Adrian Paschke

Director RuleML Inc.
Research Director CITT GmbH
Prof. for Corporate Semantic Web
Free University Berlin

Adrian.Paschke AT gmx.de
Agenda

- Semantic Web 2.0 – An Introduction
- Semantic Web Rules
  - RuleML
  - W3C RIF
- What comes next?
  - Internet of Services and Things Web 3.0
  - Era of Events / Rule-based Intelligent CEP
  - Ubiquitous Pragmatic Web of Agents 4.0
"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

Tim Berners-Lee, James Hendler, Ora Lassila, *The Semantic Web*

„Make the Web understandable for machines“
Building Blocks of the Semantic Web (and beyond)

1. Explicit Metadata on the WWW
2. Ontologies
3. Logic and Inference
4. Software Agents and Web Services
1. Explicit Metadata on the Web

- Metadata are data about data

- Metadata on the Web:
  - *Machine processable information about information on the Web*
  - Projects e.g.: PICS, Dublin Core, RDF, IEEE LOM (Learning Objects Metadata), FOAF, ...

- Problem domains:
  - Syntax:
    - Which representation and interchange format for metadata?
  - Semantics:
    - Which metadata are allowed for resources (metadata vocabulary, schema)
  - Association problem:
    - How to connect metadata with resources (who defines the metadata, are metadata separated from the content, etc.)
2. Ontologies

“An ontology is an explicit specification of a conceptualization “
T. Gruber

Ontologies described the common knowledge of a domain (semantics):
→ Semantics interoperability between (connected) vocabularies

Typical components:
1. Classes (concepts) of the domain
2. Properties (roles) of the classes
3. Constraints
4. Individuals (instances) of classes
3. Logic and Inference

- Logic is a discipline concerned with the principles of inference and reasoning.

- Formal languages for the representation of knowledge with clear semantics:
  - Declarative knowledge representation: 
    - express what is valid, the responsibility to interpret this and to decide on how to do it is delegated to an interpreter / reasoner

- Automated reasoner, e.g., a rule engine, can derive conclusions from given knowledge (inference)
4. Software Agents and Web Services

Intelligent Software Agents act autonomously and pro-active
- They have an internal knowledge base with decision/reaction logic (e.g. rule-based expert systems)
- Examples: Personal agents (e.g. Rule Responder), search robots

Web Service
- In general: any IT service provided on the Web
- “A 'Web service' (also Web Service) is defined by the W3C as "a software system designed to support interoperable Machine to Machine interaction over a network." Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.” (Wikipedia)

=> no clear separation between web agents and web services (in the broad sense)
- but level of self-autonomous decisions is higher in web agents
Layered Architecture of the Semantic Web

- Principles (Original Semantic Web Stack as of 2003)
  - Development in layers – each layer depends on the other
  - Downwards compatible
  - Up-wards: partial understanding
Overview on the Semantic Web Layers (1)

- **XML Layer**
  - Syntactic basis

- **RDF Layer**
  - RDF as data model for facts and metadata
  - RDF schema (RDFS) as simple ontology language (mainly taxonomies)

- **Ontology Layer**
  - Expressive ontology languages
  - Actual standard: Web Ontology Language (OWL)
Overview on the Semantic Web Layers (2)

- Logic Layer
  - Extension of the ontology languages, e.g. with rules

- Proof Layer
  - Generation of proofs, interchange of proofs, validation

- Trust Layer
  - Digital signatures
  - Recommendations, ratings
Web Rules: Semantic Web + Rules

Homogeneous approach

Hybrid approach
Current Semantic Web Stack

W3C Semantic Web Stack as of 2007
Web Rules Technology

Users employ rules to express what they want, the responsibility to interpret this and to decide on how to do it is delegated to an interpreter.

Represent knowledge in a way that is understandable by ‘the business’, but also executable by rule engines, thus bridging the gap between business and technology.
Types of Web Rules

- Business
  - State constraints
  - Derivation rules
- Application
  - Action rules
  - Process constraints
- Technology
- Information
- Behaviour
- Structure

Structural constraints
Usage of Web Rules

1. Rules that influence the operational / decision processes:
   • *Derivation rules* (deduction rules): establish / derive new information that is used e.g. in a decision process.
   • *Reaction rules* that establish when certain activities should place:
     • *Condition-Action rules* (*production rules*)
     • *Event-Condition-Action (ECA) rules* + varians (e.g. ECAP).
     • *Messaging reaction rules* (event message / query reaction rules)

2. Constraints on organisation’s (or systems) structure, behaviour or information:
   • *Structural constraints* (deontic assignments).
   • *State constraints*.
   • *Process / flow constraints*. 
### Example Bonus Malus SLA Policy

#### Bonus-Malus System

<table>
<thead>
<tr>
<th>Quality of Service (QoS)</th>
<th>Average Availability (quantitative)</th>
<th>Bonus/Malus Discount</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>100 %</td>
<td>+ 5%</td>
</tr>
<tr>
<td>Normal</td>
<td>98-100 %</td>
<td>+ 0%</td>
</tr>
<tr>
<td>Low</td>
<td>&lt;98 %</td>
<td>- 5%</td>
</tr>
<tr>
<td>Below average</td>
<td>&lt;95 %</td>
<td>1000 $ penalty</td>
</tr>
</tbody>
</table>

If average availability is 100 % then QoS is high.

- **Body**
  - Predicate: =
  - Complex Term: availability(Service)
  - Constant: 100%

- **Head**
  - Predicate: qos
  - Variable: Service
  - Constant: high

If QoS is high then provide a bonus of 5% on the base price.

- **Body**
  - Predicate: qos
  - Variable: Service
  - Constant: high

- **Head**
  - Predicate: discount
  - Variable: Service
  - Constant: 5%
Web Rule Languages

- Computational Independent
  - Platform Independent
  - Platform Specific

- Specific
  - ILog
  - Blaze
  - IRL
  - SRL
  - Prova

- Independent
  - SBVR
  - PRR

- Platform Independent
  - RuleML

- Computational Independent
  - RIF
Rule Markup and Modeling Initiative (RuleML)
(www.ruleml.org)
- representatives from academia, industry and government
- promotion of the modern and future generations of Web rule technology

RuleML is currently the de facto open language standard for Web Rules

Collaborating with W3C (RIF), OMG (PRR), OASIS, DARPA and other standards/gov't bodies
RuleML Enables ...

Rule

- modelling
- markup
- translation
- interchange
- execution
- publication
- archiving

in

UML
RDF
XML
ASCII
Rule Interchange

RuleML + Test Suites

Rule system 1

Rule system 2

Data

<XML doc>

Application A

Application B

maps

Agreed on vocabulary
(e.g. OWL, RDFS)
RuleML Language Family

- Derivation Rules
- Reaction Rules
- Integrity Constraints
- Transformation Rules

Layered Approach

RuleML Translators

Reaction RuleML

Integrity RuleML

Derivation RuleML
RuleML is specified by a set of modular XSDs

XML Schema + EBNF Syntax

Full RDF compatibility via type and role tags (akin to triple syntax);

XML Schema Modularization: Layered and uniform design
"The **discount** for a *customer* buying a *product* is **5.0 percent** if the *customer* is **premium** and the *product* is **regular**."

```
<Implies>
  <head>
    <Atom>
      <op><Rel>discount</Rel></op>
      <Var>customer</Var>
      <Var>product</Var>
      <Ind>5.0 percent</Ind>
    </Atom>
  </head>
  <body>
    <And>
      <Atom>
        <op><Rel>premium</Rel></op>
        <Var>customer</Var>
      </Atom>
      <Atom>
        <op><Rel>regular</Rel></op>
        <Var>product</Var>
      </Atom>
    </And>
  </body>
</Implies>
```
Scope of Reaction RuleML

Reaction RuleML

- Active Database
- Production Rule Systems
- Rule Based Event Notification Systems
- Distributed Complex Event Processing

Transient Events
- ECA Paradigm
- Global Active Rules
- Trigger EA Rule
- Complex Event Algebra

Implicit Sequence of Knowledge Update
- CA Rule

Event Action Message
- Inbound
- Outbound
- Enterprise Service Bus
- Agent Conversation
- Protocol
- Performative
- FIPA-ACL

Event Action Axiom
- Reasoning on Effect Transitions
- Fluent state processes
- Automata state machines
- Petri nets or calculus
Reaction RuleML

- Reaction RuleML (http://ibis.in.tum.de/research/ReactionRuleML/)

- Quasi-Standard for Reactive Web Rules
  - Production rules, ECA rules and variants such as Trigger (EA), intelligent rule-based CEP, KR Event/Action Logics, Process Algebras, …

- Application Domains:
  - Event Processing Networks
  - Event Driven Architectures (EDAs)
  - Reactive, rule-based Service-Oriented Architectures (SOAs)
  - Active Semantic Web Applications
  - Real-Time Enterprise (RTE)
  - Business Activity Management (BAM)
  - Business Performance Management (BPM)
  - Service Level Management (SLM) with active monitoring and enforcing of Service Level Agreements (SLAs) or e-Contracts
  - Supply Chain Event Management
  - Policies
  - Web-based Workflow Systems
  - …
General Concepts

- General (reaction) rule form that can be specialized as needed

- Three general execution styles:
  - **Active**: 'actively' polls/detects occurred events in global ECA style, e.g. by a ping on a service/system or a query on an internal or external event database
  - **Messaging**: Waits for incoming complex event message
  - **Reasoning**: KR event/action logic reasoning and transitions (as e.g. in Event Calculus, Situation Calculus, TAL formalizations)

- Appearance
  - **Global**: ‘globally’ defined reaction rule
  - **Local**: ‘locally’ defined (inline) reaction rule nested in an outer rule
General Syntax for Reaction Rules (Reaction RuleML 0.2)

```xml
<Rule style="active" eval="strong">
    <on>
        <!-- event -->
    </on>

    <if>
        <!-- condition -->
    </if>

    <do>
        <!-- action -->
    </do>

    <ifPost>
        <!-- postcondition -->
    </ifPost>

    <doAlternative>
        <!-- alternative/else action -->
    </doAlternative>
</Rule>
```
Reaction RuleML – Rule Type Examples

- **Derivation Rule:**
  ```xml
  <Rule style="reasoning">
    <if>...</if>
    <then>...</then>
  </Rule>
  ```

- **Production Rule:**
  ```xml
  <Rule style="active">
    <if>...</if>
    <do>...</do>
  </Rule>
  ```

- **ECA Rule:**
  ```xml
  <Rule style="active">
    <on>...</on>
    <if>...</if>
    <do>...</do>
  </Rule>
  ```
Example: Request / Query

<Message mode="outbound" directive="ACL:query-ref">
  <oid> <Ind>RuleML-2008</Ind> </oid>
  <protocol> <Ind>esb</Ind> </protocol>
  <sender> <Ind>User</Ind> </sender>
  <content>
    <Atom>
      <Rel>getContact</Rel>
      <Ind>Sponsoring</Ind>
      <Var>Contact</Var>
    </Atom>
  </content>
</Message>

Event Message is local to the conversation state (oid) and pragmatic context (directive)
Complex Event / Action Algebra Operators

■ Action Algebra:

  *Succession* (Ordered Succession of Actions), *Choice* (Non-Deterministic Choice), *Flow* (Parallel Flow), *Loop* (Loops)

■ Event Algebra:

  *Sequence* (Ordered), *Disjunction* (Or), *Xor* (Mutual Exclusive), *Conjunction* (And), *Concurrent*, *Not*, *Any*, *Aperiodic*, *Periodic*

```xml
<event>
  <Sequence>
    <Concurrent>
      <Ind>a</Ind>  <Ind>b</Ind>
    </Concurrent>
    <Ind>c</Ind>
  </Sequence>
</event>
```
Selected Reaction RuleML Extended Features

- Support for Event / Action algebras
- Support for different selection and consumption policies
- Support for intervals (Time, Event)
- Support for situations (States, Fluents)
- Support for external event query languages
- Support for external vocabularies, e.g. Common Base Event (CBE)
- Support for external action execution / procedural attachments
- ...
W3C Rule Interchange Format

W3C RIF Working Group – started Dec. 2005

http://www.w3.org/2005/rules/wiki/RIF_Working_Group

Goal

Standardization of a web-based Rule Interchange Format (RIF)

Currently two dialects:

- RIF Production Rules Dialect (RIF-PRD)
  - specifies the RIF production rules dialect to enable the interchange of production rules
  - status: 1st working draft

- RIF Basic Logic Dialect (RIF-BLD)
  - specifies a basic interchange format that allows logic rules (definite Horn rules with equality) to be exchanged
  - status: last call
Definite Horn rules with equality + a number of syntactic extensions
- Slots, frames, internationalized resource identifiers (IRIs) as identifiers for concepts, XML Schema data types.

Syntax
- Normative XML syntax (closely related to striped RuleML XML syntax)
- Presentation syntax + abridged presentation syntax (for presentation purposes)
Example RIF-BLD

RIF condition

\[ \text{And (Exists ?Buyer (\text{?Buyer purchase (\text{?Buyer ?Seller}}) \text{?Seller = ?Author})} \]

XML serialization

```xml
<And>
  <formula>
    <Exists>
      <declare><Var>Buyer</Var></declare>
      <formula>
        <Atom>
          <op><Const type="&rif:iri">&cpt:purchase</Const></op>
          <args ordered="yes">
            <Var>Buyer</Var>
            <Var>Seller</Var>
            <Expr>
              <op><Const type="&rif:iri">&cpt:book</Const></op>
              <args ordered="yes">
                <Var>Author</Var>
                <Const type="&rif:iri">&bks:LeRif</Const>
              </args>
            </Expr>
            <Expr>
              <op><Const type="&rif:iri">curr:USD</Const></op>
              <args ordered="yes"><Const type="&xsd:integer">49</Const></args>
            </Expr>
          </op>
        </Atom>
      </formula>
    </Exists>
  </formula>
  <Equal>
    <side><Var>Seller</Var></side>
    <side><Var>Author</Var></side>
  </Equal>
</And>
```

Syntax almost similar to RuleML syntax
Production Rules (Condition-Action)
- Based on RIF condition language
- Actions: explicit *Retract*, implicit *ASSERT*
- Negation (default negation)

Syntax
- Normative XML syntax (closely related to Reaction RuleML XML syntax)
  - Might be concrete representation language for *OMG PRR*
- Presentation syntax + abridged presentation syntax (for presentation purposes)
<Implies>

<if> FORMULA </if>

<then rif:ordered="yes">

ACTION

ACTION*

</then>

</then>

</Implies>

Syntax is almost similar to Reaction RuleML syntax.
W3C Rule Interchange Format

- **RIF-FLD** (Framework of Logic Dialects)
  - framework of mechanisms for specifying the syntax and semantics of logic-based RIF dialects through a number of generic concepts

- **RIF-RRD** (Reaction Rules Dialect)
  - Dialect for RIF Reaction Rules
  - includes RIF Event Processing Language Dialect (EPL)

- **RIF-SWC** (RDF and OWL Compatibility)
  - Interoperabilität zwischen RIF und Semantic Web Daten- und Ontologiesprachen (RDF, RDFS, OWL).

- **RIF-DTB** (Data Types and Builtins)
  - document describes RIF data types and built-in functions and predicates

- **RIF-UCR** (Use Cases and Requirements)

- **RIF-Test** (Test Cases)
RIF Objects and Class Hierarchies

■ Instance-Class Membership

\[ inst \ # \ class \]

\[<\text{Member}>

\[<\text{lower}>inst'</lower>

\[<\text{upper}>class'</upper>

\[</\text{Member}>\]

■ Subclass

\[ sub \ #\# \ super \]

\[<\text{Subclass}>

\[<\text{lower}>sub'</lower>

\[<\text{upper}>super'</upper>

\[</\text{Subclass}>\]
Syntactical Mapping of RIF Syntax to RDF triples

RIF type and role tags map to a RDF triple "Type1 role Type2"
RIF-SWC specifies the interoperation between RIF and the data and ontology languages RDF, RDFS, and OWL.

RDF triples

ex:john ex:brotherOf ex:jack
ex:jack ex:parentOf ex:mary

RIF Rule

Forall ?x, ?y, ?z
 (?x[ex:uncleOf -> ?z] :-

Importing RDF Graphs in RIF

Import(<IRI> <profile>)
Recomendations

- RIF Basic Logic Dialect (BLD)
- RIF Framework for Logic Dialects
- RIF Datatypes and Built-Ins
- RIF Production Rule Dialect
- RIF Reaction Rules Dialect
  - (includes RIF Event Processing Language)
- RIF Core: Dialect Interoperation and Extensible Core

Planned Working Draft

- RIF Logic Programming Dialect
- RIF Full First-Order Logic Dialect
- RIF Extension for XML Data Sources
Excurse

OMG PRR
Production Rules Representation

- Formal model for vendor-neutral rule-model representation in UML for production rules

- Consortium of developers and supporters
  - Rule vendors (including Fair Isaac, ILOG, LibRT, IBM, Pega, Corticon, TIBCO)
  - Academic community (RuleML.org)
  - Related vendor community (Fujitsu, IBM)

- PRR is currently "Adopted" as a standard and in "Finalization", meaning that it is in Beta, with a final version 1.0 in 2008.

- OMG MDA PIM model

- PRR beta spec applies just to a PRR Core using UML MOF
  - Currently spec. excludes an explicit expression language.

- Can be used as the basis for other efforts such as the W3C Rule Interchange Format and the production rule version of RuleML (Reaction RuleML).
if [condition] then [action-list]
What comes next?
Internet of Service and Internet of Things

Internet of Services

TCP/IP

Firewall

Firewall

UDDI

WSDL

Travel Services

Car Rental

Hotel Reservation

Banking

Billing Server

GPS/Galileo Weather

Event Reservations

Insurance

Internet of Things

Agent Services

Smart Things

User Interface & Applications

Semantic Web

Trust

Proof

Unifying Logic

Query: SPARQL

Ontology: OWL

Rule: RIF

Data interchange:

RDF

XML

URI/IRI

Storage with

Blades, Cluster

Network, virt. Images

Load Balancer

Switches

Network,

Semantic Web

Internet of Things

Business Services Network / Service Supply Chain

TCP/IP

WSDL
Complex Event Processing

Challenge for Complex Event Processing

Detection, prediction and mastery of complex events and situations are crucial to the competitiveness of networked businesses and the efficiency of dynamic distributed service infrastructures.
Complex Event Processing – What is it?

- Complex Event Processing (CEP) is a **discipline** that deals with event-driven behavior.

- Selection, aggregation, and event abstraction for generating higher level complex events of interest.

- Intelligent Rule-based CEP.
CEP Reference Architecture

Event Consumer
- Operator, Management

Event Processing Media
- Event Rating, Situation Detection, Prediction
- Event Consolidation, Composition, Aggregation, Detection
- Event Monitoring, Tracking, Discovery, Selection

Event Modeler
- Event Pattern Definition

Event Originator
- Event Sources / Event Producer

Event Consumption
- Event Handling
- Event Aggregation
- Event Selection
- Event Definition
- Event Production

Business Events
Complex Events
Atomic Events
Core CEP Operations

1. Event Production
2. Event Definition
3. Event Selection
4. Event Aggregation
5. Event Handling
6. Event Consumption

CEP Media
Thanks for Attention