

Seven Golden Rules for a Web Rule Language

Gerd Wagner, *Eindhoven University of Technology*

Yes, we do need a Web rule language. Rules and rule markup languages, such as RuleML, will play an important role for the success of the Semantic Web. Rule expressions will be used in Web applications for defining derived terms on the basis of a taxonomy, for specifying validation constraints, for representing organizational policies and business rules, for specifying the behavior of a software agent, and for many other purposes.¹ Rule markup languages will be the vehicle for using rules on the Web. They will allow to deploy, execute, publish and communicate rules on the Web, and they will also serve as a lingua franca for exchanging rules between different systems and tools.

In a narrow sense, a Web rule language is a concrete (XML-based) rule syntax for the Web. In a broader sense, it should have an abstract syntax as a common basis for defining various concrete sublanguages serving different purposes.

RuleML, in its current version 0.8, is not sufficient as a general Web rule language. With some syntactic simplifications, however, it is a good starting point.

A Web rule language should have a formal semantics. However, there may be language constructs, which do not have a formal semantics based on classical first-order logic but which are needed to deal with certain practical problems. There are two dangers related to this trade-off, which should be avoided:

- adopting practical language constructs (such as “procedural attachments”), which seem to be important and have some intuitive, but no formal semantics, although there are alternatives, which have a formal (yet non-standard) semantics;
- paying too much attention to theoretical issues of standard first-order logic, such as computational (asymptotic worst-case) complexity, decidability and compactness.

A Web rule language standardization effort should pay special attention to the concerns of the users of SQL, Prolog and production rules (CLIPS, JESS, ILOG, etc.). It must allow to map the most important constructs of these languages.

My general advice to such an effort is summarized in the following seven golden rules (GR).

GR 1: Relational databases are more important than FOL

Many KR formalisms strictly (or blindly?) follow classical first-order logic and ignore the non-classical inference features and rule concepts, which have proven to be essential in relational databases, such as 3-valued connectives, nonmonotonic queries and (state-changing) *trigger* rules. A Web rule language cannot afford to ignore these fundamental information-processing concepts, which require abandoning classical logic.

GR 2: UML is more important than OWL

The UML represents a larger body of information and knowledge modeling experience and expertise than OWL does. The UML includes an expressive language (OCL) for integrity constraints. These constraint expressions form also a kind of rule (‘integrity rules’) and should be covered by a Web rule language. Remarkably, the UML does also provide more support for advanced ontological constructs: e.g., it supports *part-whole* relationships (with aggregation and composition) and *powertypes* as classes whose instances are subclasses of another class (BiologicalSpecies and PassengerAircraft are examples of powertypes).

GR 3: Rules are not implications

While an implication is an expression of a logical formula language, typically possessing a truth-value, a derivation rule does not possess a truth-value but has the role to generate derived sentences. There are logics, which do not have an implication connective, but which have a derivation rule concept. In standard logics (such as classical and intuitionistic logic), there is a close relationship between a derivation rule (also called “sequent”) and the corresponding implicational formula: they both have the same models. For nonmonotonic rules (e.g. with negation-as-failure) this is no longer the case: the intended models of such a rule are, in general, not

the same as the intended models of the corresponding implication.

GR 4: Web rules are not just Horn clauses

This golden rule is a corollary of GR 3. Since Horn clauses are a very limited type of implication, and rules are not implications (according to GR 3), it follows that Web rules are not just Horn clauses. Web rules are rule expressions used in Web documents and in Web applications. They have to be much more expressive than Horn clauses (see also <http://lists.w3.org/Archives/Public/www-rdf-rules/2001Sep/0079.html>).

GR 5: Web rules should be able to express database rules

In Web applications, we should expect similar uses of rules as in databases. This consideration suggests that a Web rule language has to accommodate

- *SQL assertions*: integrity rules
- *SQL views*: nonmonotonic derivation rules with 3-valued connectives and open and closed predicates
- *SQL triggers*: reaction rules, which are limited to update events

GR 6: A Web rule language should allow to express and implement *business rules*

Business rules refer to the hundreds, if not thousands, of policies, procedures and definitions that govern how a company operates and interacts with its customers and partners. Three basic types of business rules have been identified in the literature²:

- *Integrity rules*: assertions that must hold in all evolving states and state transition histories of an enterprise viewed as a discrete dynamic system. Example: “The driver of a rental car must be at least 25 years old”.
- *Derivation rules*: statements of knowledge that is derived from other knowledge by an inference or a mathematical calculation. Example: “A gold customer is a customer with more than \$1MM on deposit”.
- *Reaction rules*: expressions of policies specifying actions in response to events. Example: “When a share price drops by more than 5%, and the investment is exempt from profit tax, then sell it”.

GR 7: A Web rule language should allow multiple purposes, multiple languages and multiple semantics

The Web is a pluralistic world, no matter if it is semantic or not. There will be multiple purposes, multiple languages and multiple semantics for Web rules. The real challenge is to develop an integrated metamodel, or abstract syntax, which supports this plurality.

References

1. http://tmitwww.tm.tue.nl/staff/gwagner/myRuleML/What_is_a_rule.html
2. K. Taveter and G. Wagner, “Agent-Oriented Enterprise Modeling Based on Business Rules”, in Proc. of 20th Int. Conf. on Conceptual Modeling (ER2001), Springer-Verlag, LNCS 2224, pp. 527–540, 2001.