# Rule Interchange on the Web

Paula-Lavinia Pătrânjan (REWERSE)
Axel Polleres (DERIGalway)
Sandro Hawke (W3C Team)

EBRC 2007 - June 20, 2007

# Outline

Rule Interchange

- Motivation
- Current Efforts
- Rule Types

W3C RIF WG Work

- Charter
- Framework – The Web
- Current Status
- RIF Core

Towards RIF Dialects for PR and ECA Rules

Conclusion

# Motivation

The rule-based programming paradigm offers a

- flexible and adaptive approach towards application development
- high level means for deploying applications of various domains

For exploiting further the potential of the rule-based approach

- the business rules and Semantic Web communities
- started to develop solutions
  - reuse and integrate knowledge
  - specified in different rule languages

# Current Efforts

Efforts such as . . .

- Rule Markup Initiative - RuleML
- OMG - PRR and SBVR
- REWERSE - R2ML
- W3C Member Submissions - SWRL, WRL, SWSL Rules

. . . led to the W3C Rule Interchange Format Working Group (RIF WG)

- 77 participants from industry and academia
- chaired by representatives of IBM and ILOG
- chartered to standardize a common format for interchanging rules
  - which is not a trivial task!

# Rule Types

PR vendors, database systems vendors, and Semantic Web researchers have different views on the notion of *rules*:

- *derivation rules*
  - derive knowledge by means of logical inference
- *dynamic rules*
  - automatically execute actions when events occur and/or conditions become true
  - for example
    - Production rules (PR)
    - Event-Condition-Action (ECA) rules
- *structural rules*
  - pose constraints on the data and the logic of applications

... and these rule types raise different requirements on an interchange format

# Rule Types

*Example (derivation) rule*

IF movie ?M was produced before 1930
THEN ?M is a black and white movie

- IF-part
  - specifies a **condition** for retrieving data on movies
  - binds the variable ?M to data items
- THEN-part
  - constructs new data by using the retrieved bindings
  - a view over movies data

# Rule Types

*Example (dynamic) rule*

ON request from customer ?C to book a movie
IF customer ?C is blacklisted
DO deny ?C's request

- ON-part waits for a request for a movie to come in (an **event**)
- IF-part checks a **condition** on the customer's data
- DO-part
  - specifies the **action** to be executed
  - on a request from a blacklisted customer

# Rule Types

*Example (structural) rule*

Each movie must have a single production year.

- specifies a **condition** which must not be violated by the data
- two different production years for the same movie is an indication of corrupted data

- derivation and dynamic rules can be used to implement structural rules
- the decision depends on the application and the available support for rules

# Rule Types

Condition part is common to all possible rule “dialects”, so

- let’s start with developing a format for interchanging rule conditions
- and then extend it!

# Charter

Phase I

- simple, but extensible interchange format for rules (RIF Core)
- Dec 2005 - Nov 2007

Phase II

- extensions in form of RIF Dialects (e.g. FOL, PR)
- until June 2008

Emphasizes compatibility with

- Web technologies - XML
- Semantic Web technologies - RDF, OWL, SPARQL

# The Web as Framework for Rule Interchange

- The Web is a success story in terms of **linking** data (HTML)
- Web formats, such as XML have made it to nowadays standard formats for also non-Web **data exchange**
- The next generation of the Web will allow to link and exchange data (RDF) and its **structure** (RDF Schema, OWL) even more flexible
  → this is often called the **Semantic** Web
- An important side-effect, the Semantic Web will also allow to **exchange rules**!

ie.: The Semantic Web is about exchange of Data, Data/Domain Models and (by RIF) Rules!

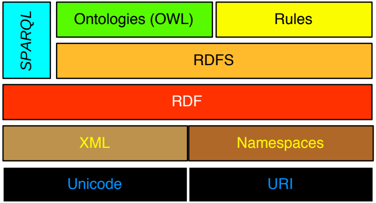
Let us talk about these foundations a bit, since they have some implications for RIF!

# The Web as Framework for Rule Interchange

- The Web is a success story in terms of **linking** data (HTML)
- Web formats, such as XML have made it to nowadays standard formats for also non-Web **data exchange**
- The next generation of the Web will allow to link and exchange data (RDF) and its **structure** (RDF Schema, OWL) even more flexible
  → this is often called the **Semantic** Web
- An important side-effect, the Semantic Web will also allow to **exchange rules**!

ie.: The Semantic Web is about exchange of Data, Data/Domain Models and (by RIF) Rules!

Let us talk about these foundations a bit, since they have some implications for RIF!



The (Semantic) Web architecture stack

# Semantic Web architecture 1/5: XML

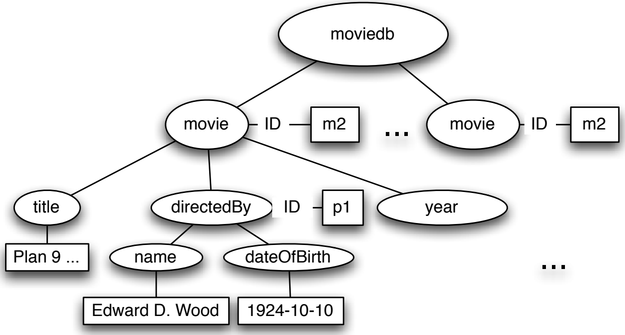
```
<?xml version="1.0"  encoding="UTF-8"?>
<moviedb xmlns="http://imd.example.org/ns/">
  <movie ID="m1">
    <title>Plan 9 from Outer Space</title>
    <directedBy ID="p1">
      <name>Edward D. Wood Jr.</name>
      <dateOfBirth>1924-10-10</dateOfBirth>
    </directedBy>
    ...
    <year>1959</year>
  </movie>

  ...
</moviedb>
```

- Tree to handle semi-structured data
- Unique identifiers to disambiguate formats (namespaces)
- Facilitates data exchange on a syntactical level
- Take-up in many applications which need common formats (ebXML, Web Services)

⇒ RIF will also have an XML syntax!

# Semantic Web architecture 1/5: XML



- Tree to handle semi-structured data
- Unique identifiers to disambiguate formats (namespaces)
- Facilitates data exchange on a syntactical level
- Take-up in many applications which need common formats (ebXML, Web Services)

⇒ RIF will also have an XML syntax!

# Semantic Web architecture 2/5: RDF

- Integrating **different** XML formats is still sometimes tricky (XSLT), due to the tree format of XML.
- The data model of the Semantic Web is **graphs** instead of trees.

An RDF graph is made up by a set of **statements** about **resources**:

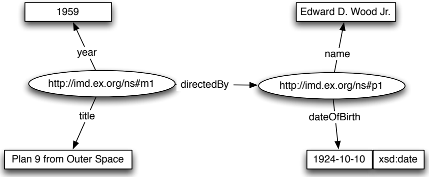
```
<http://imd.ex.org/ns#m1> title       "Plan 9 from Outer Space" .
<http://imd.ex.org/ns#m1> directedBy  <http://imd.ex.org/ns#p1> .
<http://imd.ex.org/ns#m1> year "1959"


<http://imd.example.org/ns#p1> name        "Edward D. Wood Jr." .
<http://imd.example.org/ns#p1> dateOfBirth "1924-10-10"^^xsd:date .
   ...
```

# Semantic Web architecture 2/5: RDF

- Integrating **different** XML formats is still sometimes tricky (XSLT), due to the tree format of XML.
- The data model of the Semantic Web is **graphs** instead of trees.

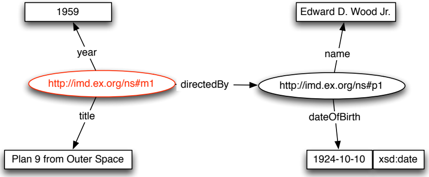
Sets of RDF statements may be viewed as directed Graphs:

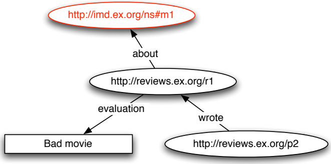


# Semantic Web architecture 2/5: RDF

- Integrating **different** XML formats is still sometimes tricky (XSLT), due to the tree format of XML.
- The data model of the Semantic Web is **graphs** instead of trees.

Sets of RDF statements may be viewed as directed Graphs:

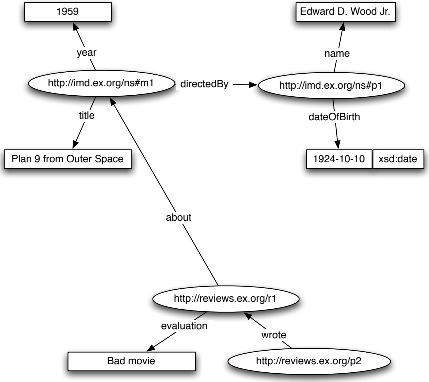




# Semantic Web architecture 2/5: RDF

- Integrating **different** XML formats is still sometimes tricky (XSLT), due to the tree format of XML.
- The data model of the Semantic Web is **graphs** instead of trees.

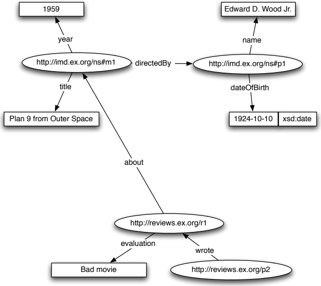
Sets of RDF statements may be viewed as directed Graphs:



The flat data model of RDF is **easier to integrate** than XML!

# Semantic Web architecture 3/5: RDFS/OWL

- besides industry used data formats like UML, the Semantic Web architecture has defined more flexible ways to **exchange and integrate** not only data, but also **data/domain models**:
- RDFS (= RDF Schema) OWL (= Web Ontology Language)
- allows to add classes and types to RDF
- allow to express subclass hierarchies, subproperty hierarchies, etc.



# Semantic Web architecture 3/5: RDFS/OWL

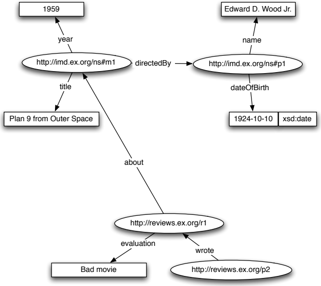
- besides industry used data formats like UML, the Semantic Web architecture has defined more flexible ways to **exchange and integrate** not only data, but also **data/domain models**:
- RDFS (= RDF Schema) OWL (= Web Ontology Language)
- allows to add classes and types to RDF
- allow to express subclass hierarchies, subproperty hierarchies, etc.

OWL and RDFS can express additional relations among types and properties, e.g.:

- *each Director is a Person (subclass)*
- *each Reviewer is a Person (subclass)*
- *somebody who directed a Movie is a Director (range restriction)*
- *somebody who wrote a Review is a Reviewer (domain restriction)*
- etc.

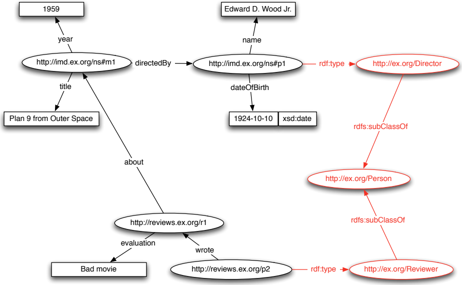
# Semantic Web architecture 3/5: RDFS/OWL

- besides industry used data formats like UML, the Semantic Web architecture has defined more flexible ways to **exchange and integrate** not only data, but also **data/domain models**:
- RDFS (= RDF Schema) OWL (= Web Ontology Language)
- allows to add classes and types to RDF
- allow to express subclass hierarchies, subproperty hierarchies, etc.



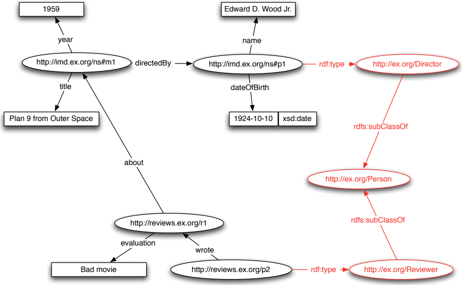
# Semantic Web architecture 3/5: RDFS/OWL

- besides industry used data formats like UML, the Semantic Web architecture has defined more flexible ways to **exchange and integrate** not only data, but also **data/domain models**:
- RDFS (= RDF Schema) OWL (= Web Ontology Language)
- allows to add classes and types to RDF
- allow to express subclass hierarchies, subproperty hierarchies, etc.



# Semantic Web architecture 3/5: RDFS/OWL

- besides industry used data formats like UML, the Semantic Web architecture has defined more flexible ways to **exchange and integrate** not only data, but also **data/domain models**:
- RDFS (= RDF Schema) OWL (= Web Ontology Language)
- allows to add classes and types to RDF
- allow to express subclass hierarchies, subproperty hierarchies, etc.



The real power of common domain models reveals in sharing and reusing them!

# Semantic Web architecture 4/5: XML vs. RDF(S)+OWL

| | XML | RDF |
|---|---|---|
| Data Model: | Tree | Graph |
| Identifiers: | element, attribute names | everything identified by URIs |
| Data: | in the leaves | in the nodes |
| Relations | in the nodes | in the edges |
| Data structure : | XML Schema<br>(syntax) | RDFS/OWL<br>(semantics) |

Some implications for Web rule interchange:

- RIF shall support both XML and RDF as data formats
- RIF will use URIs as identifiers
- RIF shall allow to take RDFS and OWL domain models into account
- BTW: How to get from XML to RDF? W3C is working on it: GRDDL, RDFa, etc.

# Semantic Web architecture 4/5: XML vs. RDF(S)+OWL

| | XML | RDF |
|---|---|---|
| Data Model: | Tree | Graph |
| Identifiers: | element, attribute names | everything identified by URIs |
| Data: | in the leaves | in the nodes |
| Relations | in the nodes | in the edges |
| Data structure : | XML Schema (syntax) | RDFS/OWL (semantics) |

Some implications for Web rule interchange:

- RIF shall support both XML and RDF as data formats
- RIF will use URIs as identifiers
- RIF shall allow to take RDFS and OWL domain models into account
- BTW: How to get from XML to RDF? W3C is working on it: GRDDL, RDFa, etc.

# Semantic Web architecture 4/5: XML vs. RDF(S)+OWL

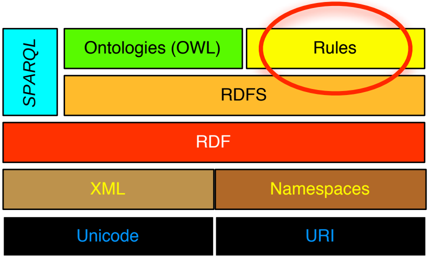
| | XML | RDF |
|---|---|---|
| Data Model: | Tree | Graph |
| Identifiers: | element, attribute names | everything identified by URIs |
| Data: | in the leaves | in the nodes |
| Relations | in the nodes | in the edges |
| Data structure : | XML Schema<br>(syntax) | RDFS/OWL<br>(semantics) |

Some implications for Web rule interchange:

- RIF shall support both XML and RDF as data formats
- RIF will use URIs as identifiers
- RIF shall allow to take RDFS and OWL domain models into account
- BTW: How to get from XML to RDF? W3C is working on it: GRDDL, RDFa, etc.

# Semantic Web architecture 5/5: Rules

- After exchanging Data and Domain Models on the Web has been enabled, Rules are the next step! ⇒ RIF



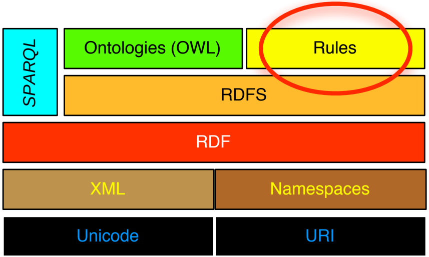
Let's talk about RIF's current state . . .

. . . 2 working drafts produced so far:

- Use Cases and Requirements
- RIF Core Design

# Semantic Web architecture 5/5: Rules

- After exchanging Data and Domain Models on the Web has been enabled, Rules are the next step! ⇒ RIF



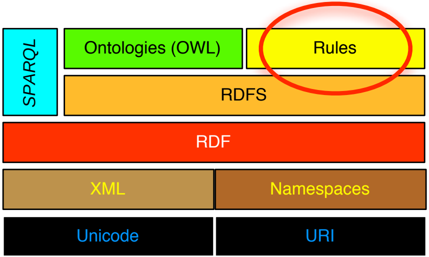
Let's talk about RIF's current state . . .

. . . 2 working drafts produced so far:

- Use Cases and Requirements
- RIF Core Design

# Semantic Web architecture 5/5: Rules

- After exchanging Data and Domain Models on the Web has been enabled, Rules are the next step! ⇒ RIF



Let's talk about RIF's current state …

…2 working drafts produced so far:

- Use Cases and Requirements
- RIF Core Design

# Current Status

Use Cases and Requirements

- almost 50 use cases for a rule interchange format submitted
- 2 Public Working Drafts of *’RIF Use Cases and Requirements’*
  - use cases from various application domains
  - requirements mainly for Phase I
- a refined Working Draft underway
- we gather Phase II requirements at moment

RIF Core

- 1st Public Working Draft of *’RIF Core Design’*
- published end of March 2007

# RIF Core Design

RIF Core shall cover the minimal overlap of different Rule dialects, that is

- an extensible formalism to express “basic” conditions
- an simple framework for “basic” rules

⇒ “basic” = **positive Horn** rules

- allow to define rulesets
- provide formal underpinning for interoperation with the remaining Semantic Web architecture

⇒ an extensible **architecture** to build RIF “dialects” around a common Core

# RIF Core Design

RIF Core shall cover the minimal overlap of different Rule dialects, that is

- an extensible formalism to express “basic” conditions
- an simple framework for “basic” rules

⇒ “basic” = **positive Horn** rules

- allow to define rulesets
- provide formal underpinning for interoperation with the remaining Semantic Web architecture

⇒ an extensible **architecture** to build RIF “dialects” around a common Core

# RIF Core Design

RIF Core shall cover the minimal overlap of different Rule dialects, that is

- an extensible formalism to express “basic” conditions
- an simple framework for “basic” rules

⇒ “basic” = **positive Horn** rules

- allow to define rulesets
- provide formal underpinning for interoperation with the remaining Semantic Web architecture

⇒ an extensible **architecture** to build RIF “dialects” around a common Core

# RIF Core Design

RIF Core shall cover the minimal overlap of different Rule dialects, that is

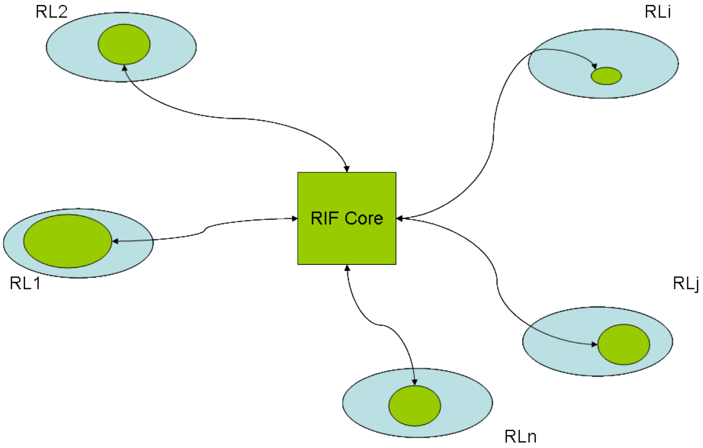
- an extensible formalism to express “basic” conditions
- an simple framework for “basic” rules

⇒ “basic” = **positive Horn** rules

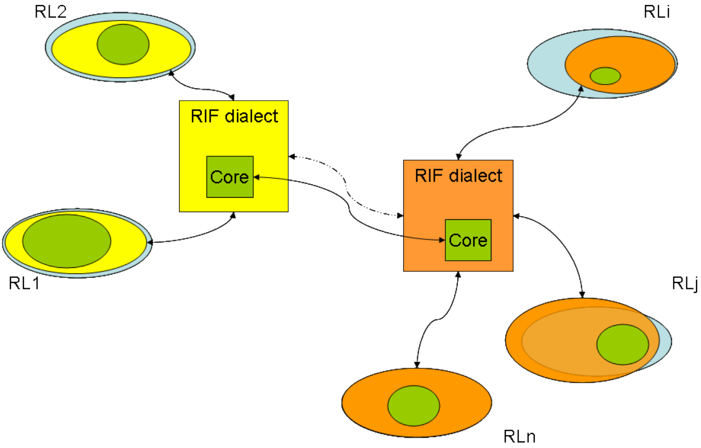
- allow to define rulesets
- provide formal underpinning for interoperation with the remaining Semantic Web architecture

⇒ an extensible **architecture** to build RIF “dialects” around a common Core

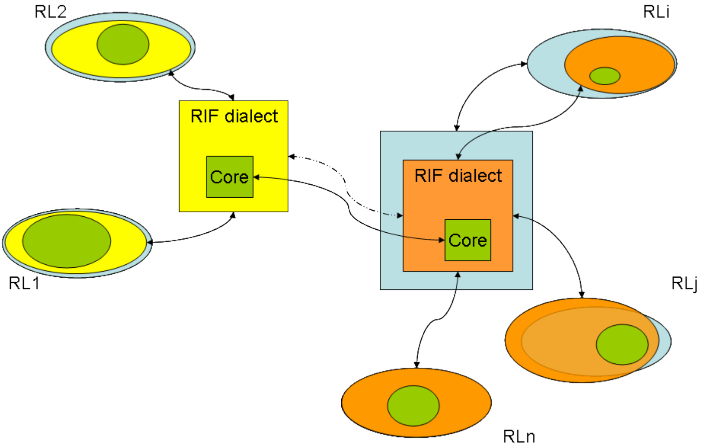
# RIF Architecture 1/2



# RIF Architecture 1/2



# RIF Architecture 1/2



# RIF Architecture 2/2

Required:

- Ruleset
    - Annotation: Semantics, Dialect, Name, Description, . . .
- Rule
    - Annotation: Name, Description, . . .
    - Event (`ON`)
    - Condition (`IF`)
    - Effect (`THEN`)
    - Action (`DO`)
    - . . .

Start with positive Horn:

`IF`: conjunctions (and disjunctions) of atomic conditions

`THEN`: atomic formulae

# RIF Architecture 2/2

Required:

- Ruleset
    - Annotation: Semantics, Dialect, Name, Description, . . .
- Rule
    - Annotation: Name, Description, . . .
    - Event (`ON`)
    - Condition (`IF`)
    - Effect (`THEN`)
    - Action (`DO`)
    - . . .

Start with positive Horn:

`IF`: conjunctions (and disjunctions) of atomic conditions

`THEN`: atomic formulae

# RIF Architecture 2/2

Required:

- Ruleset
  - Annotation: Semantics, Dialect, Name, Description, ...
- Rule
  - Annotation: Name, Description, ...
  - Event (`ON`)
  - Condition (`IF`)
  - Effect (`THEN`)
  - Action (`DO`)
  - ...

Start with positive Horn:

`IF`: conjunctions (and disjunctions) of atomic conditions

`THEN`: atomic formulae

# RIF Architecture 2/2

Required:

- Ruleset
  - Annotation: Semantics, Dialect, Name, Description, . . .
- Rule
  - Annotation: Name, Description, . . .
  - Event (`ON`)
  - Condition (`IF`)
  - Effect (`THEN`)
  - Action (`DO`)
  - . . .

Start with positive Horn:

`IF`: conjunctions (and disjunctions) of atomic conditions

`THEN`: atomic formulae

# RIF Architecture 2/2

Required:

- Ruleset
  - Annotation: Semantics, Dialect, Name, Description, . . .
- Rule
  - Annotation: Name, Description, . . .
  - Event (ON)
  - Condition (IF)
  - Effect (THEN)
  - Action (DO)
  - . . .

Start with positive Horn:

IF: conjunctions (and disjunctions) of atomic conditions

THEN: atomic formulae

$$\text{IF } C_1 \text{ AND } C_2 \text{ AND } \ldots \text{AND } C_n \text{ THEN } A$$

# RIF Architecture 2/2

Required:

- Ruleset
  - Annotation: Semantics, Dialect, Name, Description, . . .
- Rule
  - Annotation: Name, Description, . . .
  - Event (`ON`)
  - Condition (`IF`)
  - Effect (`THEN`)
  - Action (`DO`)
  - . . .

Start with positive Horn:

`IF`: conjunctions (and disjunctions) of atomic conditions

`THEN`: atomic formulae

$$(\forall)\ C_1 \wedge C_2 \wedge \ldots \wedge C_n \rightarrow A$$

# RIF Architecture 2/2

Required:

- Ruleset
  - Annotation: Semantics, Dialect, Name, Description, . . .
- Rule
  - Annotation: Name, Description, . . .
  - Event (`ON`)
  - Condition (`IF`)
  - Effect (`THEN`)
  - Action (`DO`)
  - . . .

Start with positive Horn:

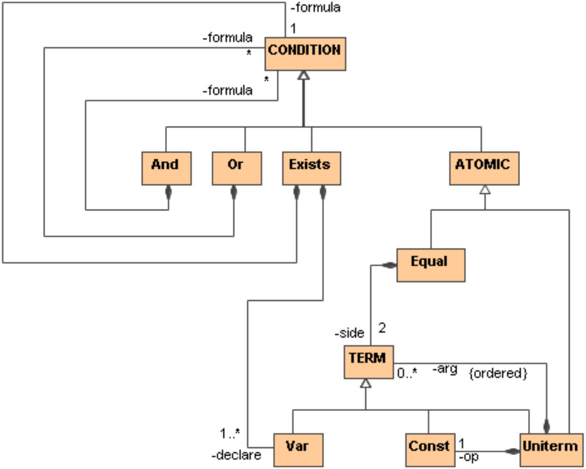
`IF`: conjunctions (and disjunctions) of atomic conditions

`THEN`: atomic formulae

$$(\forall)\ \neg C_1 \lor \neg C_2 \lor \ldots \lor \neg C_n \lor A$$

# RIF Core Conditions

An extensible formalism to express basic conditions:



# RIF Core Conditions – Example 1/2

Example: IF movie ?M was produced before 1930

RIF “readable” version of this condition:

```
Exists ?Y (
    And ( "imd:Movie"( ?M ) "imd:Year"( ?M ?Y )
          "op:date-less-than"( ?Y "1930-01-01T00:00:00Z"^^dateTime )
```

- Names of predicates are “webized” (using URIs and namespaces like in XML and RDF)
- Builtin predicates, like `op:date-less-than` around XPath and XQuery functions and operators will be also standardized (in an extensible way)

# RIF Core Conditions – Example 2/2

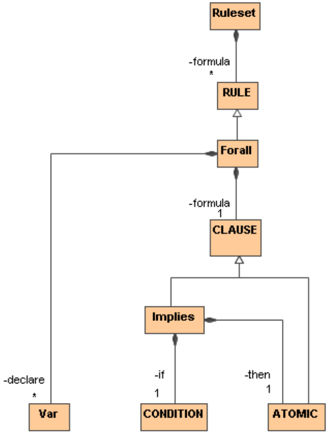
Mock-up XML serialization (currently under discussion):

```
  <Exists>
    <declare><Var>Y</Var></declare>
    <formula>
     <And>
      <formula>
        <Uniterm>
          <Const>Movie</Const>
          <Var>M</Var>
        </Uniterm>
      </formula>
      <formula>
        <Uniterm>
          <Const>Year</Const>
          <Var>M</Var>
          <Var>Y</Var>
        </Uniterm>
      </formula>
      <formula>
        <Uniterm type="builtin">
          <Const>date-less-than</Const>
         <Var>M</Var>
         <Const type="&xsd;dateTime">"1930-01-01T00:00:00</Const>
        </Uniterm>
      </formula>
  </Exists>
</And>
```

Discussed issues: How to markup typed constants, builtin functions, etc.

# RIF Core Horn Rules

A basic model for Horn rules:



Current focus:

- only cover simple IF-THEN rules
- provide a clean formal underpinning (model theory)

# RIF Core Horn Rule – Example

A rule “local” to a certain DVD shop:

IF dvd ?D shows movie ?M and ?M was produced before 1930
THEN ?M is a black and white movie

```
"moviShop:BWMovie" ( ?M ) :-
  Exists ?D ?Y (
    And ( "moviShop:Dvd"( ?D ) "imd:shows"( ?D ?M )
          "imd:Movie"( ?M ) "imd:Year"( ?M ?Y )
          "op:date-less-than"( ?Y "1930-01-01T00:00:00Z"^^dateTime )
```

- XML syntax similarly discussed
- Keep door open for later extensibility
- Discussions how to integrate with RDF/OWL data and also other data models!

# RIF Core Horn Rule – Example

A rule "local" to a certain DVD shop:

IF dvd ?D shows movie ?M and ?M was produced before 1930
THEN ?M is a black and white movie

```
"moviShop:BWMovie" ( ?M ) :-
  Exists ?D ?Y (
    And ( "moviShop:Dvd"( ?D ) "imd:shows"( ?D ?M )
          "imd:Movie"( ?M ) "imd:Year"( ?M ?Y )
          "op:date-less-than"( ?Y "1930-01-01T00:00:00Z"^^dateTime )
```

- XML syntax similarly discussed
- Keep door open for later extensibility
- Discussions how to integrate with RDF/OWL data and also other data models!

# Towards RIF Dialects for PR and ECA Rules

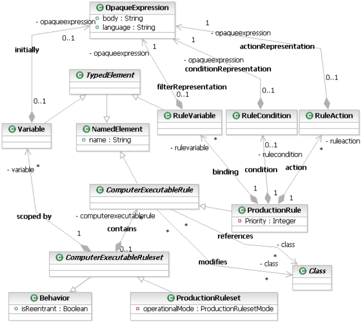
We have now

- a (more or less) stable core interchange format and
- a strong interest in extending it with PR and ECA rules

Newly established initiative

- group of representatives from ILOG, NRC Canada, REWERSE, ruleCore, TIBCO, TU Munich, etc.
- part of them also involved within the W3C RIF WG
- started the work towards PR- and ECA-based extensions to RIF Core
- ...and we still look for organizations supporting this effort! :)

# Related Efforts: W3C RIF and OMG PRR



- Alignment definitely desirable (see also Christian’s talk!)
- Alignment with related efforts in W3C (and not only) via so-called “*Liaisons*”

# Concluding Remarks

- Developing a useful format for rules on the Web
  - is a challenging but time-consuming task
  - different communities (e.g. PR vendors, Semantic Web researchers) are interested in it
- First steps towards a simple and extensible core format
  - for interchanging derivation rules
  - published as RIF Core in a 1st Working Draft of W3C
- More interesting and useful extensions to RIF Core in the near future
- ... follow the work at `http://www.w3.org/2005/rules/`