to provide a mechanism for representing rules belonging to three broad categories: first-order logic, logic-programming, and action rules [4], and we depend on RIF to achieve our goals of Linked Rules. Henceforth, when referring to *rules*, we do not refer to rules in any specific formalism.

RIF is the latest effort to create a standard to exchange rules between many kinds of rule systems. Previous attempts include R2ML (The REVERSE I1 Rule Markup Language) [8] and RuleML. These interchange languages provide normative syntax and well-defined semantics for marking up rules, which alleviates the syntactic differences between languages of rule systems. Furthermore, many rule languages and systems share commonalities in their expressivities and other features, such as built-ins. The interchange languages bring out those commonalities and lead to greater sharing across different languages and systems. Rules are typically marked up in XML¹ using schemas given by the interchange languages, and rule systems define translations, often in XSLT², for translating rule-bases from and to the native representations of other rule systems.

Rule languages may support different representations for facts. Since, RIF's syntactic framework [9] encompasses most rule languages, we will refer to parts of it to discuss the distinctions between the representations of data supported by different rule systems. There are two primary representations of data in rule systems, namely positional terms and frame terms. A positional term is of the

¹ Extensible Markup Language: <http://www.w3.org/XML/>

² Extensible Stylesheet Language Transformations: <http://www.w3.org/TR/xslt20/>

form, $t(t_1 \dots t_n)$, where $n \geq 0$, t is a constant symbol, and t_1, \dots, t_n are constants, variable symbols, list terms, or function terms. Frame terms are used to describe properties of objects, and are of the form $t[p_1 \rightarrow v_1 \dots p_n \rightarrow v_n]$, where $n \geq 0$, t is a variable or constant, p_1, \dots, p_n are constants or variable symbols, and v_1, \dots, v_n , are constants, variable symbols, list terms, or function terms. Another variant of terms referred to as named arguments term, in which arguments are named and their order is immaterial, is of the form $t(a_1 \rightarrow t_1 \dots a_n \rightarrow t_n)$, where a_1, \dots, a_n are (not necessarily distinct) argument names, which are special (constant) symbols (and t, t_1, \dots, t_n are similarly defined as in positional term).

Some of the terms are derived from rules or given as input facts and some terms are defined externally and have a procedure associated with them. The latter set of terms are often called built-ins or procedural attachments. These built-ins may be either functions that compute something and return a value or predicates that verify a condition and return boolean value.

When considering standardizations of the data over which rules should operate, we depend on the Resource Description Framework (RDF), a standard model for data interchange on the web. RDF extends the linking structure of the web by using Unique Resource Identifiers (URIs) [10] to name the relationships between things as well as the two ends of the link (these relationships are usually referred to as triples, $\langle \text{subject property object} \rangle$). The subject of an RDF statement is either a URI or an anonymous “blank” node, both of which denote resources. Blank nodes may not be directly identifiable from the RDF statement. The predicate is a URI which also indicates a resource and represents a relationship. The object is a URI, blank node or a Unicode string literal. Not all URIs are expected to be dereference-able (although, Linked Data principles expect that to be true).

A dereference-able URI is a URI which also serves as a location which may be accessed to acquire a document, such as URIs which begin with ‘http:’ and are accessible via the Hypertext Transfer Protocol (HTTP)³. The dereferencing of a URI involves two steps, as explained by an example from [5]: to look up information on an identifier `http://example.org/smith#carol`: the URI of the document must be obtained by truncating the hash. This URI may then be used to (attempt to) access the document to obtain information about `#carol`.

Reasoning with rules over the web has raised an interesting need for contextualized or scoped reasoning, especially scoped negation as failure [11, 12]. It is practical to scope negation to a dataset than to suggest that something is not true on the web. Axel Polleres et al. give a formal definition of logic programs with context and scoped negation [11]. N3Logic [12] has three special built-ins, `log:includes`, `log:notIncludes`, and `log:supports`, for carrying out scoped pattern matching, where the last considers deductive closure with respect to a set of rules. Other Semantic Web reasoning systems, such as Ontobroker⁴ also support scoping condition matching to certain *modules*. Contextualized reason-

³ HTTP/1.1. <http://www.w3.org/Protocols/rfc2616/rfc2616.html>

⁴ <http://ontobroker.semanticweb.org/>

ing is significant for web reasoning in general because information on the web is often incomplete and correctness is subject to the trustworthiness of the source.

In order to reuse rules, we should be able to use shared information models (or a mapping between two information models) so that different rules may be able to refer to the same concepts. Ontologies are used to describe the semantics of data on the Semantic Web, and as they become established for particular domains, rules in those domains may be reused by referencing such ontologies.

Combining ontologies with rules is not always straightforward, especially when complex languages like OWL are used. That said, there exist several methods of integrating ontologies and rules [13]. Rule languages can be combined with ontology languages in two different ways: hybrid approaches in which rules infer nothing about the ontology predicates, leaving such ontological reasoning to custom reasoners and homogeneous approaches in which both ontologies and rules are embedded in one logical language (e.g. SWRL [14] and ELP [15], which combine OWL sub-languages with a rule language). The principles of Linked Rules are applicable in both scenarios, and thus we do not commit to either approach in this paper. However, we note that if terminologies are based on relatively less expressive languages like RDFS or OWL 2 RL [16], then many of existing rules systems may be able to reason simultaneously with ontologies and rules in a homogeneous approach, as the inference rules for such languages can be expressed as Datalog rules [1, 16] (with some nuances related to datatype inferences).

Ontologies and rules can be reused and extended in some similar ways. Ontologies can be extended with new vocabulary for application specific needs. Such extensions may be defined by relating new classes and properties to existing ontologies, through the `rdfs:subClassOf` and `rdfs:subPropertyOf` relationships, and specializing them with restrictions (e.g. cardinality restrictions and domain and range restrictions). We believe that rules can be specialized and reused in some similar ways, which we explore in Section 4.2.

3 Linked Rules

Ontologies are used for assigning semantics to and exchanging of information over the Semantic Web. As ontologies have become defined for many domains, it has become more realistic to expect that rules which make use of data expressed using such ontologies should be shared and reused.

The paradigm of Linked Rules provides for such reuse through a set of language/system-independent principles for defining rules and sharing them on the web. As mentioned, we propose the following four principles of Linked Rules: individual rules and rule-bases should be named using URIs, such URIs should be able to be dereferenced to obtain a description in RIF, rules should operate over data expressed in RDF, and rules must be linked. These principles mirror the four principles of Linked Data, namely using URIs as names for things, using HTTP URIs so that people can look up those names, provide useful information at those URIs using web standards like RDF, and including links to other URIs, so that users can discover more things [5].

3.1 Naming Rules

Providing URIs for rules allows consumers of rules to refer to and identify individual rules and rule-bases by citing only the URI of the rule or rule-base. Should such URIs be dereference-able (e.g. HTTP URIs), then users may be able to obtain a copy of the rule by dereferencing the URI and obtain more information about the rule. (This principle is similar to the fourth essential (E4) for online rules proposed by Harold Boley [17].)

By providing URIs for online rule-bases, these rule-bases may be used as input for reasoners by providing their URIs. They may also be used to construct and extend new rule-bases by referring to pre-existing rule-bases to be extended and included through their URIs. In addition, the use of URIs allows for meta-data, such as authorship, to be attached to rule-bases through reference to their URIs in RDF graphs.

URIs for individual rules may be used to define priorities between rules combined from multiple sources, or to refer to specific rules in the justifications and proofs of inferred facts. They may also be used to construct a new rule by specializing an existing rule, as referred to by its URI. We give an example of such construction in Section 4.2, under the heading of Constrained Reuse.

3.2 RIF: A Standard Format for Rules

When dereference-able URIs of rules or rule-bases are dereferenced, the descriptions (or definitions) of the rule or rule-base should be returned. We propose returning such rules in a standard general-purpose rule format, such as RuleML or RIF, as they not only alleviate syntactic differences between various rule languages/systems, but also manifest common expressivities and features between such, contributing to wider rule sharing.

We suggest the use of RIF for this purpose as RIF is a W3C standard specifying a mechanism for compatibility with RDF [18] as part of the standard, which meshes well with the third Linked Rules principle. RIF also provides an infrastructure of standard datatypes and built-in functions through RIF-DTB [19], which is also part of the standard. Finally, there is a working draft on representing RIF documents in RDF [20]. Such a specification would allow rules to be interchanged in RDF, while RIF semantics will provide the rule interpretation. Representation of rules in RDF will allow rules to be treated as data themselves and reasoned over by other rules⁵.

RIF currently has two main dialects, the Production Rule Dialect (RIF-PRD) [21] and the Basic Logic Dialect (RIF-BLD) [22]. RIF-BLD is of limited expressivity; for example, RIF-BLD does not support negation. However, most

⁵ This is one of the design principles of N3: “information, such as but not limited to rules, which requires greater expressive power than the RDF graph, should be sharable in the same way as RDF can be shared. This means that one person should be able to express knowledge in N3 for a certain purpose, and later independently someone else reuse that knowledge for a different unforeseen purpose” [12].

logic dialects can be specified using the Framework for Logic Dialects (RIF-FLD) [9], which is part of the standard. For example, Answer Set Programming and Default Logic dialects have been specified as specialization of RIF-FLD⁶.

3.3 RDF: A Standard Format for Data

Benjamin Grosf and Harold Boley have suggested that web rules must share mechanisms with other emerging or existing standards for Knowledge Representation (KR) and ontologies on the web, including RDF and OWL [23, 17]. We believe that, although individual web rules systems may support alternative data formats, Linked Rules should be defined to operate over RDF, as RDF is a W3C standard for data interchange on the (Semantic) Web.

While RIF offers uniform basic constructs for the representation of data as positional terms, named argument terms, and frame terms, many systems support only some of them and some support none.⁷ At the syntactic and data format levels, the choice of basic constructs must not affect re-usability of rules.

Note that RDF facts can be expressed either as frames, as in RIF [18]⁸ or positional terms, as in OWL 2 RL (inference rules) [16]. Furthermore, frames can be conveniently mapped to RDF in many circumstances [24]. Support for RDF also means that linked rules may make use of ontologies which are likely to be designed using RDF-based languages like RDFS and OWL.

Admittedly, in the past, concerns have been raised about using an RDF data model with rules. For example, it is cumbersome to represent general n-ary relations and sequences in RDF, although [25] describes a couple of ways of representing n-ary relations on the Semantic Web. For example, an n-ary relation can be captured as an OWL concept (class) with n properties. Many proposals have been put forth during the 2010 W3C RDF Next Steps workshop to extend the RDF specifications to address some other concerns. Some of these proposals, including standardization of the Turtle syntax [26], skolemization of blank nodes, and graph identification, have been taken up by the RDF working group⁹. Furthermore, collections and lists can be conveniently represented in the Turtle syntax as lists of items separated by white space enclosed in parentheses¹⁰.

3.4 Linking Rules

Ultimately, rules must be linked to something. Rules may be linked to other rules that they extend and reuse. We discuss several different ways to reuse rules in

⁶ RIF-CASPD. <http://ruleml.org/rif/RIF-CASPD.html> and Default Logic Dialect. <http://silk.semwebcentral.org/RIF-SILK.html>

⁷ e.g. N3 and user defined rules in Jena support none of them. Allegrograph supports prolog based reasoning, but cannot accept frame terms.

⁸ The blank nodes are interpreted as only local symbols in [24], the same interpretation may be made for linked rules as well.

⁹ RDF Next Steps Report. <http://www.w3.org/2009/12/rdf-ws/Report.html>

¹⁰ Here, Turtle syntax only helps with compact representation of lists and does not offer collections/lists as a datatype.

Section 4.2. Rules may also be linked to the ontologies which provide terminology and data used in the rules. They may also be linked to other meta-data such as authorship or rule dialect as described later in Section 4.1. Additionally, rules may be linked with the outputs of the reasoning they are a part of, as well as the proofs or justifications for inferred RDF triples.

4 Beyond the Basics

Although we propose only four basic Linked Rules principles necessary for linked rules to exist, we believe that there are a number of additional practices which will better facilitate rule reuse within linked rules. Proper rule meta-data, rule discovery, and mechanisms for rule reuse must all be considered in order for rules to be easily reused.

4.1 Describing Rules

We believe that rules should be described sufficiently well so that they can be discovered and evaluated for fitness for (re)use. Users of rules will likely want to know why rules have been defined, who authored them, what terminologies and data they use or require¹¹, rule complexity, and so on. We believe that specifying the terminologies used by the rules as well as the dialect and complexity used to define the rules¹², is the minimum amount of meta-data needed for linked rules, as they are generally always known at rule construction time, and they are likely to be needed in order to evaluate fitness of rules for reuse in most cases. With this minimum amount of meta-data, users can identify if the data model assumed by the rules matches the one being used by the consumer. Furthermore, users will be able to determine if such rules can be handled by the reasoners accessible to them.

Additional meta-data, such as the differences between the ontologies used for input and inferred facts may also be very useful. Rules are defined in some language or dialect which provides the required expressive features but not all the features are always used. Descriptions about the expressive features that have been used by a rule and the restrictions that rules obey may also be useful. (Benjamin Grosf has suggested some of the above meta-data for webizing KR [23].)

A vocabulary in the vein of the Vocabulary of Interlinked Datasets (VoID)¹³ for expressing such rule meta-data may help in achieving uniform annotation of rules, thus contributing to better sharing and reuse of rules.

¹¹ e.g. OWL 2 RL inference rules which operate over OWL 2 RL ontology, use terminologies from OWL meta-modeling ontology, <http://www.w3.org/2002/07/owl#>.

¹² Note: The *Dialect* directive can be used to declare the dialect in RIF documents [9].

¹³ <http://www.w3.org/TR/void/>

4.2 Mechanisms for Rule Reuse

One of the motivations behind Linked Rules is to make rule reuse possible, so that rules for similar intelligent reasoning tasks need not be recreated from scratch. We will describe four ways in which rules could be reused, namely **simple reuse**, **rule importation**, **contextualized reuse**, and **constrained reuse**.

Simple reuse is the simplest case of reuse in which pre-defined rule-bases may be used with datasets of interest to deduce implicit information, map from one vocabulary to another, or verify if the information satisfies a rule-based policy, and so on. Online reasoners, like the AIR reasoner, can take external requests to perform simple reuse with input datasets and rule-bases.

Rule importation is the style of rule-set inclusion supported in RIF (and OWL¹⁴). Rules imported into another rule-set are unmodified, simply merged with the other rules defined in the document, allowing for modular rule-set development. Care must be taken when importing rules in this way, as rule languages may have different semantics and complexities (and dialects). Use of RIF only ensures a consistent syntactic representation and semantic conformance of rules; reasoners may not be able to cope with the computational complexity of some dialects.

Rule importation can be used to combine rules for different reasoning tasks from different sources. These rules may then be extended with new rules for application-specific needs. For example, if the combined form of several rules is expressible in courteous logic [23], they may be extended with priorities between rules from different sources.

Sometimes some datasets and rules are needed only to ascertain truth of only a few conditions, and are not necessary to take part in the complete reasoning process. It may also be beneficial to avoid mixing two reasoning spaces or to check whether a condition is satisfied by a specific dataset using certain rules. In all these cases *contextualized reuse* is relevant, so that the rules may be reused only in certain *contexts*.

A context is typically defined by some datasets and/or rule-sets, and may be specified by either explicit or implicit (through variables) references to sources satisfying certain criteria. In Section 5.2 we show an example of contextualized reuse, where an online retailer verifies all intended transactions of users' online behaviors with the policy of the user involved. In the example, a specific user's rules are reused within a context to ascertain compliance with the intended transaction. Note that we use the term context in a sense different from the quoted formulas in Notation 3 (N3) [12] or meta-statements in F-Logic [27].

Constrained reuse of a rule occurs when additional constraints are used to further restrict the possible bindings of the variables occurring in the rule.¹⁵ The resulting rule can be considered to be a derivation or specialization of the original rule. The added constraints can be used to specify more explicit assumptions which were not considered necessary in the original spec-

¹⁴ Here, similarity in import semantics (OWL doesn't have rules).

¹⁵ Note. This reuse is intended for rules of form, *If conditions then conclusions/actions*.

ification or to restrict the application of a rule to a limited set of items.¹⁶ For example, consider one form of rule definition supported by RIF-PRD, *Forall ?v1...?vn such that (p1...pm) (rule)*, where $p1 \dots pm$ are conditions [21]. This form can be used for constrained reuse; $p1 \dots pm$ can be new conditions used to restrict the bindings of certain variables appearing in the predefined *rule*. Obviously, the same symbols should be used to refer to the variables.

4.3 Using SPARQL for Checking Condition against Linked Data

If web rules are meant to check conditions against the linked data, such as checking hierarchical geographical relationships in Geonames¹⁷ dataset, rule systems (and languages) may benefit from supporting SPARQL based condition checking. Copying millions of RDF triples to a local knowledge base may negatively affect the performance of many reasoning systems, especially in-memory reasoners. However, as many of these datasets are available via SPARQL endpoints¹⁸, and others can be loaded to RDF triple stores, it may be possible to match the simple conditions against online RDF databases using SPARQL queries (and express such matching within the rule definition).¹⁹ Note that this is different from using SPARQL construct queries to define rules proposed in [28, 29].

4.4 Representing Linked Rules Features in RIF

In previous sections, we have discussed some features from which web rules could benefit, namely contextualized and constrained reuse of rules and SPARQL-query-based condition matching against online RDF datasets. We *recommend* that Linked Rules languages and systems support them. Since RIF is expected to be the interchange format for Linked Rules, we investigate the expressibility of those features in RIF.

New RIF built-in functions and predicates may be defined as externally-defined terms in RIF-FLD [19], but are not expected to be standards. Built-ins can be introduced to support SPARQL-query-based condition matching against SPARQL endpoints as well as for contextualized reuse. Contextualized reuse entails query answering against datasets and the facts inferred by some rules²⁰.

Remote terms may be used to query remote RIF documents, or remote modules, in RIF-FLD [9]. A remote term is of the form $\phi@r$ where ϕ is a singular condition and r is a variable, constant or parameterized reference to a single RIF module. We conjecture that the remote terms can be extended to allow r

¹⁶ e.g. a rule may be initially designed for all students, but it can be restricted to graduate students before reuse for application driven needs.

¹⁷ <http://www.geonames.org/ontology/documentation.html>

¹⁸ Definition of endpoints from: Web Services Description Requirements. W3C Working Draft 28 October 2002. <http://www.w3.org/TR/ws-desc-reqs/#normDefs>

¹⁹ Online condition matching can be slow, but may be acceptable if it outweighs any reduction in performance caused by having local copies of the data.

²⁰ In some dialects, query answering semantics may have to be defined (e.g. when query answering is undecidable or the response is trivalent (true, false, undefined)).

to be a list term, so that multiple modules can be specified. The interpretation of documents [9], then, will have to be extended so that the semantic multi-structures include semantic structures of all possible unions of combinations of remote modules, in addition to the semantic structures for each remote module²¹. However, in general contextualized reasoning datasets and/or rule-sets may be dynamically determined and not known prior to reasoning. Such modules cannot be specified by *Module* directives.

In Section 4.2, we gave an example of rule definition through constrained reuse of another rule in RIF-PRD. However, constrained rules cannot be defined using well-formed formulas (WFFs) in RIF-FLD [9]. RIF-FLD WFFs may be extended with RIF-PRD-like rule definitions, but this is insufficient. Even in RIF-PRD, a rule cannot be referenced just by its URI; the rule definition must contain some (un)conditional action block. RIF WFFs, in general, must be extended so that part definitions can be given by URI references to pre-defined descriptions. We discuss how a complete rule definition may be constructed from its distributed parts in the next section.

4.5 Compiling Linked Rules

Linked Rules may link to a number of different things, including other rule-bases, individual rules, and, potentially, parts of rules. During rule compilation, all URI references for rule parts must be replaced by their descriptions, obtained by dereferencing the URI of the rule part and extracting any relevant portion of the dereferenced document. Note that this is different from dereferencing a URI and including all of its contents, as may be done for rule importation.

5 AIR and Linked Rules

The aforementioned Linked Rules principles and recommended features have developed from our experience in developing the web rule language, AIR. As such, we will discuss the features of AIR in terms of a Linked Rules language.

5.1 Brief Overview of AIR

AIR is a Semantic Web rules system using N3 syntax for expressing rules. N3 extends RDF with the ability to use graphs as literal values, universal or existential quantification of variables in a graph, built-in functions and operators represented as RDF properties, and symmetrical use of literal values. Rules are specified as a `log:implies` relationship between graph patterns²², where `log:implies` is a built-in function.

²¹ e.g. if two modules are referenced by *Module* directive (and none by *Import* directive), the multi-structure by current semantics $\{J, K, M^{j_1}, M^{j_2}\}$ (where J and K are the RIF-FLD semantic structures) will be extended to $\{J, K, M^{j_1}, M^{j_2}, M^{j_{(1,2)}}\}$

²² Like Basic Graph Patterns in SPARQL, but built-in expressions can be included.

AIR extends N3 by providing a method of expressing rules such that rules are first class entities; every rule is an instance of *Belief-rule* concept²³, and defined as a production rule:

ruleid **if** {*graph-pattern*}; **then** <*actions*>; **else** <*actions*> ,

where *ruleid* is a rule identifier. An action may either activate another rule (*:actionid* **rule** *:ruleid*) or assert some facts (*:actionid* **assert** {*graph-pattern*}). Just as each rule has an identifier, each action has an identifier, *actionid*.

AIR rules may be grouped into rule-sets, and the member rules are declared as: *:rs-id* **rule** *:ruleid*, where *rs-id* is the identifier of an instance of the *Rule-set* concept.

AIR also extends the set of built-ins supported in N3. For instance, *sparql:queryEndpoints* and *air:justifies* have been added to AIR to support SPARQL queries and contextualized reasoning.

A typical usage of *sparql:queryEndpoints* has the following form:

<*sparql-endpoint*> *:CST-QUERY* *sparql:queryEndpoint* *:RESULTS* ,

where *sparql-endpoint* is the URI of some SPARQL endpoint, *:CST-QUERY* is a SPARQL CONSTRUCT query string and *:RESULTS* is a variable which binds to the result triples returned by the endpoint at the time of reasoning. Note that the end-point and the query string are specified in a sequential list. Figure 1 shows an example of an AIR rule iPRn Turtle syntax where the *air:justifies* built-in is used.

The semantics of the AIR language is defined in [6]²⁴. Unlike other production rule systems neither retraction nor execution of external functions are supported in AIR. The declarative semantics of AIR without the built-ins can therefore be defined in terms of a minimal model of an equivalent logic program.

5.2 Linked Rules Principles and Features in AIR

Every rule and rule-base in an AIR rule-set has a URI (or a blank node) which identifies it. A translation of AIR to RIF-PRD has been defined [30] but when URIs are dereferenced, we currently expect that only AIR documents will be obtained. Other reasoners could plausibly expect RIF to be returned in order to achieve compatibility with the AIR system. Once the AIR to RIF-PRD translation is implemented, URIs may be dereferenced to get RIF documents. AIR supports use of the N3 data format, and thus also supports RDF data format. Furthermore, since rules are first-class entities, they can be freely annotated (e.g. they may be assigned *rdfs:labels*). Moreover, new rule-bases can reuse pre-defined rules. The AIR reasoner is responsible for fetching the definition of a rule, if not available in the same document. Similarly, actions can activate rules not defined in the same document. Therefore, AIR currently supports three of the four Linked Rules principles.

AIR supports built-ins which access remote triple stores. With respect to rule reuse, like many other rule systems, AIR supports simple reuse and rule

²³ AIR's rule specification ontology- <http://dig.csail.mit.edu/TAMI/2007/amord/air>

²⁴ Online version with more details: <http://tw.rpi.edu/proj/tami/AIR.Language.Formalization>

```

In BigBook's Rules File:

@forAll :TXN, :DATA, :CUSTOMER, :RECIPIENT, :POLICY, :RULES . #variables
@prefix : <http://bigbooks.example.com/rules> .

:BigBooksDissemination a air:Rule-set;
  air:rule :UserConsent, ... # only UserConsent rule is described here

:UserConsent a air:Belief-rule ;
  air:if {
    :TXN a :Transaction ; :recipient :RECIPIENT ;
      :data :DATA . # document containing usage behavior data of :CUSTOMER
    :DATA :about :CUSTOMER .
    :CUSTOMER a :Customer ;
      :policy :POLICY . # customer's policy document
    :POLICY log:semantics :RULES . # retrieve rules (N3 graph) from document
    (({:TXN a :Transaction ; :data :DATA ; :recipient :RECIPIENT ;
      :sender <http://bigbooks.example.com> .
    }) (:RULES)) air:justifies :CONCLUSIONS .
    :CONCLUSIONS log:includes { :TXN air:compliant-with :POLICY . } .
  } ;
  air:then [ air:assert { :TXN air:compliant-with :UserConsent . } ] ;
  air:else [ air:assert { :TXN air:non-compliant-with :UserConsent . } ] ] .

```

Fig. 1. Contextualized Reuse example: In order to determine a customer's consent, the transaction information is reasoned independently with the concerned CUSTOMER's :RULES, dereferenced from its :POLICY document, by BigBooks. BigBooks uses the air:justifies built-in in the UserConsent rule for scoped reasoning.

importation. Furthermore, contextualized reasoning is supported through use of the air:justifies built-in. An example of such use is given in Figure 1, in which BigBooks, an online book retailer, intends to determine whether it will share collected usage behavior of its customers based on customer preferences.

If it is assumed that customers publish their preferences as a policy, maintained online (and known to BigBooks), defined using AIR rules to infer either compliance or non-compliance with the policy, BigBooks can then define the UserConsent rule to collect information of individual transactions and reason over them independently with concerned customer's RULES.

Input to the air:justifies built-in is a list of two lists, where the first is a list of input triples while the second is a list of rules. Note that multiple transactions, involving multiple customers, can be reasoned in one go without any interference.

In addition, AIR, also allows the activation of another rule as one of its actions; not all rules are considered to be active in the beginning. Any active variable bindings are passed on to a newly activated rule. When a rule is activated, variables with active bindings are substituted by their binding values.

This aspect of AIR, namely conditional activation of rules, helps in conditional reuse of rules. To illustrate, we can restrict BigBooks dissemination rules to be used only for valued customers (assuming that all ValuedCustomers are also Customers), by creating a new rule ValuedUserConsent to replace the UserConsent rule in the BigBooksDissemination rule-set, as follows:

```

:ValuedUserConsent a air:Belief-rule ;
  air:if { :CUSTOMER a :ValuedCustomer . } ;
  air:then [ air:rule :UserConsent ] .

```

6 Summary

We proposed a set of principles for enabling the sharing of rules on the web so that they may be combined, re-used, and extended more effectively. In particular, individual rules and rule-bases should be named with dereference-able URIs, the rules must be defined to operate over data in RDF format and interchanged using RIF, and they should be linked to other ontologies, rules and meta information. Additionally, we recommended features that would help in effective re-use of rules (contextualized and constrained re-use) and effective use of linked data for simple condition matching (via SPARQL queries). Some of these features are expressible in RIF, and for others we discussed possible extensions to RIF, which we will continue to explore. Finally, we explored AIR as a Linked Rules system.

There are some practical considerations concerning rule reuse that must be investigated and we outline some of them. Rule systems may support partial selection and reuse of rules on the web, similar to how Prolog provides for importing rules for certain predicates, for example. When rules depend on each other, changes to a dependee should be accommodated. As circular dependencies are possible, they must be avoided or resolved. Rule reuse will be trust-based and mechanisms for establishing trust will be crucial. Finally, better negotiation of differences between diverse rule systems can contribute to wider rule reuse. It would be worth exploring whether similarities between systems (beyond that captured by RIF representation) can be enumerated, or whether common interoperable semantics can be defined pair-wise for systems.

Acknowledgements. We thank Professor Harold Boley for insightful comments on the paper, and Jesse Weaver for reviewing the paper.

References

1. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. <http://www.w3.org/TR/rdf-schema/>
2. W3C: OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009. <http://www.w3.org/TR/owl2-overview/>
3. Boley, H., Paschke, A., Shafiq, O.: RuleML 1.0: The Overarching Specification of Web Rules. In: RuleML. (2010)
4. Kifer, M., Boley, H.: RIF Overview. W3C Working Group Note 22 June 2010. <http://www.w3.org/TR/rif-overview/>
5. Berners-Lee, T.: Linked Data-Design Issues. <http://www.w3.org/DesignIssues/LinkedData.html> (2006)
6. Khandelwal, A., Bao, J., Kagal, L., Jacobi, I., Ding, L., Hendler, J.: Analyzing the AIR Language: A Semantic Web (Production) Rule Language. In: Web Reasoning and Rule Systems (RR). (2010)
7. Waterman, K.K., Wang, S.: Prototyping Fusion Center Information Sharing; Implementing Policy Reasoning Over Cross-Jurisdictional Data Transactions Occurring in a Decentralized Environment. In: IEEE Conference on Homeland Security Technologies (IEEE HST). (2010)

8. REVERSE: R2ML - The REVERSE I1 Rule Markup Language. <http://oxygen.informatik.tu-cottbus.de/reverse-i1/?q=node/6>
9. Boley, H., Kifer, M.: RIF Framework for Logic Dialects. W3C Recommendation 22 June 2010. <http://www.w3.org/TR/rif-fld/>
10. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax. Internet Draft Standard RFC 2396, August 1998; <http://www.isi.edu/in-notes/rfc2396.txt>
11. Polleres, A., Feier, C., Harth, A.: Rules with Contextually Scoped Negation. In: Proceedings of 3rd European Semantic Web Conference (ESWC). (2006)
12. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A Logical Framework For the World Wide Web. In: J. of Theory and Practice of Logic Programming. (2007)
13. Antoniou, G., Damasio, C.V., Grosf, B., Horrocks, I., Kifer, M., Maluszynski, J., Patel-Schneider, P.F.: Combining Rules and Ontologies: A survey. In: Technical Report, Project Deliverable I3-D3, REVERSE. (2005)
14. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosf, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. <http://www.w3.org/Submission/SWRL/> (2004)
15. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable Rules for OWL 2. In: Proc. of the 7th International Conference on The Semantic Web. (2008)
16. W3C: OWL 2 RL. W3C Recommendation 27 October 2009. http://www.w3.org/TR/owl2-profiles/#OWL_2_RL
17. Boley, H.: Are Your Rules Online? Four Web Rule Essentials. In: RuleML. (2007)
18. de Bruijn, J.: RIF RDF and OWL Compatibility. W3C Recommendation 22 June 2010. <http://www.w3.org/TR/rif-rdf-owl/>
19. Polleres, A., Boley, H., Kifer, M.: RIF Datatypes and Built-Ins 1.0. W3C Recommendation 22 June 2010. <http://www.w3.org/TR/rif-dtb/>
20. Hawke, S.: RIF In RDF. W3C Working Draft 22 June 2010. <http://www.w3.org/TR/rif-in-rdf/>
21. de Sainte Marie, C., Hallmark, G., Paschke, A.: RIF Production Rule Dialect. W3C Recommendation 22 June 2010. <http://www.w3.org/TR/rif-prd/>
22. Boley, H., Kifer, M.: RIF Basic Logic Dialect. W3C Recommendation 22 June 2010. <http://www.w3.org/TR/rif-bld/>
23. Grosf, B.N.: Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML. In: J. of Electronic Commerce Research & Applications. (2003)
24. Lassila, O., McGuinness, D.L.: The Role of Frame-Based Representation on the Semantic Web. In: J. of Electronic Transactions on Artificial Intelligence. (2001)
25. Noy, N., Rector, A.: Defining N-ary Relations on the Semantic Web. W3C Working Group Note 12 April 2006. <http://www.w3.org/TR/swbp-n-aryRelations/>
26. Beckett, D., Berners-Lee, T.: Turtle - Terse RDF Triple Language. W3C Team Submission 28 March 2011. <http://www.w3.org/TeamSubmission/turtle/>
27. Yang, G., Kifer, M.: Reasoning about Anonymous Resources and Meta Statements on the Semantic Web. In: Data Semantics I. (2003)
28. Boley, H., Kifer, M., Ptnjan, P.L., Polleres, A.: Rule Interchange on the Web. In: Reasoning Web. (2007)
29. Schenk, S., Staab, S.: Networked Graphs: a Declarative Mechanism for SPARQL Rules, SPARQL views and RDF Data Integration on the Web. In: Proc. of the 17th international conference on World Wide Web (WWW). (2008)
30. Khandelwal, A.: AIR to RIF-PRD. http://tw.rpi.edu/proj/tami/AIR_to_RIF-PRD (2011)