Business Rules for E-Commerce:

Interoperability and Conflict Handling

Dr. Benjamin Grosof

IBM T.J. Watson Research Center

Hawthorne, NY, USA

grosof@us.ibm.com http://www.research.ibm.com/people/g/grosof

Outline

- Business Rules: overview of our technical focus and approach.
- EECOMS consortium for advanced supply chain management in manufacturing.
- More on technical approach.
- Exploratory areas.

Rules: Fundamental Technical Approach

- Aim to enable: **Exchange business rules**, dynamically.
- Context: key application logic is represented via rules, in many systems.
- E.g., exchange between supply chain app's for manufacturing industry.
- E.g., rules about when and how to order or return items, that impact planning.
- E.g., rules about terms & conditions associated with purchasable item.
- Declarative approach: provide **semantics** that is clean and deep.
- **Facilitate specification** of a given rule set by multiple authors, non-technical authors, authors in different enterprises or writing different applications, dynamically. Abstraction level more easily human-understandable.
- Enable **conflict handling** in multiple rule systems.
- **Interoperate** between multiple rule systems via common-core **interlingua**.

Courteous and Situated Logic Programs: Overview

- Point of departure: Logic programs as pure knowledge representation.
- Courteous logic programs **handle conflict**.
- Merging, updating of rule sets: is crucial, often generates conflict.
- Priorities between rules: partially-ordered.
- Guarantee consistency.
- Fast: low overhead computational cost for the conflict handling.
- Situated logic programs hook beliefs to **drive procedural API's**.
- Procedural attachments for condition-testing, action-performing. Specified as part of the knowledge representation: sensor, effector link statements.

EECOMS Supply Chain Project: Overview

- EECOMS = Extended Enterprise Consortium for Integrated Collaborative Manufacturing Systems.
- IBM-led consortium, matching-funded by US government's NIST Advanced Technology Program. $29M over 3 years ('98-'01). (http://www.atp.nist.gov)
- Advanced supply chain management. Consortium includes manufacturing software vendors, rules and tool vendors, manufacturer customers.
- Business Focus: improve **"agility"** of manufacturing. Respond to common but unpredictable events such as late delivery, plant line breakdown, larger than expected order. React quickly, including modify plans, schedules. Integrate: typically multi-application, very often multi-enterprise.
- Technical Focus: **rules and conflict handling**; virtual situation room for human collaborative workflow; attendant tools, agents, and security issues.
- Is follow-on to CIIMPLEX (IBM-led NIST ATP $22M) & challenges it identified. Shares: consortium, scenarios, agent-based approach.

EECOMS: Participants

- IBM is leader of consortium, via its Manufacturing ISU. Runs EECOMS Project Office, EECOMS Integration Center and Tools Dev..
- Ingersoll Rand (several sites). A major manufacturer. Acts as customer.
- Berclain, QAD: manufacturing/supply-chain software vendors.
- The Haley Enterprise: (small) leading rules vendor.
- Vitria, EnvisionIt, Scandura, IndX: (small) tools/specification vendors.

- Research organizations:
  - IBM T.J. Watson
  - Universities: U. NC Charlotte, U. MD Baltimore, U. Florida Gainesville. (Formally, are sub-contractors to IBM.)

Supply Chain: Example Rules

- Vendor's rules that prescribe how buyer must place or modify an order:
- A) 14 days ahead if the buyer is a qualified customer.
- B) 30 days ahead if the ordered item is a minor part.
- C) 2 days ahead if the ordered item's item-type is backlogged at the vendor, the order is a modification to reduce the quantity of the item, and the buyer is a qualified customer.

- Suppose more than one of the above applies to the current order? **Conflict!**

- Helpful Approach: **precedence** between the rules. Often only *partial* order of precedence is justified. E.g., C $\succ$ A.

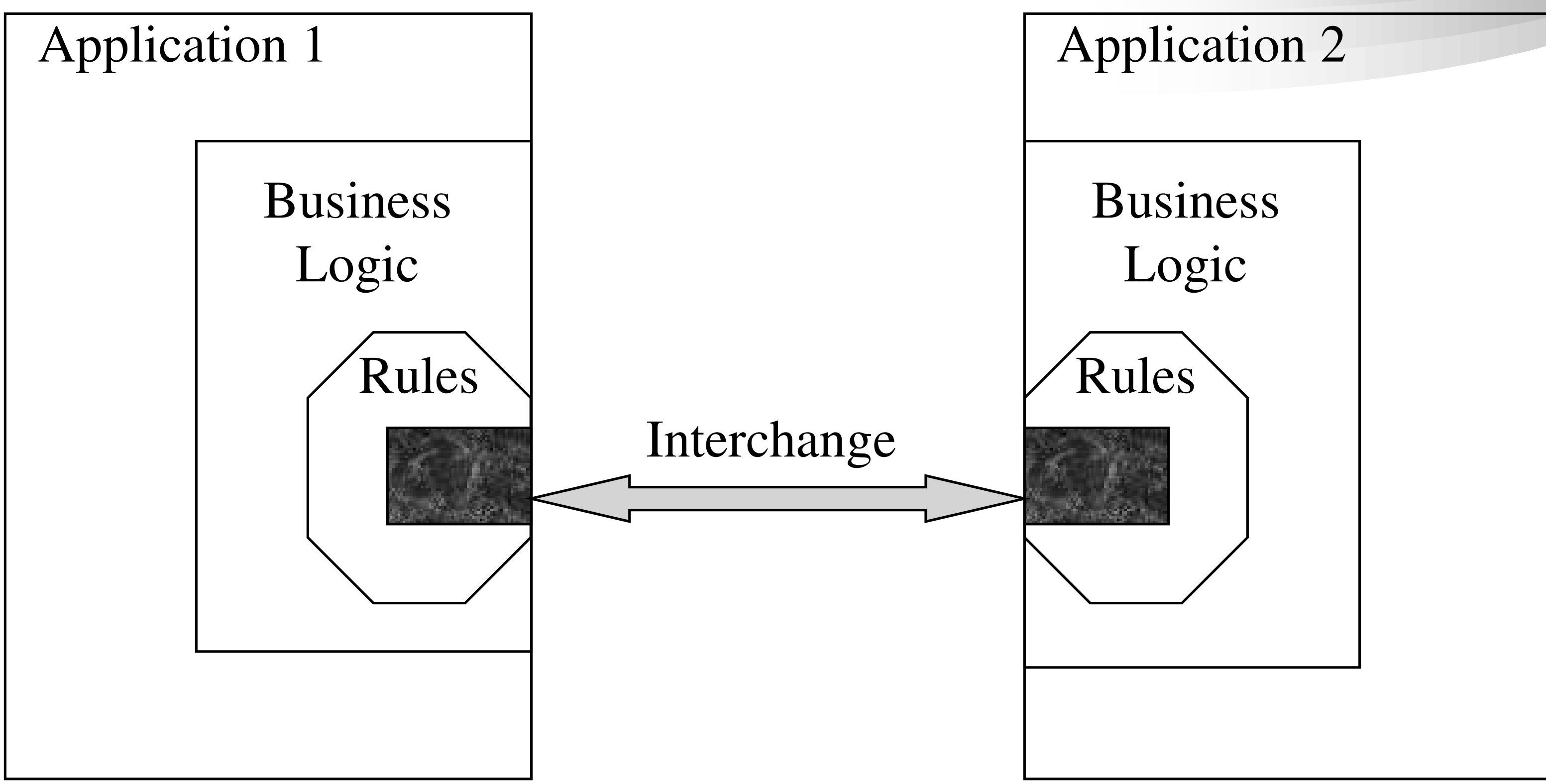
Roles for Rules

- 1st step: Rules as rigorous specification without execution.

- 2nd step: Rules as executable specification.
  - Coarse-grain integration with rest of code. E.g.:
    - rules engine in business object, application object, agent, ….

- 3rd step: Fine-grain integration with rest of code. E.g.:
  - OO App. Dev. tool’s rules feature.

Rules across Applications



Exploratory: Software Areas

- Relation to DB, OO App. Dev., Workflow, Enterprise Integration.
- Import/Export relationship to current and emerging standards for business rules: OMG, SQL, XML/RDF, ANSI KIF, ECA, FIPA, vendors.

Exploratory: Technical Areas

- OO context for rules execution, specification, procedural attachments: reformulating implicit context $\leftrightarrow$ explicit context conditions for applicability of rules.
- Events vs. conditions, in rule antecedents.
- Authoring of rules via graphical templates and limited natural-language.

Optional Slides follow this one

Courteous *Logic Programs: Overview*

- Rules permit optional labels $\langle ... \rangle$, classical **negation** ¬, negation-as-failure **~ :**
- $\langle$rule1$\rangle$ P(x,y,w) ∧¬Q(y,z) ∧ ~R(x,y) → S(x,z,w).
- $\langle$rule2$\rangle$ A(x,Fred) ∧ ~¬B(Fred,Jane) → ¬S(x,Fred,Jane).
- **Priorities** between rules via fact comparing their labels :
- Overrides(rule1,rule2). Means rule1 higher-pri than rule2.
- Priorities can represent updating, merging, closed-world, exceptional cases, greater authority or reliability.
- Guaranteed **consistent**, unique set of conclusions.
  - Never believe both p & ¬p.
- **Fast**: low computational overhead for the conflict handling.

Situated Courteous LP's: *Overview*

- Above is pure-belief courteous LP. Situated LP: further permits a clean kind of **procedural attachments**.
- Procedural attachments for **sensing** (queries) when testing an antecedent condition or for **effecting** (actions) upon concluding a consequent condition. Attached procedure is invoked when testing or concluding in inferencing.
- Statements for sensor link or effector **link** specify an association from a relation to a procedural call pattern:
- S ➥ myObject::myMethod . *example of a sensor link statement*
- P(u,v) ➥ yourAPI::yourMethod(yourType1 u , yourClass2 v). *effector ex.*
- Overall: cleanly separate out the procedural semantics as an extension of the pure-belief declarative semantics. Easily separate chaining from action.
- Recap: usual rules, with negation, plus prioritized conflict handling, plus procedural attachments for sensing or effecting.

Rules Authoring Approaches

- Graphical.
- Natural Language (limited).
- Pre-defined templates and vocabulary.
  - domain/industry specific.
  - application/suite specific.

EECOMS: Schedule

- May 1997: Proposal.
- Oct. 1997: Award by NIST.
- 1998: Real work begins.
- 2001: Wind-up. **Commercialization** (required by NIST).