RuleML 1.02: Deliberation, Reaction, and Consumer Families

Tara Athan[1] Harold Boley[2] Adrian Paschke[3]

[1] Athan Services (athant.com), West Lafayette, Indiana, USA
taraathan@gmail.com
[2] Faculty of Computer Science, University of New Brunswick, Fredericton, Canada
harold.boley@unb.ca
[3] AG Corporate Semantic Web, Freie Universitaet Berlin, Germany
paschke@inf.fu-berlin.de

Abstract. We describe the development of RuleML to Version 1.02 in order to integrate a subset of features from Deliberation and Reaction RuleML into a new family of languages called Consumer RuleML, designed to be embedded into other languages such as LegalRuleML.

1 Introduction

RuleML[4] is a knowledge representation architecture designed for the interchange of the major kinds of Web rules in an XML format that is uniform across various rule logics and platforms. It has broad coverage and is specified as a system of extensible language families, whose modular definition of schemas permits rule interchange with high precision. Realizing the "overarching" design in [BPS10], RuleML 1.0 spans the complementary families of Deliberation RuleML 1.0 [5] and Reaction RuleML 1.0 [6] [Pas14].

Deliberation RuleML 1.01 increases the resolution of the language lattice of this family with refined language options, e.g. for Datalog$^+$ and Hornlog$^+$. There is no corresponding Version 1.01 release of Reaction RuleML.

The most recent release of RuleML, Version 1.02, is surveyed in the following sections. In Sect. 2, we describe the three familes of RuleML 1.02 – Deliberation, Reaction, and Consumer RuleML – and present an example. In Sect. 3, we explain the backward compatibility of this release. In Sect. 4, the roadmap of future RuleML releases is discussed, and we conclude with Sect. 5.

[4] RuleML Home: http://ruleml.org
[5] Specification of Deliberation RuleML 1.0: http://deliberation.ruleml.org/spec
[6] Specification of Reaction RuleML 1.0: http://http://reaction.ruleml.org/1.0

2 RuleML 1.02

RuleML 1.02 encompasses updated versions of the existing Deliberation and Reaction families to Deliberation RuleML 1.02 [7] and Reaction RuleML 1.02 [8]. The highlight of the RuleML 1.02 release is the new Consumer RuleML[9] family, an initial integration of Deliberation and Reaction RuleML, as shown in Fig. 1.

2.1 Consumer RuleML 1.02

Consumer RuleML obtained its name from being considered a "consumer" in two respects.

On one hand, Consumer RuleML is designed to be embedded into other languages. In particular, the absence of containers (e.g., <`Rulebase`>) and performatives (e.g., <`Assert`>, <`Query`>) means Consumer RuleML is a consumer, for structure and pragmatics, of containers and performatives from other languages, e.g. from FIPA, SOAP, or domain-specific XML languages such as Legal-RuleML [ABG+13].

On the other hand, Consumer RuleML consumes external resources, from other syntactic and semantic specifications, accessed by attributes linking to external definitions of syntactic constraints (`@type`), semantic profiles (`@style`) and extended quantifications (`@closure`). The external resources may be, for example, definitions in controlled natural language (e.g., in mathematical English) or formal specifications expressed in Reaction RuleML.

The following example shows a Consumer RuleML 1.02 rule that uses the specialized temporal syntax of Reaction RuleML to support an inference about the temporal scope of German climate data in dbpedia.[10]

```
<fipa-message act="inform">
  <sender>
    <agent-identifier>
      <name id="RuleML"/>
    </agent-identifier>
  </sender>
  <receiver>
    <agent-identifier>
      <name id="Tara-Athan"/>
    </agent-identifier>
  </receiver>
  <content xmlns:ruleml="http://ruleml.org/spec"
    xmlns:rs="http://ruleml.org/1.02/profiles/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

[7] Specification of Deliberation RuleML 1.02: `http://deliberation.ruleml.org/1.02/`

[8] Specification of Reaction RuleML 1.02: `http://reaction.ruleml.org/1.02/`

[9] Specification of Consumer RuleML 1.02: `http://consumer.ruleml.org/1.02/`

[10] `http://dbpedia.org`

```
xmlns:agg="http://aggregation-ontology.org/"
xmlns:clim="http://climate-ontology.org/">
<!-- The Consumer RuleML content starts here.-->
<ruleml:Rule closure="universal" style="rs:psoa">
  <ruleml:if>
    <ruleml:Atom>
      <!-- A PSOA frame representation of January climate data
         for some location "location" with a value of "temp"
          using the http://dbpedia.org vocabulary.-->
      <ruleml:oid>
        <ruleml:Var>location</ruleml:Var>
      </ruleml:oid>
      <ruleml:Rel iri="http://dbpedia.org/ontology/Place"/>
      <ruleml:slot>
        <ruleml:Ind iri="http://dbpedia.org/property/janMeanC"/>
        <ruleml:Var>temp</ruleml:Var>
      </ruleml:slot>
      <ruleml:slot>
        <ruleml:Ind iri="http://dbpedia.org/ontology/country"/>
        <ruleml:Ind iri="http://dbpedia.org/resource/Germany"/>
      </ruleml:slot>
    </ruleml:Atom>
  </ruleml:if>
  <ruleml:then>
    <ruleml:Operation type="agg:WindowedRecurring" style="agg:profile">
      <!-- A complex temporal modal operation specifying an aggregation model.
        An assumption is made that the dbpedia information is in agreement
        with the most recent climate data from the offical national source.-->
      <ruleml:Atom>
        <!-- A formula representing a monthly climate datum
          (temp at location).-->
        <ruleml:Rel iri="clim:climateMeanSurfaceAirTemperatureCelsius"/>
        <ruleml:Var>location</ruleml:Var>
        <ruleml:Var>temp</ruleml:Var>
      </ruleml:Atom>
      <ruleml:Interval>
        <!-- This argument of the modal operation brings in the additional
          knowledge that the most recent averaging period for German climate
          data is 1981-2010. This knowledge is not directly available from
          dbpedia, but is from the Deutscher Wetterdienst, which is the
          provenance of most Wikipedia climate data for Germany.
        -->
        <ruleml:Data xsi:type="xs:gYear">1981</ruleml:Data>
        <ruleml:Data xsi:type="xs:gYear">2010</ruleml:Data>
      </ruleml:Interval>
      <ruleml:Time>
        <!-- The month of the datum is January, expressed in the xs:gMonth
        datatype. -->
        <ruleml:Data xsi:type="xs:gMonth">--01</ruleml:Data>
      </ruleml:Time>
```

```
          <!--
            This rule has validity until around 2020, when the
            next climatological dataset should become available.
            Such a qualification could have been explicitly represented
            using Reaction RuleML's "qualification" feature, but this
            has not yet been included in Consumer RuleML.
          -->
        </ruleml:Operation>
      </ruleml:then>
    </ruleml:Rule>
  </content>
</fipa-message>
```

The premise of this rule would unify with the following fact, derived from dbpedia. While the fact's `elevation` slot has no counterpart in the rule premise, this is taken care of by the "look-in"/"slotribution" semantics [Bol15] specified via `"rs:psoa"`.

```
<fipa-message act="inform">
  <sender>
    <agent-identifier>
      <name id="dbpedia"/>
    </agent-identifier>
  </sender>
  <receiver>
    <agent-identifier>
      <name id="Tara-Athan"/>
    </agent-identifier>
  </receiver>
  <content xmlns:ruleml="http://ruleml.org/spec"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:rs="http://ruleml.org/1.02/profiles/">
    <!-- The Consumer RuleML content starts here.-->
    <ruleml:Atom  style="rs:psoa">
      <!-- The source of this fact is
        http://dbpedia.org/page/Berlin .-->
      <ruleml:oid>
        <ruleml:Ind iri="http://dbpedia.org/resource/Berlin"/>
      </ruleml:oid>
      <ruleml:Rel iri="http://dbpedia.org/ontology/Place"/>
      <ruleml:slot>
        <ruleml:Ind iri="http://dbpedia.org/ontology/elevation"/>
        <ruleml:Data xsi:type="xs:double">34.000000</ruleml:Data>
      </ruleml:slot>
      <ruleml:slot>
        <ruleml:Ind iri="http://dbpedia.org/property/janMeanC"/>
        <ruleml:Data xsi:type="xs:double">0.500000</ruleml:Data>
      </ruleml:slot>
      <ruleml:slot>
        <ruleml:Ind iri="http://dbpedia.org/ontology/country"/>
```

```
        <ruleml:Ind iri="http://dbpedia.org/resource/Germany"/>
      </ruleml:slot>
    </ruleml:Atom>
  </content>
</fipa-message>
```

As shown in Fig. 1, while Consumer RuleML 1.02 introduces no new language features, it does integrate some features that are not held in common by Reaction and Deliberation RuleML, and thus a particular valid Consumer RuleML fragment may not validate against either Reaction or Deliberation RuleML schemas.

2.2 Deliberation RuleML 1.02

The Deliberation RuleML modular schema design was created to allow additions to the language lattice (while maintaining backward compatibility). Sublanguages in the Deliberation RuleML language lattice are defined by a set of schema modules, and are identified by a myng-code, a compact representation of the MYNG REST query[11], which may be used to obtain the driver schema that includes the defining set of schema modules into one schema. The myng-code is composed of ten components, representing collections of modules that have some common characteristics. Each myng-code component corresponds to a REST query parameter. For example, the supremum language of the lattice, containing all other Deliberation RuleML 1.02 sublanguages, is called

```
myng-b3f-d7-a7-l1-p3ff-i7f-tf3f-q7-ef-s4f
```

and its Relax NG schema is available from[12]

```
http://deliberation.ruleml.org/1.02/
          myng-b3f-d7-a7-l1-p3ff-i7f-tf3f-q7-ef-s4f.rnc
```

which is redirected to the REST call

```
http://deliberation.ruleml.org/1.02/relaxng/schema_rnc.php?
        backbone=x3f&
        default=x7&
        termseq=x7&
        lng=x1&
        propo=x3ff&
        implies=x7f&
        terms=xf3f&
        quant=x7&
        expr=xf&
        serial=x4f
```

[11] The PHP script at `http://deliberation.ruleml.org/1.02/relaxng/schema_rnc.php` implements the MYNG REST interface to the parameterized schema for Deliberation RuleML 1.02.

[12] `.rnc` is the filename extension for Relax NG

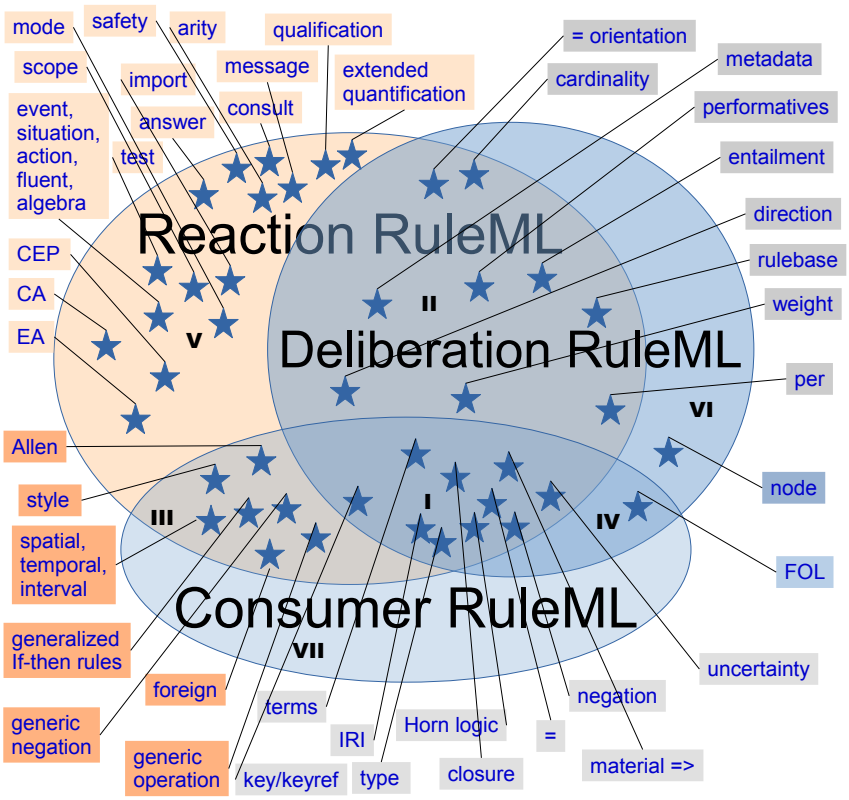

Fig. 1. The syntactic containment relationships between the Reaction, Deliberation and Consumer families are shown in this Venn diagram. The filled ellipses indicate sets of language constructs. The stars indicate language features, e.g. rules, first-order logic expressivity, or performatives, that may be combined to generate language constructs. The diagram indicates that there are no language features exclusive to Consumer RuleML – these would appear in region VII. However, because Consumer RuleML integrates some features that are not held in common by Reaction and Deliberation RuleML (regions III and IV), there are constructs (in region VII) in Consumer RuleML that are neither Deliberation nor Reaction RuleML constructs.

With the exception of the `d` component (`default` query parameter), each myng-code component corresponds to a GUI facet. In earlier versions, the `d` component served to configure options regarding attributes with default values, in the sense of XSD [AB14]. In RuleML 1.02, attributes no longer have syntactic default values[13], so this facet of the MYNG GUI has been omitted in Version 1.02. However, it is retained in the myng-code, with a restricted range of values, for backward compatibility.

2.3 MYNG GUI

Deliberation RuleML 1.02 is equipped with an improved MYNG GUI 1.02 for configuring a Deliberation sublanguage with a set of desired features. The dynamic output of the MYNG GUI (in Versions 1.01+) includes several components:

- The myng-code and possibly an anchor[14] of the form selection
- The MYNG REST interface URL for a Relax NG schema of the selected language
- An abbreviated URL for the Relax NG schema using the myng-code
- The URL for the XSD schema of an anchor
- A display of the generated Relax NG schema driver (on demand only)

Relax NG schemas configured using MYNG (online through the MYNG REST interface URL or as local copy after download) may be used outside of MYNG for schema-aware authoring, instance validation, or parser generation through XML tools such as oXygen XML Editor and JAXB. New MYNG GUI functionality has been added in Version 1.02, including

- Fill and Clear buttons for the form as a whole as well as expressivity-specific Fill/Clear and Check/Uncheck All for a number of facets

2.4 Reaction RuleML 1.02

Reaction RuleML is a standardized rule markup/serialization language and semantic interchange format for reaction rules and rule-based event processing [BPS10, Pas14, PB09]. Reaction rules include spatial/temporal Derivation Rules (DR), Knowledge Representation (KR) calculi / logics for temporal/event/action/situation reasoning, as well as Event-Condition-Action (ECA) rules, Production (CA)

[13] See `http://consumer.ruleml.org/1.02#Semantic_Variant_Attributes` for a mechanism of specifying *semantic default values* through semantic profiles.

[14] A few Deliberation RuleML sublanguages of special significance have been designated as **anchor** sublanguages (or simply anchors); XSD schemas (automatically generated in advance by offline scripts) are made available for anchors. An *anchor of a Deliberation RuleML sublanguage* is a minimal Deliberation RuleML anchor that contains the sublanguage. The anchor of myng-b3f-d7-a7-l1-p3ff-i7f-tf3f-q7-ef-s4f is naffologeq.

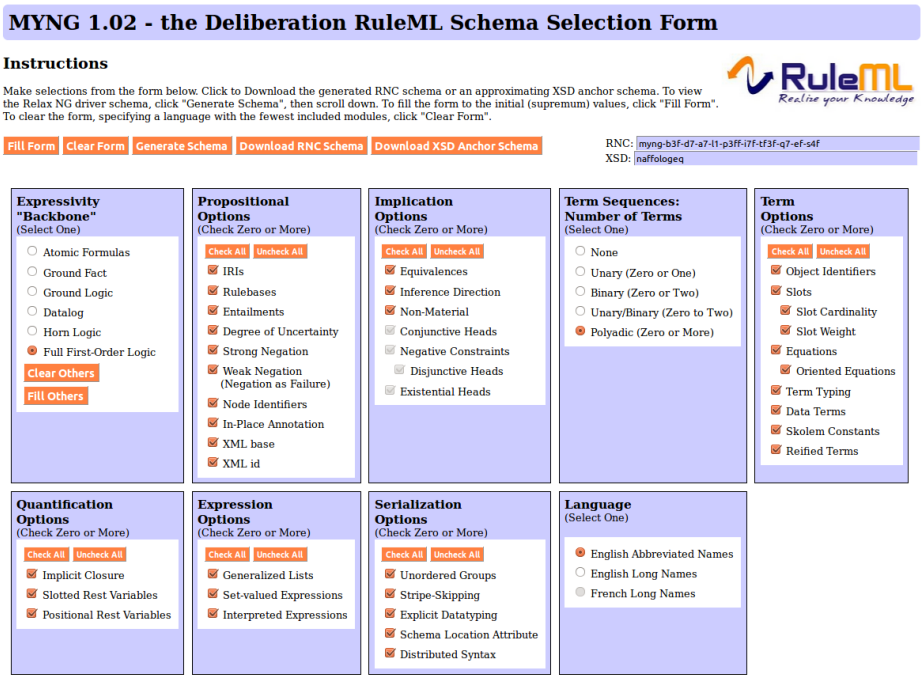

Fig. 2. The top portion of the MYNG GUI includes brief instructions, a set of buttons for managing the form and downloading schemas, and text fields displaying the myng-code and anchor. The figure shows the two rows of GUI facets. Not shown in this screenshot are the schema URLs and the Relax NG driver schema, which are displayed below the second row of GUI facets, in addition to some more detailed instructions on usage of the schema.

rules, and Trigger (EA) rules, and distributed rule-based Complex Event Processing (CEP). Reaction RuleML 1.02 incorporates this reactive spectrum of rules into RuleML, employing a system of step-wise extensions of the Deliberation RuleML 1.02 foundation.

2.5 Complete Feature Set of RuleML 1.02

In Fig. 1, the overlapping feature sets of the three families create seven disjoint feature regions, which will be unpacked below. The union of those sets constitutes the complete feature set of RuleML 1.02.

- Region I (Deliberation, Reaction, and Consumer) Deliberation, Reaction, and Consumer RuleML 1.02 have a number of features in common:
  - Terms, including individuals, data terms, variables, skolem constants, reified terms, generalized lists, and functional expressions
  - Atomic formulas, including polyadic positional arguments, slots, rest variables, and object identifiers
  - Horn logic expressivity, including logical connectives and quantification over variables
  - Equations
  - Negations, including classical (strong) and negation-as-failure (weak) forms, with, e.g., well-founded or answer-set semantics
  - Degree of uncertainty modifiers, which may be used, e.g., for fuzzy or probabilistic logics
  - Type modifiers, which may be used, e.g., for sorted logics or as an abbreviation for classification through unary predicates
  - Material or non-material implications
  - IRIs as constants
  - Implicit quantification (closure), including reference to externally defined extended quantifiers such as numerical quantification
  - Mechanism for distributed definitions (`@key` and `@keyref`)
- Region II (Reaction and Deliberation) Some features common to Reaction and Deliberation RuleML 1.02 were intentionally kept out of Consumer RuleML 1.02 in order to minimize its complexity and maintain Consumer RuleML's focus on semantics over pragmatics:
  - Reasoning direction modifiers
  - Metadata annotations
  - Basic performatives (Assert, Retract, Query)
  - Rulebase containers
  - Metalogical entailment assertions
  - Equation orientation modifiers
  - Functional expression interpretation modifiers (per), which may be used, e.g., to specify that a built-in functional expression should be replaced by its value
  - Cardinality modifiers, which may be used, e.g., to specify that a functional expression or slot has a certain multiplicity of its values

  - Slot weights
- Region III (Reaction and Consumer) Some advanced features from Reaction RuleML have been made available in Consumer RuleML 1.02, although they have not (yet) been incorporated into Deliberation RuleML:
  - Rules (<`Rule`>) generalizing <`Implies`>), which may be used, e.g., for defeasible rules
  - Specialized syntax dedicated to spatial, temporal, and interval terms and formulas
  - Allen interval operators
  - Generic negations
  - Generic operations, e.g. for modal logic
  - Syntax for indicating the semantic profile (`@style`)
  - Extensibility through foreign namespaces (content)
- Region IV (Deliberation and Consumer) Consumer RuleML 1.02 follows Deliberation RuleML in regards to expressivity:
  - First-order logic expressivity
- Region V (Reaction only) Reaction RuleML 1.02 contains advanced features that are not available in either Deliberation or Consumer RuleML:
  - Event, situation, action, and fluent terms, formulas, and algebra operators
  - Production (CA) rules
  - Trigger (EA) and Event-Condition-Action (ECA) rules
  - Distributed rule-based Complex Event Processing (CEP)
  - Actions/performatives for semantic import/consult, updates, answers, VVI tests
  - Message descriptors
  - Interface definitions and attributes for
    * scope,
    * mode
    * safety
    * arity
    * cardinality
    * qualifications
    * quantification
- Region VI (Deliberation only) Deliberation RuleML 1.02 has one feature that is not contained in either Reaction or Consumer RuleML:
  - IRI labeling denoting abstract propositions and rules (`@node`)

3 Backward Compatibility

Deliberation RuleML 1.02 is largely backward compatible with Version 1.01, with the exception of the following:

– The content model of attributes whose values may denote Web links (i.e. `@iri`, `@node`, and `@type`) no longer includes relative IRI references. Instead, the CURIE syntax is used to abbreviate IRIs. The upgrader XSLT transformation[15] to Deliberation RuleML 1.02 expands relative IRI references to IRIs.

Reaction RuleML 1.02 is also largely backward compatible with Version 1.0, with the exception of the following:

– Operator has been renamed to Operation
– truth-valued generics and operations are only used as formulas, but are no longer allowed as functional terms
– act edge has been renamed and generalized to a do edge interpreting performatives uniformly as (knowledge) actions

Consumer RuleML 1.02 is the first version of this family of RuleML sublanguages, and so has no backward compatibility properties.

4 Roadmap of RuleML 1.02 Implementation and Development of RuleML 1.03 and Beyond

The RuleML Wiki hosts the RuleML issue tracking system[16], which is used to manage the development of the RuleML language. Users are encouraged to post bug reports and enhancement requests. Only a selection of enhancements will thus be given here.

In **Deliberation RuleML** Version 1.03 and beyond, we expect to see the following enhancements.

– Incorporate declarative features developed in Reaction RuleML, including spatial, temporal, and interval terms and formulas, semantic profiles, Allen operators and modal operators (or, moving them into a common 'root' family from where they can be inherited by the specific families).
– Generalize the (RNC and XSD) schemas, MYNG, and the formatters (normalizers and compactifiers) according to the PSOA syntax and semantics in a manner combinable with the current POSL syntax and semantics [Bol15].
– Work upward, beyond first-order logic, in the deliberation expressivity hierarchy [BPS10], providing schemas etc. for higher-order and modal logics.

Of highest priority in the development of **Reaction RuleML** is the update of the translator service framework [PBZ+12] to Reaction RuleML 1.02 so that it can be used, e.g., in Rule Responder [PB14, ZTP+12]. The initial goal is to have the Prova translators updated for 1.02, followed by JAXB translators, e.g. into Drools and other rule engine languages. Furthermore, the development of semantic profiles for reaction rules, events, and actions [Pas14], semantic profiles

[15] RuleML Upgraders: `http://deliberation.ruleml.org/1.02/xslt/upgrader/`
[16] `http://wiki.ruleml.org/index.php/Category:Issues`

for modularization, distribution, and imports [PB14], as well as verification, validation, and integrity testing [Pas05]. Support for generalized RNC schema, MYNG, and formatters are in the focus of Reaction RuleML 1.03.

In **Consumer RuleML** Version 1.03 and beyond, we expect to see the following enhancements.

- Broaden the coverage of Deliberation and Reaction RuleML.

In MYNG Version 1.03 and beyond, we expect to see the following enhancements.

- The MYNG coverage should be extended to Reaction and Consumer RuleML.
- New anchor languages should be added, based on user suggestions.
- The MYNG functionality should be improved, including:
  - Provide a visual indicator of the (lattice) distance from an anchor to the RuleML sublanguage selected by GUI options, especially the zero-distance case of exact match between an anchor and the selected options.
  - Implement the inverse functionality of determining the options corresponding to a given anchor or myng-coded language selected by the user.
- Progress will continue towards making MYNG a *language-lattice development environment*:
  - Add validation of an instance against the configured Relax NG schema, using, e.g., Validator.nu.[17]
  - Develop automatic determination of the lub schema for a valid instance of the supremum language, performing test validation against the *penultimate* Relax NG schemas containing all modules but one, and exploiting (monotonicity) properties of the language lattice.[18]
  - Add Schematron validation, e.g. using Validator.nu, to impose additional language restrictions, such as the 'minus' restrictions of Datalog$^{\pm}$, which restore decidability lost in Datalog^{+}.
  - Enable on-the-fly conversion of Relax NG schemas, configured using MYNG, into monolothic XSD schemas.

5 Conclusions

RuleML 1.02 has been specified in terms of its families of languages – Deliberation, Reaction, and Consumer RuleML. Consumer RuleML is seen to be a nexus of integration between Deliberation and Reaction RuleML, paving the way for an expansion of their common core of syntactic features. We expect that feedback from the further integration of Consumer RuleML into other languages, such as LegalRuleML, will foster additional development of the RuleML architecture.

[17] `http://validator.nu`
[18] `http://wiki.ruleml.org/index.php/MYNG_Checker`

References

AB14. Tara Athan and Harold Boley. The MYNG 1.01 Suite for Deliberation RuleML 1.01: Taming the Language Lattice. In *Rule Challenge @ RuleML 2014*, volume 1296 of *CEUR*, page 14. CEUR-WS.org/Vol-1296/, 2014.

ABG+13. Tara Athan, Harold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. OASIS LegalRuleML. In Enrico Francesconi and Bart Verheij, editors, *International Conference on Artificial Intelligence and Law (ICAIL '13), Rome, Italy, June 10-14*, pages 3–12. ACM, 2013.

Bol15. Harold Boley. PSOA RuleML: Integrated Object-Relational Data and Rules. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules (RuleML 2015) - 11th International Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*. Springer, 2015.

BPS10. Harold Boley, Adrian Paschke, and Omair Shafiq. RuleML 1.0: The Overarching Specification of Web Rules. In *Proc. 4th International Web Rule Symposium: Research Based and Industry Focused (RuleML-2010), Washington, DC, USA, October 2010*, Lecture Notes in Computer Science. Springer, 2010.

Pas05. Adrian Paschke. The ContractLog Approach Towards Test-driven Verification and Validation of Rule Bases - A Homogeneous Integration of Test Cases and Integrity Constraints into Evolving Logic Programs and Rule Markup Languages (RuleML) . *IBIS*, 10, 2005.

Pas14. Adrian Paschke. Reaction ruleml 1.0 for rules, events and actions in semantic complex event processing. In *Rules on the Web. From Theory to Applications - 8th International Symposium, RuleML 2014, Co-located with the 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, August 18-20, 2014. Proceedings*, pages 1–21, 2014.

PB09. Adrian Paschke and Harold Boley. Rules Capturing Events and Reactivity. In Adrian Giurca, Dragan Gasevic, and Kuldar Taveter, editors, *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*, pages 215–252. IGI Publishing, May 2009.

PB14. Adrian Paschke and Harold Boley. Distributed rule-based agents with rule responder and reaction ruleml 1.0. In *Proceedings of the RuleML 2014 Challenge and the RuleML 2014 Doctoral Consortium hosted by the 8th International Web Rule Symposium, Challenge+DC@RuleML 2014, Prague, Czech Republic, August 18-20, 2014.*, 2014.

PBZ+12. Adrian Paschke, Harold Boley, Zhili Zhao, Kia Teymourian, and Tara Athan. Reaction RuleML 1.0: Standardized Semantic Reaction Rules. In Antonis Bikakis and Adrian Giurca, editors, *Rules on the Web: Research and Applications*, volume 7438 of *Lecture Notes in Computer Science*, pages 100–119. Springer Berlin Heidelberg, January 2012.

ZTP+12. Zhili Zhao, Kia Teymourian, Adrian Paschke, Harold Boley, and Tara Athan. Loosely-Coupled and Event-Messaged Interactions with Reaction RuleML 1.0 in Rule Responder. In *RuleML (2)*, 2012.