

WHY WE NEED AN XML STANDARD FOR REPRESENTING BUSINESS RULES

Christian de Sainte Marie – ILOG

Introduction

We are interested in the topic of communicating policy decisions to other parties, and, more generally, to communicating and sharing business rules and policies.

The usual acceptation is that business rules are atomic and highly specific and structured statements that constrain some aspect of a business, control or influence its behaviour.

If we replace “business” by “Web service” in the definition, we see that it overlaps broadly with the definition of constraints and capabilities for Web services. This is not surprising, as Web services are meant to be part of business processes, and their capabilities and use are therefore constrained by the process and the service owners’ business policies.

Business rules are therefore an important aspect of Web Services. However, they are also the missing link in the protocol stacks: the based on a common standard for exchanging data and data models (XML and XML Schema), the Semantic Web stack defines standards for defining and exchanging the semantic of data, whereas the Web Service stack defines standards for defining and exchanging service models and, to a degree the semantics of the services (WSDL, BPEL4WS, etc) along with the protocol required for acquiring the services (SOAP, WS-I profiles, but also all the protocols related to security etc).

But there is no specific stack identified for the standards needed to define and exchange the business logic and policies that specifies the conditions under which specific data make sense and/or specific services are offered.

Production rules

When they are meant to be directly enforced by an automated system – as it is the case for business rules that constrain or describe a Web service –, business rules are often represented as production rules: a *production rule* is an independent statement of programming logic that specifies the execution of one or more actions in the case that its conditions are satisfied; it is usually defined in the context of being executable by an inference engine.

We believe that the representation and interchange of business policies and rules in the form of production rules is an important requirement for the deployment and widespread adoption of Web Services, as evidenced by the number of relevant standards that require the use of business rules (WS-Policy, EPAL etc, to name a few in the domain of access and security).

The semantic content of the business rules (the business vocabularies) is domain specific, but there are no reasons why the syntax for sharing the rules should not be common. The current lack of an authoritative, widely accepted standard for sharing business rules and policies is damaging in the sense that each particular domain- or function-specific standard tends to define its own syntax.

However, inference and rules are usually considered more directly relevant to the Semantic Web than to the Web Service Architecture. As a consequence, the

standardisation activities is mostly focused on logical rules, and production rules have received little attention in the Semantic Web or the Web Services community: *logical rules* – or inference rules – are concerned with representing the relationship between the truth values of logical statements, whereas production rules – or reaction rules – are concerned with the representation of behaviours. They allow the specification of the reactive and communicative behaviour of a system.

Production rules are the most convenient way to represent most business rules, as well as the most widely supported by existing inference engines, including the main commercial ones.

For instance, with respect to the proposed use case, the Web service behaviour in reply to a request message depending on the message's characteristics regarding messaging protocol reliability, header and type of information disclosed can easily be specified, and thus exposed, in the form of production rules. The exposed rules could be, either directly the rules that control the behaviour of the Web service, or, reversely, the rules to be applied by the requester to obtain the expected behaviour.

Conclusion

Our position is that an XML standard for sharing and communicating business rules is necessary, that it would be, in particular, useful for Web services to scope and to share policies and to communicate policy decisions; and that it should preferably take the form of a light weight syntax for communicating production rules.

Appendix: Related proposals and standardisation efforts

Numerous standard proposals – in particular in the realm of Web Services – include the definition or representation of business rules for their specific purposes. Although this shows how pervasive business rules are, there is no generally accepted XML-based language for sharing and communication business rules or production rules. In this section, we give a short description of some of the most relevant proposals, as well as related standardisation effort at the OMG and other groups.

Business Rules Markup Language (BRML)

The Business Rules Markup Language (BRML) was developed at IBM, in connection with IBM's Business Rules for E-Commerce Project. It is presented as an “XML Rule Interlingua for Agent Communication, based on Courteous/Ordinary Logic Programs.” It is used in connection with 'CommonRules' from IBM.

BRML complements and extends ANSI-draft Knowledge Interchange Format (KIF), and provides the first XML encoding for KIF. In version 1.0, only a broad sub-case of KIF is represented: clauses. BRML goes beyond KIF to support logical non-monotonicity, including the negation-as-failure, the most practically important form of negation, and prioritized conflict handling. CommonRules includes a sample translator to/from KIF's existing (non-XML) string format.

Business Semantics for Business Rules (OMG BSBR)

There is no generally accepted approach for defining or modelling business rules. The OMG issued a Request for Proposals (RFP) with the objective to allow business people to define the policies and rules by which they run their business in their own language, in terms of the things they deal with in the business, and to capture those rules in a way

that is clear, unambiguous and readily translatable into other representations. Among those representations are presentation forms for business people and software engineers, and executable rules for many kinds of automated systems. This RFP solicits proposals for the following:

- a metamodel for the specification of business rules by business people, with a MOF representation;
- a metamodel for the capture of vocabularies and definitions of the terms used in business rules;
- an XML representation of business rules and vocabularies based on XMI that permits exchange among software tools that manage, display, use, and translate business rules

This effort is conducted within the Business Enterprise Integration Domain Task Force (DTF), which regroups the Business Modelling activities and the Business Rules working group. Initial draft proposals are currently being submitted to OMG in answers to this RFP (deadline for revised submission: October 10, 2004). According to its procedure and standard times, the OMG can be expected to approve and issue a BSBR standard meta-model within one year from initial submission.

A Java runtime API for Rule Engines (JSR-094)

Developed within the Java Community Process, the JSR-094¹ defines a Java runtime API for rule engines targeting both the J2EE and J2SE platforms. The API prescribes an object model and a set of fundamental rule engine operations based upon the assumption that most clients will need to be able to execute a basic multi-step rule engine cycle.

We mention JSR-094 because, although it defines generic API support for parsing rulesets that are represented using XML, it does not define a rule language standard. The widespread adoption of the API thus depends on the widespread adoption of a common XML language for rules representation.

The JSR-094 has been adopted last year after final public review.

Production Rules Representation (OMG PRR)

Through its Business Enterprise Integration DTF, the OMG issued a RFP that addresses the representation of production rules in UML model. With respect of production rules, this RFP solicits proposals for the following:

- A MOF2 compliant metamodel with precise dynamic semantics to represent production rules, where "production rules" refers to rules that are executed by an inference engine. This metamodel is intended to support a language that can be used with UML models for explicitly representing production rules as visible, separate and primary model elements in UML models;
- An XMI W3C XML Schema Description (xsd) for production rules, based on the proposed metamodel, in order to support the exchange of production rules between modeling tools and inference engines;
- An example of a syntax that is compliant with the proposed metamodel for expressing production rules in UML models. This syntax will be considered non-normative.

¹ JSR stands for "Java Specification Request."

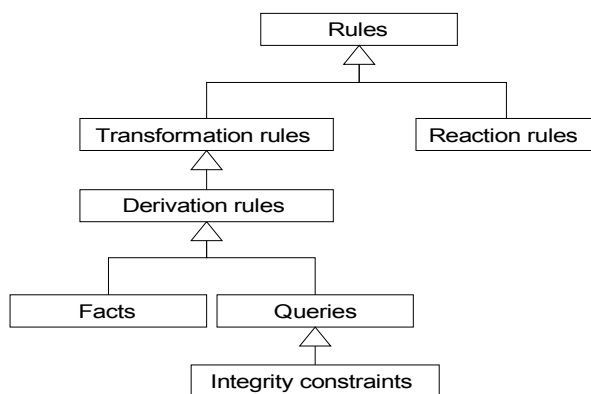
Initial draft proposals in answers to this RFP have been submitted on August 2, 2004. According to its procedure and standard times, the OMG can be expected to approve and issue a PRR standard meta-model within one year from initial submission.

Rule Markup Language (RuleML)

The Rule Markup Initiative has taken steps towards defining a shared Rule Markup Language (RuleML), permitting both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential-transformational tasks. The initiative started during PRICAI 2000 and was launched on the Internet in November 2000. The goal of the Rule Markup Initiative is to develop RuleML as the canonical Web language for rules using XML markup, formal semantics, and efficient implementations. The participants of the initiative constitute an open network of individuals and groups from both industry and academia.

RuleML covers the entire rule spectrum, from derivation rules to transformation rules to reaction rules. RuleML can thus specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviors of workflows, services, and agents.

Besides the initial XML-only RuleML and the current XML/RDF-combining RuleML, there is also work on an approach towards an RDF-only RuleML. Recent efforts also went into defining MOF-RuleML, an abstract syntax of RuleML as an OMG's Meta-Object Facility (MOF) model. The RuleML group also contributes to the definition of the BSBR and PRR meta-models within the OMG.



The current RuleML design RuleML encompasses a hierarchy of rules, including reaction rules (event-condition-action rules), transformation rules (functional-equational rules), derivation rules (implicational-inference rules), also specialized to facts ('premiseless' derivation rules) and queries ('conclusionless' derivation rules), as well as integrity-constraints (consistency-maintenance rules) (cf.

figure left).

However, until now, RuleML has been mostly involved with the right branch of the tree, defining in particular derivation rules, facts, and queries.

Simple Rule Markup Language (SRML)

ILOG contributed an open, XML-based rule language to the community by registering the schema on xml.org. Simple Rule Markup Language (SRML) describes a generic rule language consisting of the subset of language constructs common to the popular forward-chaining rule engines. Because it does not use constructs specific to a proprietary vendor language, rules specified using this DTD can easily be translated and executed on any conforming rule engine, making it useful as an interlingua for rule exchange between Java rule engines. The ruleset is the root element of the SRML XML document, and it consists of a list of rules. Rules have a condition part and an action part, and the condition part must have at least one condition. Conditions are composed of test expressions, and can be simple conditions or not conditions. Simple conditions

can be bound to variables while not conditions cannot. The action part of a rule consists of actions, which can be variable declarations and assignments, as well as the traditional assert, retract and modify statements of rule languages. The assert action adds an object to working memory. The retract action removes an object from working memory. The modify action modifies an object in working memory. Expressions appear throughout the language and can be assignable (variables or fields), constants (literals such as strings, ints, floats, booleans, etc.), arithmetic or boolean expressions.

Semantic Web Rule Language (SWRL)

The member submission of a proposal for a Semantic Web Rule Language (SWRL) has been recently acknowledged by the W3C . The proposal is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language (and above). It extends the set of OWL axioms to include Horn-like rules. It thus enables Horn-like rules to be combined with an OWL knowledge base. SWRL provides a high-level abstract syntax that extends the OWL abstract syntax described in the OWL Semantics and Abstract Syntax document . An extension of the OWL model-theoretic semantics is also given to provide a formal meaning for OWL ontologies including rules written in this abstract syntax.

The proposed rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

Both the antecedent (body) and consequent (head) consist of zero or more atoms. An empty antecedent is treated as trivially true (i.e. satisfied by every interpretation), so the consequent must also be satisfied by every interpretation; an empty consequent is treated as trivially false (i.e., not satisfied by any interpretation), so the antecedent must also not be satisfied by any interpretation. Multiple atoms are treated as a conjunction.

SWRL specifies, for these rules, an XML syntax based on RuleML and the OWL XML presentation syntax, as well as an RDF concrete syntax based on the OWL RDF/XML exchange syntax.

References

- [1] <http://www.alphaworks.ibm.com/tech/commonrules>
- [2] http://java.sun.com/aboutJava/communityprocess/jsr/jsr_094_ruleengine.html
- [3] <http://www.ruleml.org>
- [4] <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
- [5] <http://www.w3.org/TR/owl-semantics/>