Introduction to RuleML

Benjamin Grosof
MIT Sloan School of Management
Information Technologies group
http://www.mit.edu/~bgrosof

Harold Boley
National Research Council of Canada (NRC)
IIT Ebusiness (on leave from DFKI)
Harold.Boley@rc.ca http://www.dfki.de/~boley

Slides presented at 10/29/2002 Teleconference Meeting of
Joint US/EU ad hoc Markup Language Committee
http://www.daml.org/committee

With thanks to Steve Ross-Talbot, Bruce Spenser, Said Tabet, and Gerd Wagner

Outline of Talk

- Overview of RuleML Today
- Motivation, Background
  - – heterogeneous commercial rule systems/rep'ns
  - – evolutionary strategy for standards
- Fundamental Technical Issues and Approaches
  - – logic programs and extensions
  - – Webizing; syntax mechanics
  - – relationship to other Semantic Web standards
- Plans: Organizational, Technical

Overview of RuleML Today

- RuleML Initiative (2000--)
  - Dozens of institutions (~35), researchers; esp. in US, EU
  - Mission: Enable semantic exchange of rules/facts between most commercially important rule systems
  - Standards specification: 1st version 2001; basic now fairly stable
  - A number of tools (~12 engines, translators, editors), demo applications
  - Successful Workshop on Rules at ISWC was mostly about RuleML / LP
  - Can itself use a "home" institutionally.
    - Candidates: DAML, Joint Committee, W3C, Oasis
- Initial Core: Horn Logic Programs KR

…Webized (in markup)… and with expressive extensions

URI's, XML, RDF, …   *non-mon, actions, …*

Overview of RuleML Today, Continued

- Fully Declarative KR (not simply Prolog!)
  - Well-established logic with model theory
  - Available algorithms, implementations
  - Close connection to relational DB’s; core SQL is Horn LP
  - *See [Baral & Gelfond ’94] for good survey on declarative LP.*
- Abstract graph syntax
  - 1st encoded in XML…
  - … then RDF (draft), … then DAML+OIL (draft)
- Expressive Extensions incrementally, esp. already:
  - Non-monotonicity: Negation as failure; Courteous priorities
  - Procedural Attachments: Situated actions/effecting, tests/sensing
  - *In-progress:* Events cf. OPS5/Event-Condition-Action

Outline of Talk

- Overview of RuleML Today
- Motivation, Background
  - heterogeneous commercial rule systems/rep’ns
  - evolutionary strategy for standards
- Fundamental Technical Issues and Approaches
  - logic programs and extensions
  - Webizing; syntax mechanics
  - relationship to other Semantic Web standards
- Plans: Organizational, Technical

Flavors of Rules Commercially Most Important today in E-Business

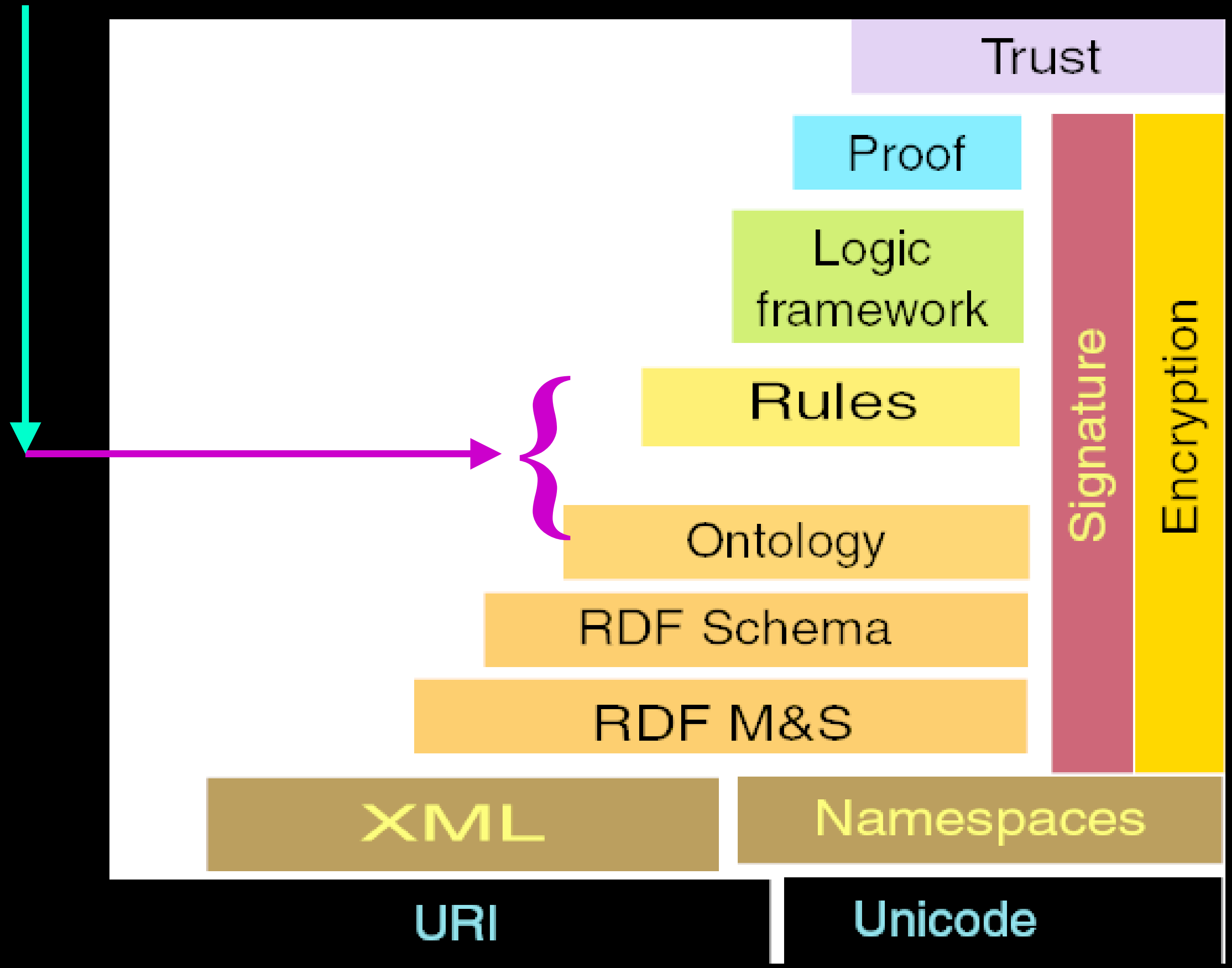
- E.g., in OO app’s, DB’s, workflows.

- Relational databases, SQL: Views, queries, facts are all rules.
  - SQL99 even has recursive rules.
- 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.
- Prolog. *“logic programs” as a full programming language.*
- *(Lesser: other knowledge-based systems.)*

Vision: Uses of Rules in E-Business

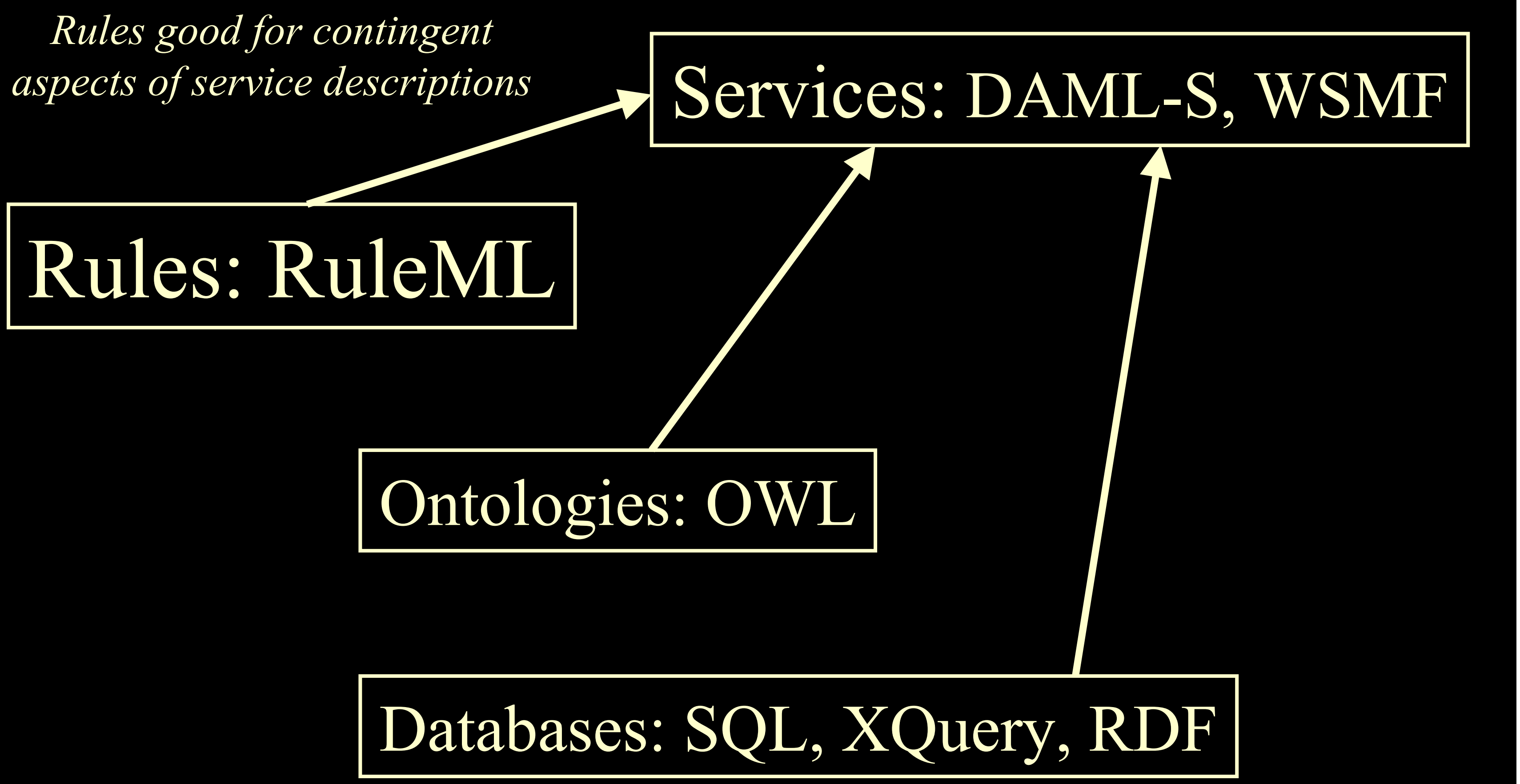
- Rules as an important aspect of coming world of Internet e-business: rule-based business policies & business processes, for B2B & B2C.
  - represent seller's offerings of products & services, capabilities, bids; map offerings from multiple suppliers to common catalog.
  - represent buyer's requests, interests, bids; → matchmaking.
  - represent 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.

Motivation from Semantic Web "Stack"



[Diagram http://www.w3.org/DesignIssues/diagrams/sw-stack-2002.png is courtesy Tim Berners-Lee]

Vision: Semantic Web and Web Services Use DB's, Ontologies, and Rule Systems



Rules good for contingent aspects of service descriptions

Services: DAML-S, WSMF

Rules: RuleML

Ontologies: OWL

Databases: SQL, XQuery, RDF

Why Standardize Rules Now?

- Rules as a form of KR (knowledge representation) are especially useful:
  - relatively mature from basic research viewpoint
  - good for prescriptive specifications (vs. descriptive)
    - a restricted programming mechanism
  - integrate well into commercially mainstream software engineering, e.g., OO and DB
    - easily embeddable; familiar
    - vendors interested already: Webizing, app. dev. tools
- *⇒⇒ Identified as part of mission of the W3C Semantic Web Activity*

Outline of Talk

- Overview of RuleML Today
- Motivation, Background
  - heterogeneous commercial rule systems/rep'ns
  - evolutionary strategy for standards
- Fundamental Technical Issues and Approaches
  - logic programs and extensions
  - Webizing; syntax mechanics
  - relationship to other Semantic Web standards
- Plans: Organizational, Technical

Technical Approach of RuleML: I

- 1. Expressively: Start with: Datalog Logic Programs *as kernel*
  - Rule := H ← B1 /\ … /\ Bk ; k ≥ 0, H and Bi's are atoms.

    head if body ;
- Declarative LP with model-theoretic semantics
  - forward ("derivation"/ "transformation") and backward ("query") inferencing
- Rationale: captures well a simple shared core among CCI rule sys.
  - Tractable! (if bounded # of logical variables per rule)
- Horn LP -- differences from Horn FOL:
  - Conclusions are a set of ground atoms.
  - Consider Herbrand models only, *in typical usage.*
    - Can extend to permit equalities in rules/conclusions.
  - Rule has non-empty head, *in typical usage.*

Technical Approach of RuleML: II

- 2. Syntax: Permit rules to be labeled -- need names on the Web!
- 3. Syntax: Permit URI’s as predicates, functions, etc. (names)
  - namespaces too
- 4. Expressively: Add: extensions cf. established research
  - negation-as-failure (well-founded semantics) -- in body *(stays tractable!)*
    - “Ordinary” LP (cf. declarative pure Prolog)
  - classical negation: limited to head or body atom – syntactic sugar
  - prioritized conflict handling cf. Courteous LP *(stays tractable!)*
    - modular rulesets; modular compiler to Ordinary LP
  - procedural attachments: actions, queries ; cf. Situated LP
  - 1st-order logic type expressiveness cf. Lloyd LP’s – syntactic sugar
    - ∨,∀,∃ in body; ∧,∀ in head *(stays tractable!)*
  - logical functions (arity > 0)

Technical Approach of RuleML: III

- 5. Expressively: Add: restrictions cf. established R&D
  - E.g., for particular rule systems, e.g., Prolog, Jess, …
    - Also “pass-thru” some info without declarative semantics (pragmatic meta-data)
- 6. Syntax for XML:
  - Family of DTD’s/Schemas:
    - a generalization-specialization hierarchy (lattice)
    - define DTD’s modularly, using XML entities (~macros)
    - optional header to describe expressive-class using “meta-”ontology
- 7. Syntax: abstract unordered graph syntax (data model)
  - Support RDF as well as XML (avoid reliance on sequence in XML)
  - “Roles” name each child, e.g., in collection of arguments of an atom
  - Orderedness as optional special case, e.g., for tuple of arguments of an atom
- 8. Syntax: module inclusion: merge rulesets ; import/export
  - URI’s name/label knowledge subsets

Outline of Talk

- Overview of RuleML Today
- Motivation, Background
  - heterogeneous commercial rule systems/rep’ns
  - evolutionary strategy for standards
- Fundamental Technical Issues and Approaches
  - logic programs and extensions
  - Webizing; syntax mechanics
  - relationship to other Semantic Web standards
- Plans: Organizational, Technical

Technical Plans: I

- *Refine existing* expressiveness
  - Procedural attachments, built-in functions, and events are foci now
- *Refine existing* syntax
  - RDF and modules are foci now
- *More expressive extensions:*
  - Ontologies imported from OWL; exporting to OWL too
    - E.g., cf. Description Logic Programs [Grosof & Horrocks ’02]
  - Typing
  - Events, e.g., cf. Event-Condition-Action / production rules; “reactive”
  - Equalities in heads/conclusions
  - Equivalence/rewriting/transformation rules
  - Integrity checking (still regarded as special queries)
  - *Later:* temporal, …

Technical Plans: II

- “Header” meta-data
  - specify KR incl. expressive/syntactic restrictions
  - Seems good opportunity to use OWL

- More tools: translators, editors, inference engines
  - IBM has announced it will support in CommonRules V3

- More application scenarios
  - Services, e-contracting, financial info integration, …
    - E.g., SweetDeal [Grosof & Poon ’02]

Relationships to other Semantic Web areas

- Services
  - *See talks on Services and Rules from 10/02 DAML PI Mtg*
    - *http://ebusiness.mit.edu/bgrosof/#DAMLRulesInvitedTalkOct02*
- OWL/Ontologies
  - Overlap: Description Logic Programs (DL)
    - *See paper and talk at http://ebusiness.mit.edu/bgrosof/#DLP*
  - Interesting to compare expressive power, usage scenarios
    - E.g., can DL represent discount pricing policies?
- RDF -- current: 1. exploit new collection classes ; 2. RDF Query
- DQL, Explanations ; Xquery and RDF Query too!
- CommonLogic
  - Aim: maximize congruity on overlap – e.g., Horn case
  - Pat Hayes liaison from DAML / Joint Committee

Organizational Plans/Efforts: I

- Establish closer relationship with DAML
- Establish closer relationship with Joint Committee
- W3C Working Note (in progress)
- Aim: W3C Working Group - 2003?
  - Relationship to RDF Query area
- Use www-rdf-rules list (already blessed by W3C)

Organizational Plans/Efforts: II

- Possible OASIS Technical Committee on Policy RuleML – interchange policies via RuleML
- Industry outreach: developers, executives
- Events (being planned) for 2003:
  - WWW Conf., ISWC, W3C Plenary (Boston 3/03), …
- Website:
  - editing/revamping
  - setting up ruleml.org site

RuleML Subgroups -- started recently

- Reaction Rules
  - Leads: Gerd Wagner & Steve Ross-Talbot
- Ontology Combo
  - Leads: Benjamin Grosof & Andreas Eberhart
- Defeasible Rules
  - Grigoris Antoniou & Michael Schroeder

- Thanks!

- Questions?

- For More Info:
  - http://ebusiness.mit.edu/bgrosof/#RuleML
  - http://www.dfki.de/ruleml

Bibliography

- Baral, Chitta and Gelfond, Michael. "Logic Programming and Knowledge Representation". J. Logic Programming 19,20:73-148, 1994. *Good Review of LP.*

  http://ebusiness.mit.edu/bgrosof/paps/lp+kr-baral+gelfond.pdf
- Grosof, Benjamin. "Representing E-Business Rules for the Semantic Web: Situated Courteous Logic Programs in RuleML". Proc. Wksh on Information Technologies and Systems (WITS-01), held 2001 at the Intl. Conf. on Information Systems (ICIS). *Describes SweetRules tool as well as RuleML.*
- Grosof, Benjamin and Horrocks, Ian. "Description Logic Programs: Combining Logic Programs with Description Logic". Working Paper. Draft of 10/17/2002 is at: http://ebusiness.mit.edu/bgrosof/#DLP along with complementary talk slides.
- Grosof, Benjamin and Poon, Terrence. "Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions". Proc. Intl. Wksh. on Rule Markup Languages for Business Rules on the Semantic Web, held 6/02 at the 1st Intl. Semantic Web Conf. (ISWC-2002). *Describes SweetDeal.*
- Grosof, Benjamin and Gandhe, Mahesh and Finin, Tim. "SweetJess: Translating DamlRuleML to Jess". Proc. Intl. Wksh. on Rule Markup Languages for Business Rules on the Semantic Web, held 6/02 at the 1st Intl. Semantic Web Conf. (ISWC-2002).