Rules in XML: The RuleML Initiative

Harold Boley DFKI http://www.dfki.uni-k.de/~boley
Benjamin Grosof MIT Sloan http://www.mit.edu/~bgrosof
Said Tabet Nisus Inc. http://people.ne.mediaone.net/stabet

http://www.dfki.uni-kl.de/ruleml

Background

(1st: See Flyer !!!!!!)

- Rules on the Web have become mainstream topic
  - **identified as a Design Issue of the Semantic Web**
- Rules for inferencing in:
  - **business rules, e-commerce, Agents, K-b systems, workflow, database queries and triggers**
- Rules for transformation in:
  - **(XML) document translation**

Overall Goals

- Provide a basis for a standardized rule markup approach, with declarative knowledge representation (KR) semantics
  - Aid integration of heterogeneous rule systems and applications, via shared rule markup language
  - Start with commercially important flavors of rules
- Start simple with a kernel KR, then add extensions incrementally.
- Become an industry standard (e.g. via W3C)

Technical Approach of RuleML

- Start with: Datalog Logic Programs with rules labeled *as kernel*
  - similar to Business Rules Markup Language *(IBM CommonRules)*
- Add extensions/restrictions, creating a family of DTD's organized as a generalization-specialization hierarchy (lattice)
  - URI's; RDF triple is a fact *(NB: other relnsh's to RDF too!)*
  - negation-as-failure (well-founded semantics); classical negation
  - prioritized conflict handling cf. Courteous Logic Programs
  - procedural attachments: actions, queries
  - logical functions; user-defined functions
  - 1st-order logic type expressiveness cf. Lloyd LP's, DAML+OIL, KIF
  - more: equivalence/rewriting rules; ... temporal, Bayesian, fuzzy, …
  - *define DTD's modularly, using XML entities (~macros)*

Webizing Rule KR

- URIs for logical vocabulary and knowledge subsets
- labels for rules/rulebases, import/export
- headers: meta-data describes doc's expressive class
- procedural attachments using Web protocols;
  queries or actions via CGI/servlets/SOAP/…

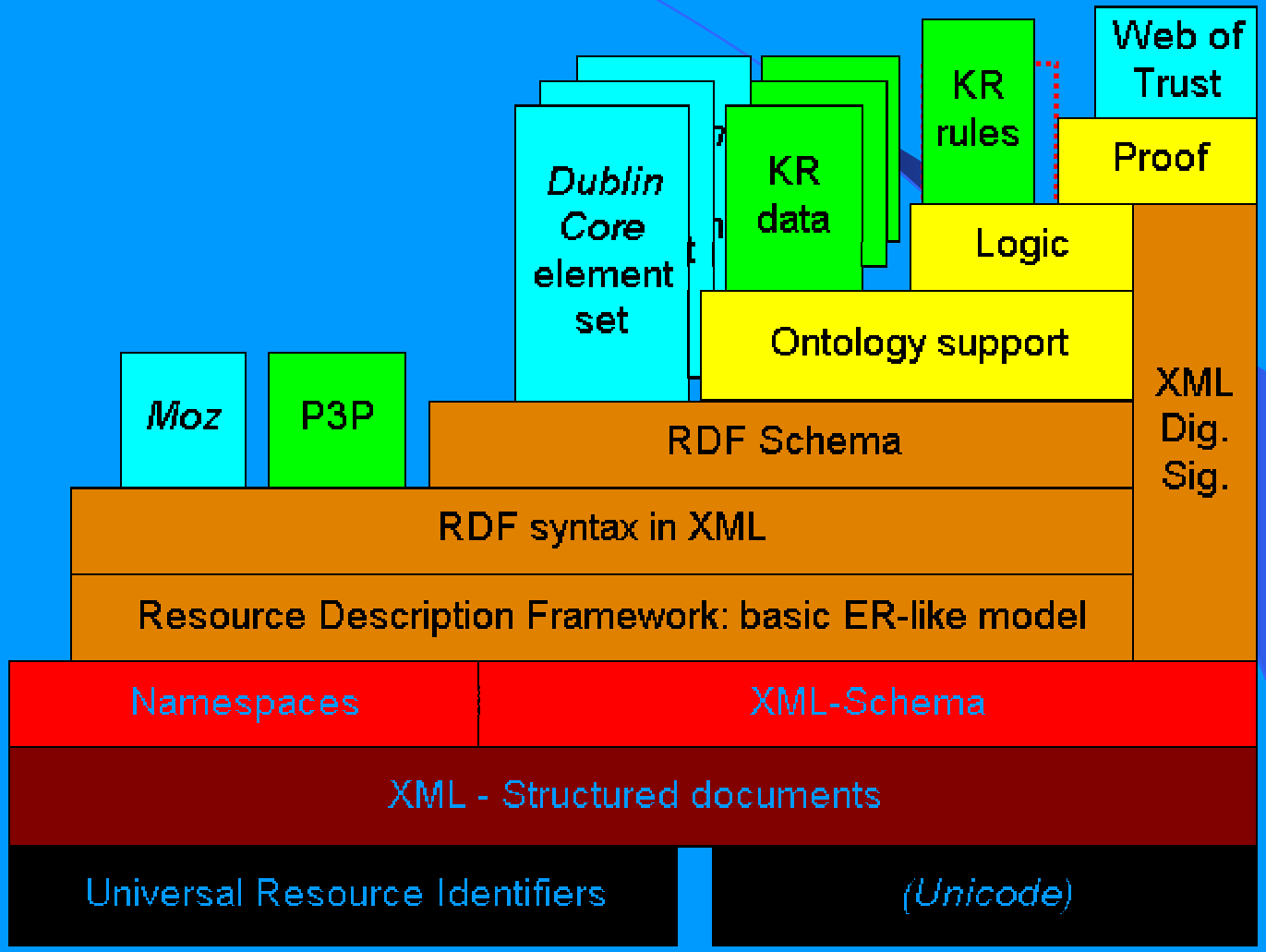
- *Other practical mechanics:*
  - build on existing W3C standards: namespaces, …
  - share mechanisms with RDF/RDFS, DAML+OIL
  - use ontologies for rules, and rules for ontologies
    - ontology tags in: rulebase, predicate symbol, …

Further Directions

- move to XML Schema based rather than DTD based
- additional XML syntaxes: RDF; surface/"style-sheeted"
- more KR's: KIF/classical, Notation 3, Bayesian, fuzzy, rewriting, temporal, …

- provide Rule mechanism to emerging W3C standards:
  - Semantic Web / RDF, P3P, …

Building the Future (TimBL)



Relevant Other Efforts in W3C and Markup

- RDF, RDFS, DAML(+OIL), Semantic Web
- P3P privacy policies: APPEL rules
- XML Query

- Others:
  - XSLT
  - MathML
  - Predictive Model Markup Language (rules from data mining)

Overview of Current Status

- Technical: Strawman: Initial DTD family V0.7 released 1/31/01
  - Datalog LP with URI's; some extensions/restrictions
  - Goal: give feel, start getting feedback
- Organizers: Harold Boley, Benjamin Grosof, Said Tabet
  - also authored the Strawman
- Participants: many interested; some actively giving feedback;
  - evolving towards more formal organization
- Website: http://www.dfki.de/ruleml

RuleML Participants

- Agent Frameworks (Leon Sterling, Department of Computer Science and Software Engineering, University of Melbourne, Australia)
- BRML/DAML-RULES (Benjamin Grosof, MIT Sloan School of Management, USA)
- Euler (Jos De Roo, AGFA, Belgium)
- Jess (Ernest Friedman-Hill, Distributed Systems Research, Sandia National Labs, USA)
- FLIP (Jose Hernandez-Orallo, DSIC, Politechnical University of Valencia, Spain)
- PDDL: Planning Domain Definition Language (Drew V. McDermott, Department of Computer Science, Yale University, USA)
- Protege-2000 (Mark Musen, Stanford Medical Informatics, USA)
- RBML: Rule Base Markup Language (Chris Roberts, Sun Microsystems, USA)
- RFML (Harold Boley, DFKI, Germany)
- URML (David Ash, Real Time Agents Inc.; Prabhakar Bhogaraju, MindBox; Said Tabet, Nisus; USA)
- XRML (Jae Kyu Lee, KAIST, Korea)
- Rules XML Schema (Carlos Morales, Blaze Software, USA)

RuleML Participants (Industry)

Rules Engine Providers	Other Technology Providers	Users
JESS, MindBox, WebMind, Ilog, Blaze, Allaire, Haley, …	Nokia, Ericsson, Phone.com, BEA, HP/Bluestone, IBM, SilverStream, ATG, Oracle, Broadvision, Blue Martini, …	go2Online, Tribune of Chicago, Advertising.com, Fidelity, Amazon, City Bank, Chase, …

Cooperation with Java Rule Engines Effort

One or more rule engines will be needed for executing RuleML modules. On 2000-11-15, the RuleML Initiative thus joined forces with the Java Specification Request JSR-000094 Java Rule Engine API.

This cooperation will enable a direct cross-fertilization between the complementary specifications of the open XML-based Rule Markup Language and of the Java runtime API for rule engines.

Slides With More Detail Follow this one

Flavors of Rules Commercially Most Important today in E-Business

- E.g., in OO app's, DB's, workflows.

- Relational databases, SQL: Views, queries, facts are all rules.
- Prolog. *"logic programs" as a full programming language.*
- Production rules (OPS5 heritage): e.g.,
  - Blaze, ILOG, Haley: rule-based Java/C++ objects.
- Event-Condition-Action rules (loose family), cf.:
  - business process automation / workflow tools.
  - active databases; publish-subscribe.
- *(Lesser: other knowledge-based systems.)*

Need to Go Beyond Classical/KIF

- Classical-logic/KIF has major limitations:
  - logically monotonic.
    - yet virtually all practical rule (and probability) systems are non-monotonic.
  - pure-belief, no procedural attachments.
    - yet most practical rule systems do invoke procedures external to the inference engine.
- Candidates to complement KIF exist:
  - logic programs, Bayes nets, ...

Example Domain: Rule-based Contracts for E-commerce

- Rules as way to specify (part of) business processes, policies, products: as (part of) contract terms.
- Complete or partial contract.
  - As default rules. Update, e.g., in negotiation.
- Rules provide high level of conceptual abstraction.
  - easier for non-programmers to understand, specify, dynamically modify & merge. E.g.,
  - by multiple authors, cross-enterprise, cross-application.
- Executable. Integrate with other rule-based business processes.

Criteria for Contract Rule Representation

1

2

3

- *High-level:* Agents reach common understanding; contract is easily modifiable, communicatable, executable.
- Inter-operate: heterogeneous commercially important rule systems.
- Expressive power, convenience, natural-ness.
- ... but: computational tractability.
- Modularity and locality in revision.
- Declarative semantics.
- Logical non-monotonicity: default rules, negation-as-failure.
  - essential feature in commercially important rule systems.
- Prioritized conflict handling.
- Ease of parsing.
- Integration into Web-world software engineering.
- Procedural attachments.

OLP

Courteous

XML

Situated

Ordinary Logic Programs as basic representation: Advantages

- Declarative: semantics is independent of inferencing procedure implementation, e.g., forward vs. backward chaining, sequencing of executing rules or conditions within rules.
- Expressive: relational expressions cf. SQL, large fragment of first-order logic, chaining, basic logical non-monotonicity (unlike first-order logic / ANSI-draft Knowledge Interchange Format).
- Efficient: computationally tractable given two reasonable restrictions:
  - 1. Datalog = no logical functions of non-zero arity.
  - 2. Bounded number v of logical variables per rule.
  - m = O(n^(v+1)), where n = ||LP||, m = ||ground-instantiated LP||.
  - Inferencing time is O(m) for broad case (stratified), O(m^2) generally (for well-founded semantics).
  - By contrast, first-order-logic inferencing is NP-hard.

Ordinary Logic Programs: Advantages (continued)

- Widely deployed and familiar:
  - relational DB's, SQL
  - Prolog
  - knowledge-based systems and intelligent agents
    - (e.g., IBM's Agent Building Environment)
- Common core shared semantically by many rule systems: e.g.,
  - relational DB's, SQL
  - Prolog
  - production rules (OPS5 heritage)
  - Event-Condition-Action rules
  - first-order-logic

Larger Vision: rules in e-business overall

- Rules as an important aspect of coming world of Internet e-business: rule-based business processes for both B2B and B2C.
  - represent seller's offerings of products & services, capabilities, bids; map offerings from multiple suppliers to common catalog.
  - represent buyer's requests, interests, bids; $\rightarrow$ matchmaking.
  - represent business processes, e.g., sales help, customer help, procurement, authorization/trust, brokering, workflow.
  - high level of conceptual abstraction; easier for non-programmers to understand, specify, dynamically modify & merge.
  - executable but can treat as data, separate from code
    - potentially ubiquitous; already wide: e.g., SQL views, queries.
- Rules in communicating applications, e.g., embedded intelligent agents.