

The RuleML Perspective on Deliberation-Reaction Standards

Adrian Paschke, Harold Boley, Tara Athan

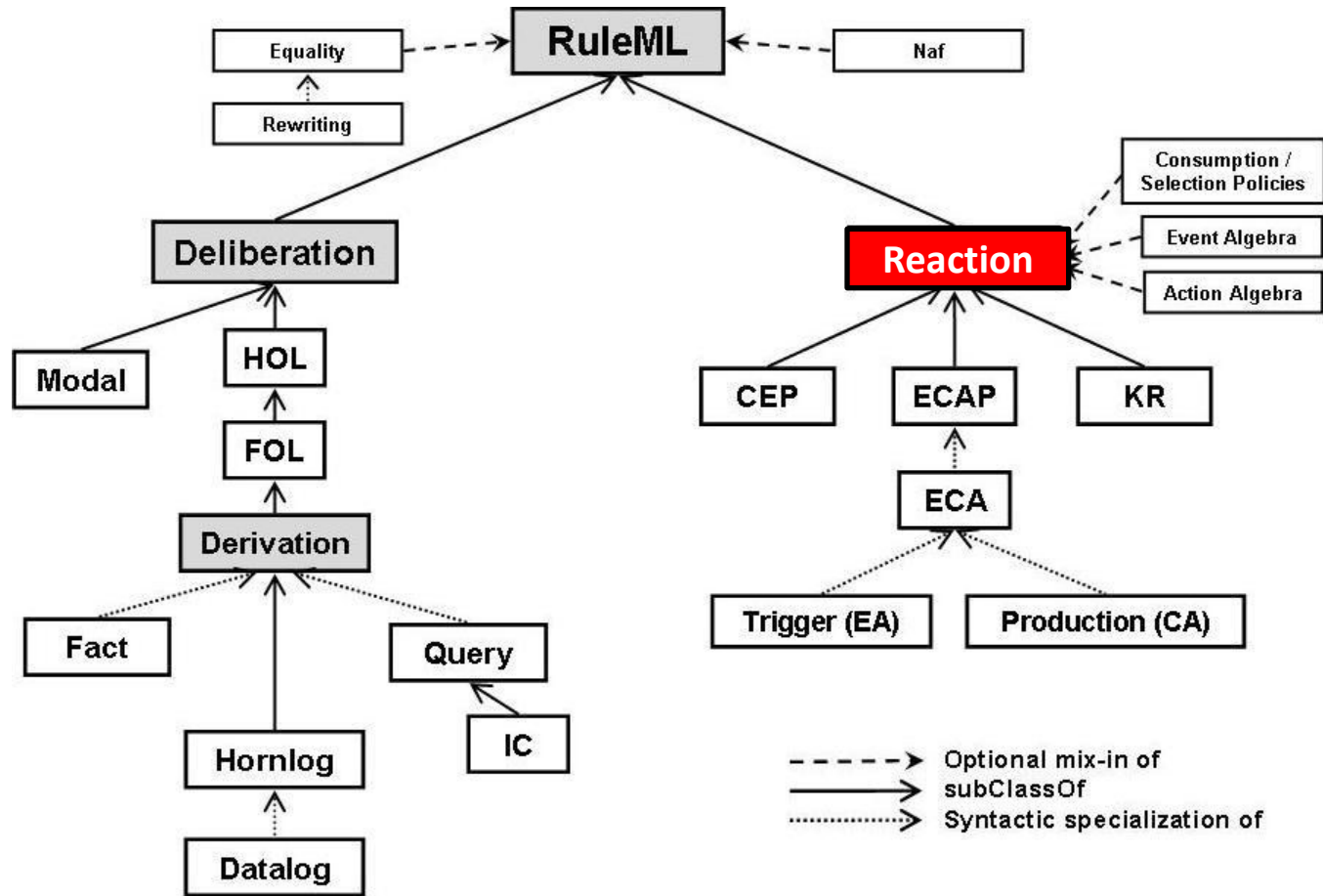
In **Ontolog RulesReasoningLP: Series Session 5**
9 January 2014



Agenda

- Reaction RuleML
- Semantic Profiles
- Semantic Metamodel and External Ontologies (Semantic Sorts)
- Syntactic Customization

The RuleML Family



Reaction RuleML 1.0 Paper: http://link.springer.com/chapter/10.1007%2F978-3-642-32689-9_9

Reaction RuleML 1.0 Tutorial: <http://www.slideshare.net/swadpasc/reaction-ruleml-ruleml2012paschketutorial>

RuleML 1.0 Paper http://link.springer.com/chapter/10.1007/978-3-642-16289-3_15

RuleML Overview Slides: <http://cs.unb.ca/~boley/talks/RuleML-Overarching-Talk.pdf>

Quick Overview: Reaction RuleML Dialects

* + variants and alternatives

- **Spatio-Temporal Derivation RuleML** (*if-then*)*
 - Time, Spatial, Interval
- **KR RuleML** (*if-then* or *on-if-do*)*
 - Situation, Happens_(@type), Initiates, Terminates, Holds, fluent
- **Production RuleML** (*if-do*)*
 - Assert, Retract, Update, Action
- **ECA RuleML** (*on-if-do*)*
 - Event, Action, + (event / action algebra operators)
- **CEP** (arbitrary combination of on, if, do)
 - Receive, Send, Message

(Reaction) Rules: Specializable Syntax

	<Rule @key @keyref @style>	
Info, Life Cycle Mgt.	<meta> <!-- descriptive metadata of the rule -->	</meta>
	<scope> <!-- scope of the rule e.g. a rule module -->	</scope>
Interface	<evaluation> <!-- intended semantics -->	</evaluation>
	<signature> <!-- rule signature -->	</signature>
Imple- mentation	<qualification> <!-- e.g. qualifying rule metadata, e.g. priorities, validity, strategy -->	</qualification>
	<quantification> <!-- quantifying rule declarations, e.g. variable bindings -->	</quantification>
	<oid> <!-- object identifier -->	</oid>
	<on> <!-- event part -->	</on>
	<if> <!-- condition part -->	</if>
	<then> <!-- (logical) conclusion part -->	</then>
	<do> <!-- action part -->	</do>
	<after> <!-- postcondition part after action, e.g. to check effects of execution -->	</after>
	<else> <!-- (logical) else conclusion -->	</else>
	<elsedo> <!-- alternative/else action, e.g. for default handling -->	</elsedo>
	</Rule>	

Reaction RuleML – Example Rule Types

- **Derivation Rule:**
(temporal/event/action/
reasoning)

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

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

```
<Rule style="active"> >  
  <on>...</on>  
  <do>...</do>  
</Rule>
```
- **ECA Rule:**

```
<Rule style="active">  
  <on>...</on>  
  <if>...</if>  
  <do>...</do>  
</Rule>
```

Agenda

- Reaction RuleML
- Semantic Profiles
- Semantic Metamodel and External Ontologies (Semantic Sorts)
- Syntactic Customization

Integration of Semantic Profiles

1. Include/Import external Semantic Profile

```
<xi:include href="../../../profiles/SituationCalculusProfile.rml" xpointer="xpointer(/RuleML/*)" />
<evaluation>
  <Profile keyref="&ruleml;ReifiedClassicalSituationCalculus" />
</evaluation>
```

2. Reference pre-defined Semantic Profile as profile type

```
<evaluation>
  <Profile type="&rif;RDFS" iri="http://www.w3.org/ns/entailment/RDFS"/>
</evaluation>
```

Reference published external semantic profiles, e.g. RIF, OWL, profiles ...

3. Locally define Semantic Profile

```
<Assert>
  <evaluation>
    <Profile key="&ruleml;ReifiedSituationCalculus" >
      <formula><Rulebase > ... RuleML definition... </Rulebase></formula>
      <content> ... xs:any XML content, e.g. RIF, Common Logic XML... </content>
    </Profile>
  </evaluation>
  <Rulebase>
    <Rule> ... </Rule>
    <Rule> ... </Rule>
  </Rulebase>
</Assert>
```

Note: also other non RuleML content models are supported

Complex Event Processing – Semantic Profiles

(defined in [<evaluation>](#) semantic profiles)

1. Definition

- Definition of event/action pattern e.g. by event algebra
- Based on declarative formalization or procedural implementation
- Defined over an atomic instant or an interval of time, events/actions, situation, transition etc.

2. Selection

- Defines selection function to select one event from several occurred events (stored in an event instance sequence e.g. in memory, database/KB) of a particular type, e.g. “*first*”, “*last*”
- Crucial for the outcome of a reaction rule, since the events may contain different (context) information, e.g. different message payloads or sensing information

3. Consumption

- Defines which events are consumed after the detection of a complex event
- An event may contribute to the detection of several complex events, if it is not consumed
- Distinction in event messaging between “multiple receive” and “single receive”
- Events which can no longer contribute, e.g. are outdated, should be removed

4. Execution

- Actions might have an internal effect i.e. change the knowledge state leading to state transition from (pre-)condition state to post-condition state
- The effect might be hypothetical (e.g. a hypothetical state via a computation) or persistent (update of the knowledge base),
- Actions might have an external side effect

Example - Reaction Rules with Semantic Profiles

```
<Rule style="active">  
  <evaluation>  
    <Profile> e.g. selection and consumptions policies </Profile>  
  </evaluation>  
  <signature>  
    <Event key="#ce1">  
      ... event pattern definition (for event detection)  
    </Event>  
  </signature>  
</Rule>
```

```
<on>  
  <Event keyref="#ce1"/> <!-- use defined event pattern for detecting events -->  
</on>  
<if>  
  ...  
</if>  
<do>  
  <Assert safety="transactional"> <!-- transactional update -->  
    <formula>  
      <Atom>  
        ...  
      </Atom>  
    </formula>  
  </Assert>  
</do>  
</Rule>
```

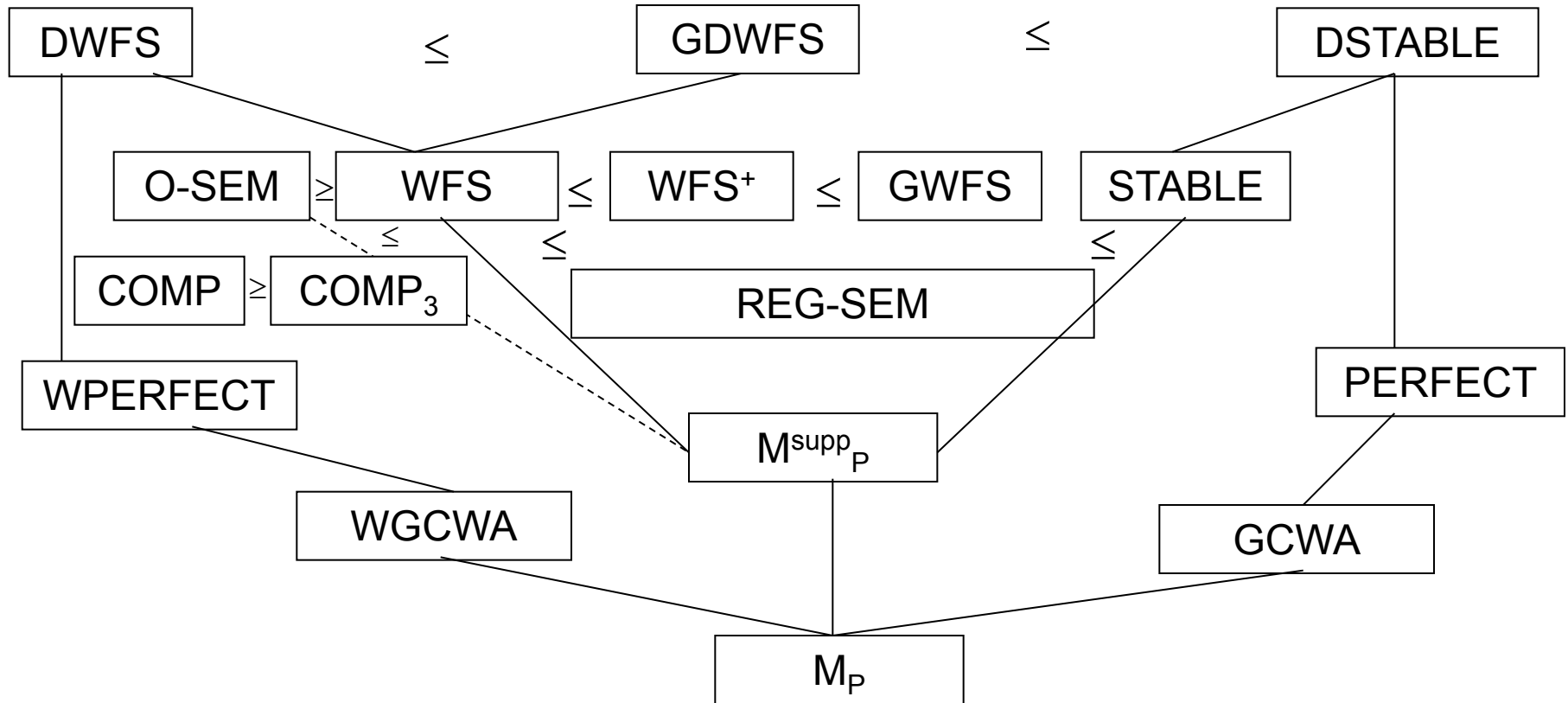
Interface

(semantic profile +
event pattern
signature for event
processing /
detection)

Implementation

(reaction rule
triggered by event
detection)

Semantic Layering - Example LP Semantic Profiles



A semantic SEM' extends a semantic SEM:

SEM' \geq SEM iff $\forall (P, F) \text{ SEM}(P) \models F \rightarrow \text{SEM}'(P) \models F$

Example: Use of Semantic LP Profiles for Interpretation

<!-- rule interface with two alternative interpretation semantics and a signature.

The interface references the implementation identified by the corresponding key -->

```
<Rule keyref="#r1">
```

```
<evaluation index="1">
```

```
<!-- WFS semantic profile -->
```

```
<Profile type="&ruleml;Well-Founded-Semantics" />
```

```
</evaluation>
```

```
<evaluation index="2">
```

```
<!-- alternative ASS semantic profile define in the metamodel -->
```

```
<Profile type="&ruleml;Answer-Set-Semantics" />
```

```
</evaluation>
```

```
<!-- the signature defines the queryable head of the backward-reasoning rule -->
```

```
<signature>
```

```
<Atom><Rel>likes</Rel><Var mode="+"/><Var mode="-"/></Atom>
```

```
</signature>
```

```
</Rule>
```

<!-- implementation of rule 1 which is interpreted either by WFS or by ASS semantics and only allows queries according to it's signature definition. -->

```
<Rule key="#r1" style="reasoning">
```

```
<if>... </if>
```

```
<then>
```

```
<Atom><Rel>likes</Rel><Var>X</Var><Var>Y</Var></Atom>
```

```
</then>
```

```
</Rule>
```

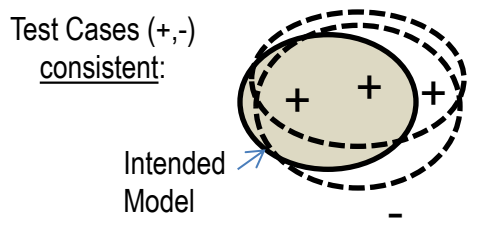
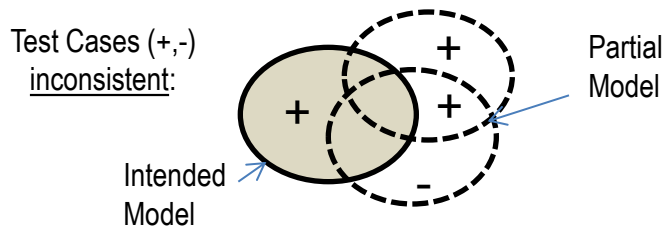
Interface
with
evaluation
semantics
and
public rule
signature

Implemen
tation

Test Cases for Self-validating Rule Bases

- Test Cases constrain the possible models and **approximate the intended models** of the rule base

- **Queries** are used to test the rule base



- A test case is defined by $T := \{X, A, N\}$, where

- $X \subseteq L$ assertion base (input data, e.g. facts)
- $A \in L$ a formula denoting a test query
- $N := +, -$ a positive or negative label

- Semantics

$$M_0 \models_{TC} (X, A, +) \text{ iff } \forall m \in M_0 : m \in \Sigma(\text{Mod}(X), R) \Rightarrow m \in \text{Mod}(A)$$

$$M_0 \models_{TC} (X, A, -) \text{ iff } \exists m \in M_0 : m \in \Sigma(\text{Mod}(X), R) \Rightarrow m \notin \text{Mod}(A)$$

- \models_{TC} compatibility relation
- Mod association function between sets of formulas and sets of models
- Σ model selection function

$$A \notin C_R(X) \text{ for } T := \{X, A, +\} \text{ and } A \in C_R(X) \text{ for } T := \{X, A, -\}$$

- $C_R(X)$ deductive closure of X . Decidable inference operator based on formal proofs

Agenda

- Reaction RuleML
- Semantic Profiles
- Semantic Metamodel and External Ontologies (Semantic Sorts)
- Syntactic Customization

RuleML Types (Sorted Logic)

- Types (sorts) can be assigned by using the **@type** attribute
- External vocabularies / ontologies define types, e.g.,
 - `<Var type="&vo;Vehicle">Car</Var>`
 - `<Ind iri="&vo;Corolla" type="&vo;Sedan"/>`
- Semantics defined / linked by Semantic Profile, e.g. order-sorted logic using RDFS or OWL entailment profiles

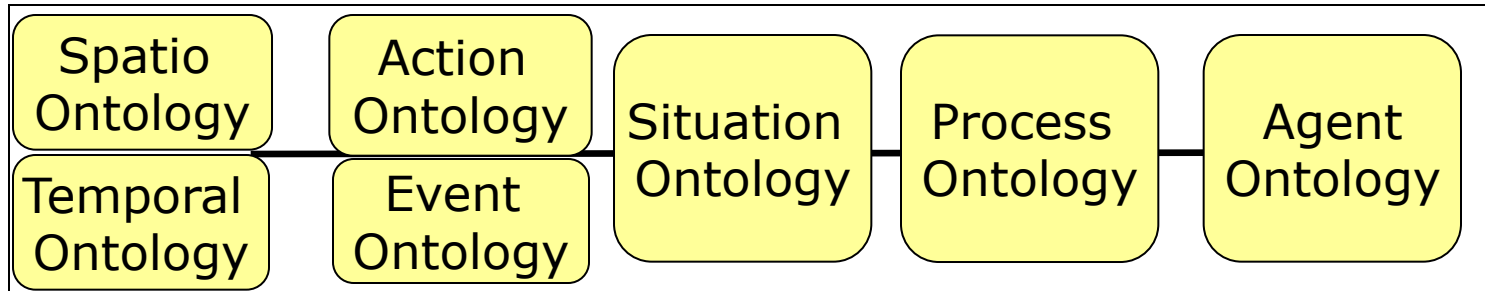
Reaction RuleML Examples with Types from RuleML Metamodel and External Ontologies

```
<Quantifier type="&ruleml;Forall"> == <Forall>  
<Operator type="&ruleml;And"> == <And>  
<Operator type="&ruleml;Conjunction"> == <Conjunction>  
<Negation type="&ruleml;InflationaryNegation"> == <Naf>  
<Action type="&ruleml;Assert"> == <Assert>  
<Action type="&ruleml;Retract"> == <Retract>  
<Event type="&ruleml;SimpleEvent"> == <Atom> ... </Atom>  
<Event type="ibm:CommonBaseEvent"> == IBM CBE  
<Operator type="snoop:Sequence"> == Snoop Algebra  
  == <Operator type="&ruleml;Sequence"> == <Sequence>  
<Ind iri="person.xml#xpointer(//Person/LastName[1]/text())"/>  
<Action iri="BPEL.xml#xpointer(//invoke[@name=checkHotel])">
```

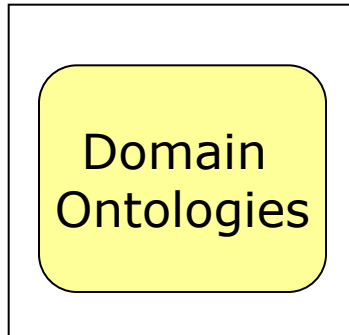

Reaction RuleML Metamodel

Top Level Ontologies

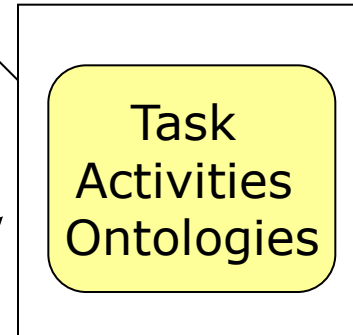
General concepts such as space, time, event, action and their properties and relations



Vocabularies **related to specific domains** by specializing the concepts introduced in the top-level ontology



Vocabularies **related to generic tasks or activities** by specializing the concepts introduced in the top-level ontology



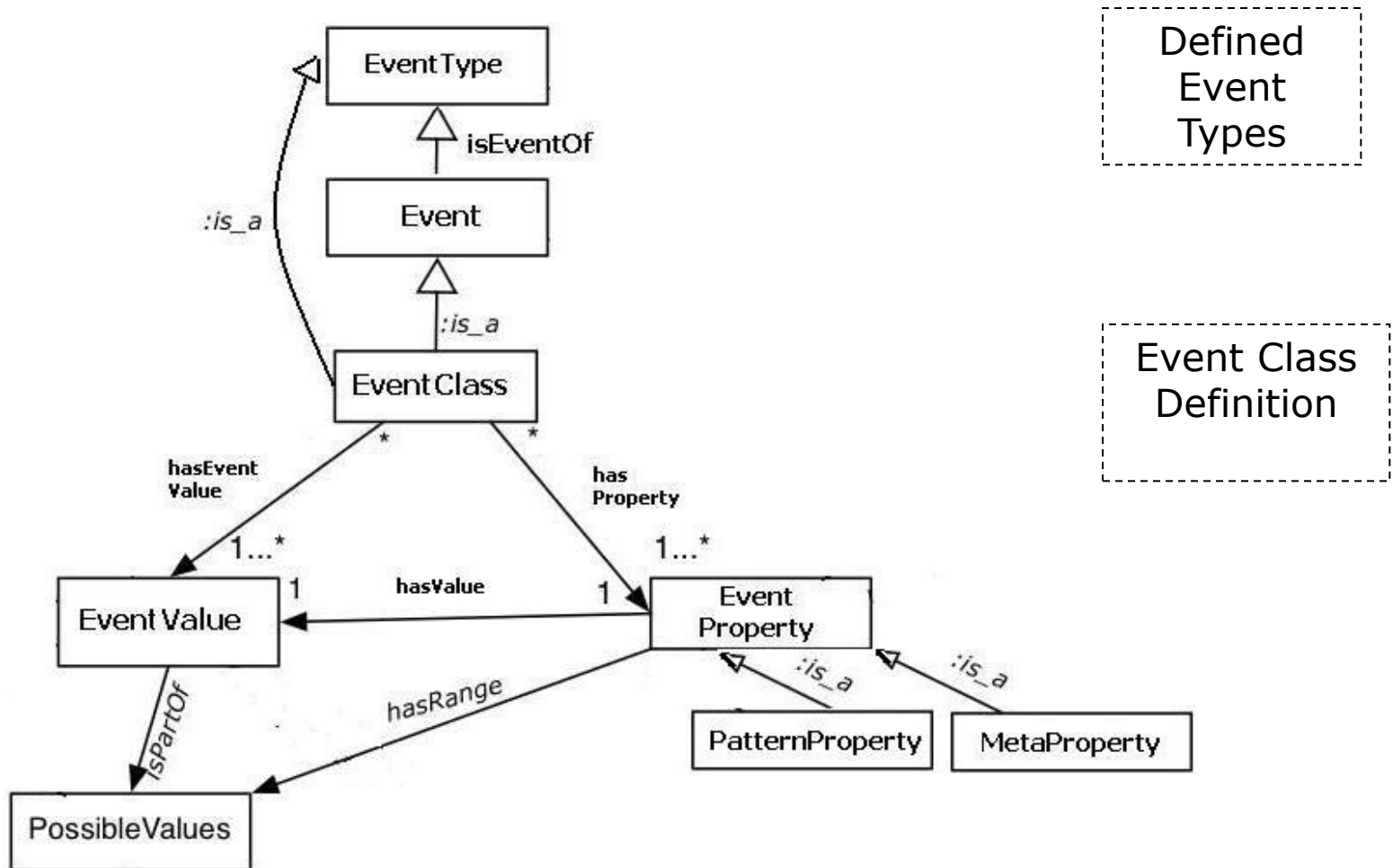
Specific user/application ontologies



E.g. ontologies describing roles played by domain entities while performing application activities

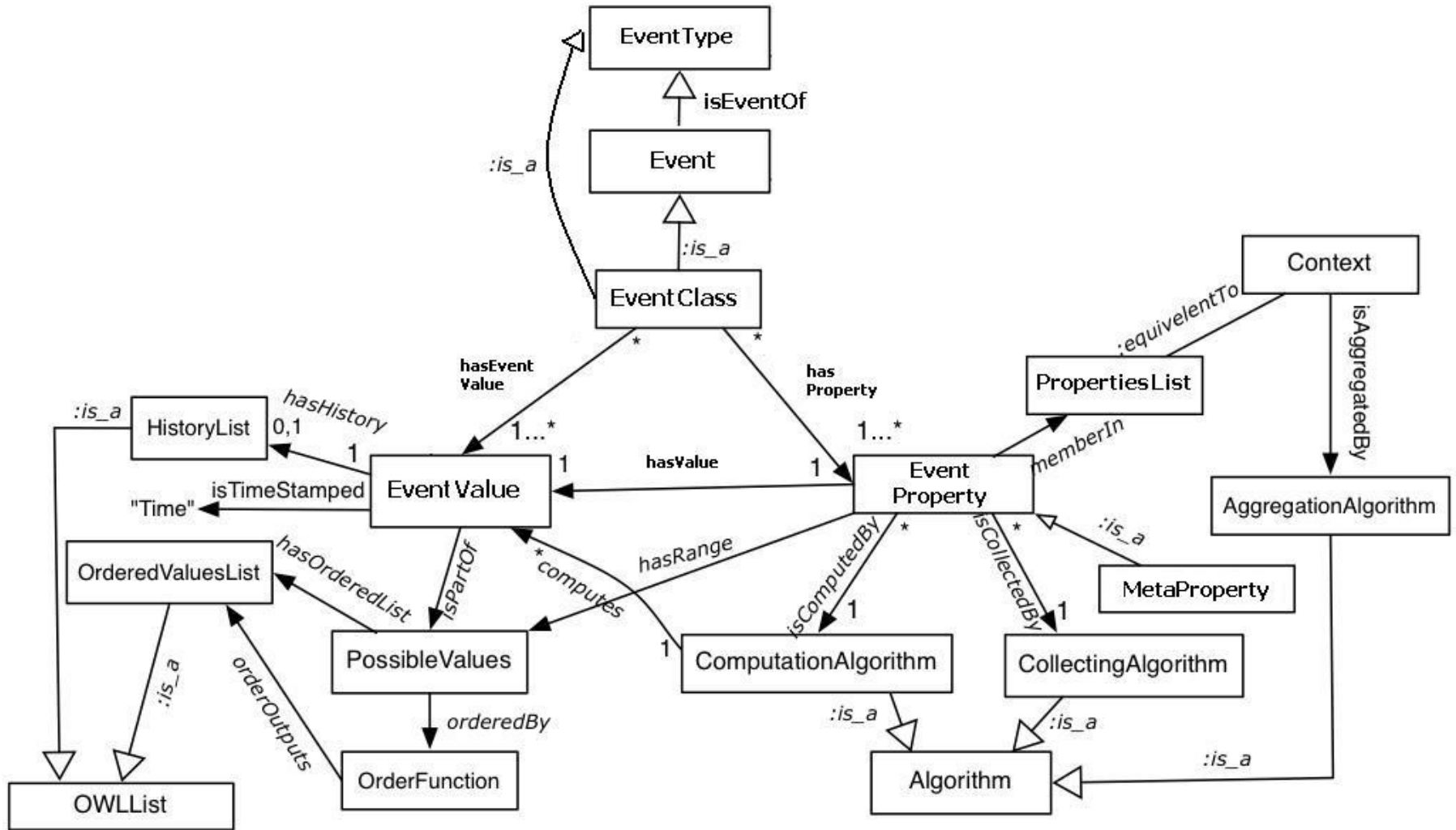
Example - Event MetaModel

(for defining Event Types as Instances of the MetaModel Event Class)



Extended Event Meta Model Ontology

(with computational properties and complex value definitions)



Example - Typed Complex Event Pattern Definition

```
<Event key="#ce2" type="&ruleml;ComplexEvent">
  <signature> <!-- pattern signature definition -->
    <Sequence>
      <signature>
        <Event type="&ruleml;SimpleEvent">
          <signature><Event>...event_A...</Event></signature>
        </Event>
      </signature>
      <signature><Event type="&ruleml;ComplexEvent" keyref="ce1"/></signature>
      <signature>
        <Event type="cbe:CommonBaseEvent" iri="cbe.xml#xpointer(//CommonBaseEvent)"/>
      </signature>
    </Sequence>
  </signature>
</Event>

<Event key="#ce1">
  <signature> <!-- event pattern signature -->
    <Concurrent>
      <Event><meta><Time>...t3</Time></meta><signature>...event_B</signature></Event>
      <Event><meta><Time>...t3</Time></meta><signature>...event_C</signature></Event>
    </Concurrent>
  </signature>
</Event>

<Event key="#e1" keyref="#ce2"><content>...</content></Event>
```

Event
Pattern
defining
Event
Templates
signature

Event
Instance¹⁹

Agenda

- Reaction RuleML
- Semantic Profiles
- Semantic Metamodel and External Ontologies (Semantic Sorts)
- Syntactic Customization

RuleML Sublanguages Customized by MYNG as Relax NG Schemas (1)



Selection Form

Instructions

Make a selection from the form below, then click "Refresh Schema" to update the Schema URL. The main module is also displayed below the form. To reset the form to the default (supremum) values, click "Reset Form".

Schema URL = http://ruleml.org/1.0/relaxng/schema_rnc.php?backbone=x3f&default=x7&termseq=x7&lng=x1&propo=x3ff&implies=x7&terms=xf3f&quant=x7&expr=xf&serial=xf

Expressivity "Backbone" (Check One)

- Atomic Formulas
- Ground Fact
- Ground Logic
- Datalog
- Horn Logic
- Disjunctive Logic
- Full First-Order Logic

Treatment of Attributes With Default Values (Check One)

- Required to be Absent
- Required to be Present
- Optional

Term Sequences: Number of Terms (Check One)

- None
- Binary (Zero or Two)
- Polyadic (Zero or More)

Language (Check One)

- English Abbreviated Names
- English Long Names
- French Long Names

Serialization Options (Check Zero or More)

- Unordered Groups
- Stripe-Skipping
- Explicit Datatyping
- Schema Location Attribute

RuleML Sublanguages Customized by MYNG as Relax NG Schemas (2)

Propositional Options (Check Zero or More)

- IRIs
- Rulebases
 - Entailments
- Degree of Uncertainty
- Strong Negation
- Weak Negation (Negation as Failure)
- Node Identifiers
- In-Place Annotation
- XML base
- XML id

Implication Options (Check Zero or More)

- Equivalences
- Inference Direction
- Non-Material

Term Options (Check Zero or More)

- Object Identifiers
- Slots
 - Slot Cardinality
 - Slot Weight
- Equations
 - Oriented
- Term Typing
- Data Terms
- Skolem Constants
- Reified Terms

Quantification Options (Check Zero or More)

- Implicit Closure
- Slotted Rest Variables
- Positional Rest Variables

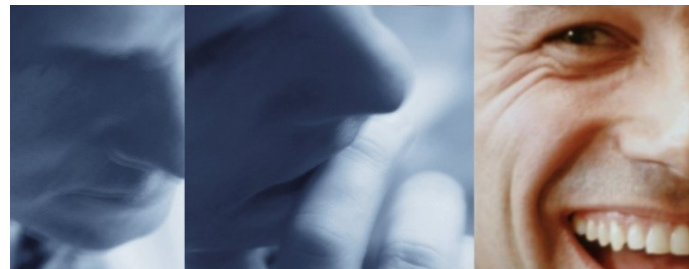
Expression Options (Check Zero or More)

- Generalized Lists
- Set-valued Expressions
- Interpreted Expressions

Summary

- Fine grained **syntactic language configuration** (expressiveness) with MYNG (as Relax NG schemas)
- Intended semantics defined by **Semantic Profiles**
- **Semantic Test Cases** for verification and validation of the intended semantics of rule programs (and their scoped modules)
- Sorted Logic supports **Types** defined in external semantic ontologies and Meta Model vocabulary

Thank you !



Questions?

Acknowledgement to the members of the Reaction RuleML
technical group

RuleML Online Community

- RuleML MediaWiki (<http://wiki.ruleml.org>)
- Mailing lists (<http://ruleml.org/mailman/listinfo>)
- Technical Groups
(http://wiki.ruleml.org/index.php/Organizational_Structure#Technical_Groups)
 - Uncertainty Reasoning
 - Defeasible Logic
 - Reaction Rules
 - Multi-Agent Systems
 - ...
- RuleML sources are hosted on Github
(<https://github.com/RuleML>)

Further Reading – (Reaction) RuleML

- Harold Boley, Adrian Paschke, Omair Shafiq: RuleML 1.0: The Overarching Specification of Web Rules. RuleML 2010: 162-178
http://dx.doi.org/10.1007/978-3-642-16289-3_15
<http://www.cs.unb.ca/~boley/talks/RuleML-Overarching-Talk.pdf>
- Adrian Paschke, Harold Boley, Zhili Zhao, Kia Teymourian and Tara Athan: Reaction RuleML 1.0: Standardized Semantic Reaction Rules, 6th International Conference on Rules (RuleML 2012), Montpellier, France, August 27-31, 2012
http://link.springer.com/chapter/10.1007%2F978-3-642-32689-9_9
<http://www.slideshare.net/swadpasc/reaction-ruleml-ruleml2012paschketutorial>
- Adrian Paschke: Tutorial on Semantic Complex Event Processing and Reaction RuleML, at DemAAL 2013 - Dem@Care Summer School on Ambient Assisted Living, 16-20 September 2013, Chania, Crete, Greece
<http://www.slideshare.net/swadpasc/dem-aal-semanticceppaschke>
- Paschke, A., Boley, H.: Rule Markup Languages and Semantic Web Rule Languages, in Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, ISBN:1-60566-402-2, 2009
<http://www.igi-global.com/chapter/rule-markup-languages-semantic-web/35852>
Sample chapter: <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=18533385>
- Paschke, A., Boley, H.: Rules Capturing Event and Reactivity, in Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, ISBN:1-60566-402-2, 2009
<http://www.igi-global.com/book/handbook-research-emerging-rule-based/465>
Sample Chapter: <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=16435934&article=0&fd=pdf>
- Adrian Paschke and Harold Boley: Rule Responder: Rule-Based Agents for the Semantic-Pragmatic Web, in Special Issue on Intelligent Distributed Computing in International Journal on Artificial Intelligence Tools (IJAIT), Vol. 20,6, 2011
<https://www.researchgate.net/publication/220160498>