

Rule-based Technology for Automating Contracting by Agents

*Invited Presentation for
American Bar Association (ABA) Spring Meeting 3/22-25/01:
Cyberlaw Committee, Internet Law Subcommittee & Electronic Agents Task Force;
March 24, 2001 at Philadelphia Convention Center, Philadelphia, PA, USA*

Prof. Benjamin Grosf

MIT Sloan School of Management
bgrosf@mit.edu <http://www.mit.edu/~bgrosf/>

3/26/2001

by Benjamin Grosf copyrights reserved

Outline

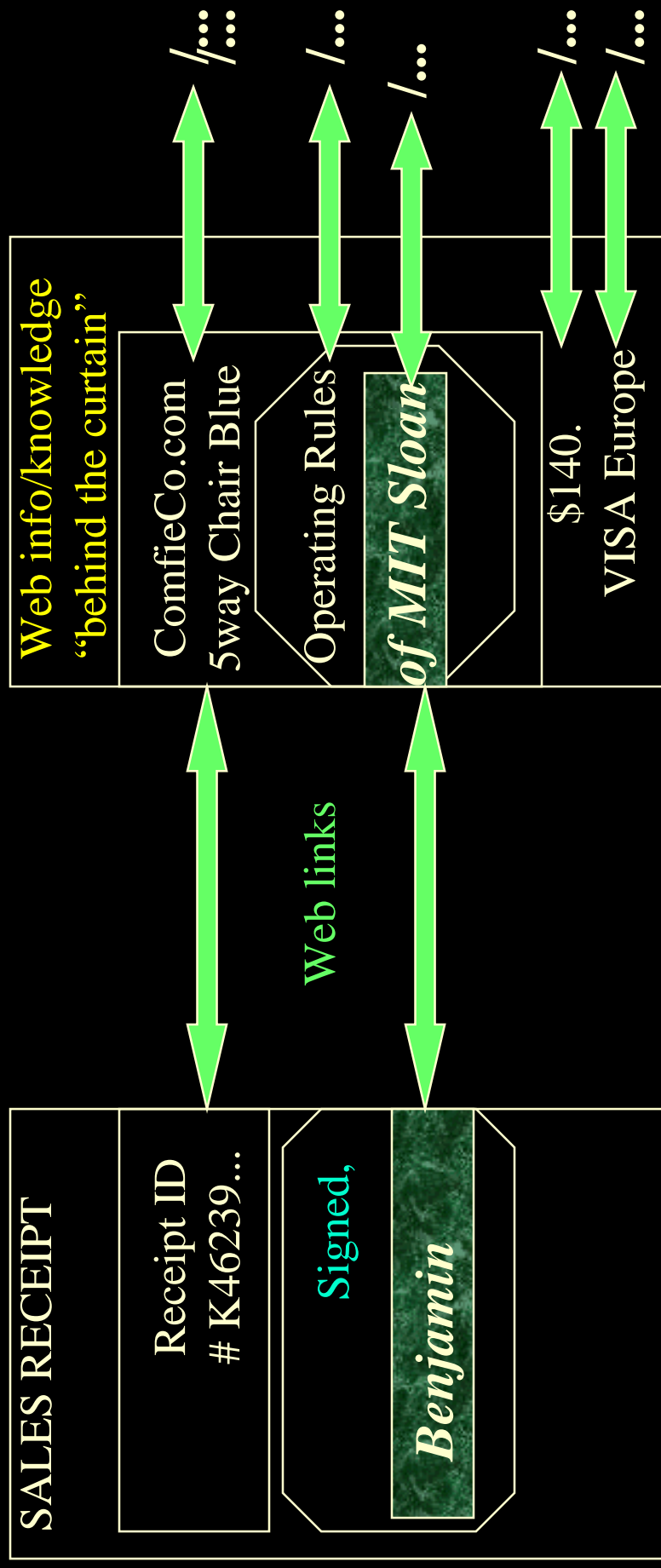
- 1. Intro: E-Signatures → deeper issues
- 2. What's Doable Now in rule-based agent contracting
 - functionality: communicate, execute, modify
 - what kind of stuff represented by rules
- 3. Example of Agent Contract Communication:
 - Approach: Inter-operable, modular XML Rules represent parts of Contract Content
- 4. Applications:
 - Current
 - Vision
- 5. *Discussion*
- + *Optional Slides: my background; agent delegations*

Deeper Issues of E-Signatures

- WHAT'S THE DEAL ? ... !!
- SIGN AS WHAT ?? ... !!
- *Vision/Approach:* A net of documents combined by links, on the Web

Looks Simple To Start... then Gets Interestingly Precise

A Vision/Approach of what Web & Agents enable



What's Doable Today in rule-based agent contracting,

based on latest technological & conceptual progress

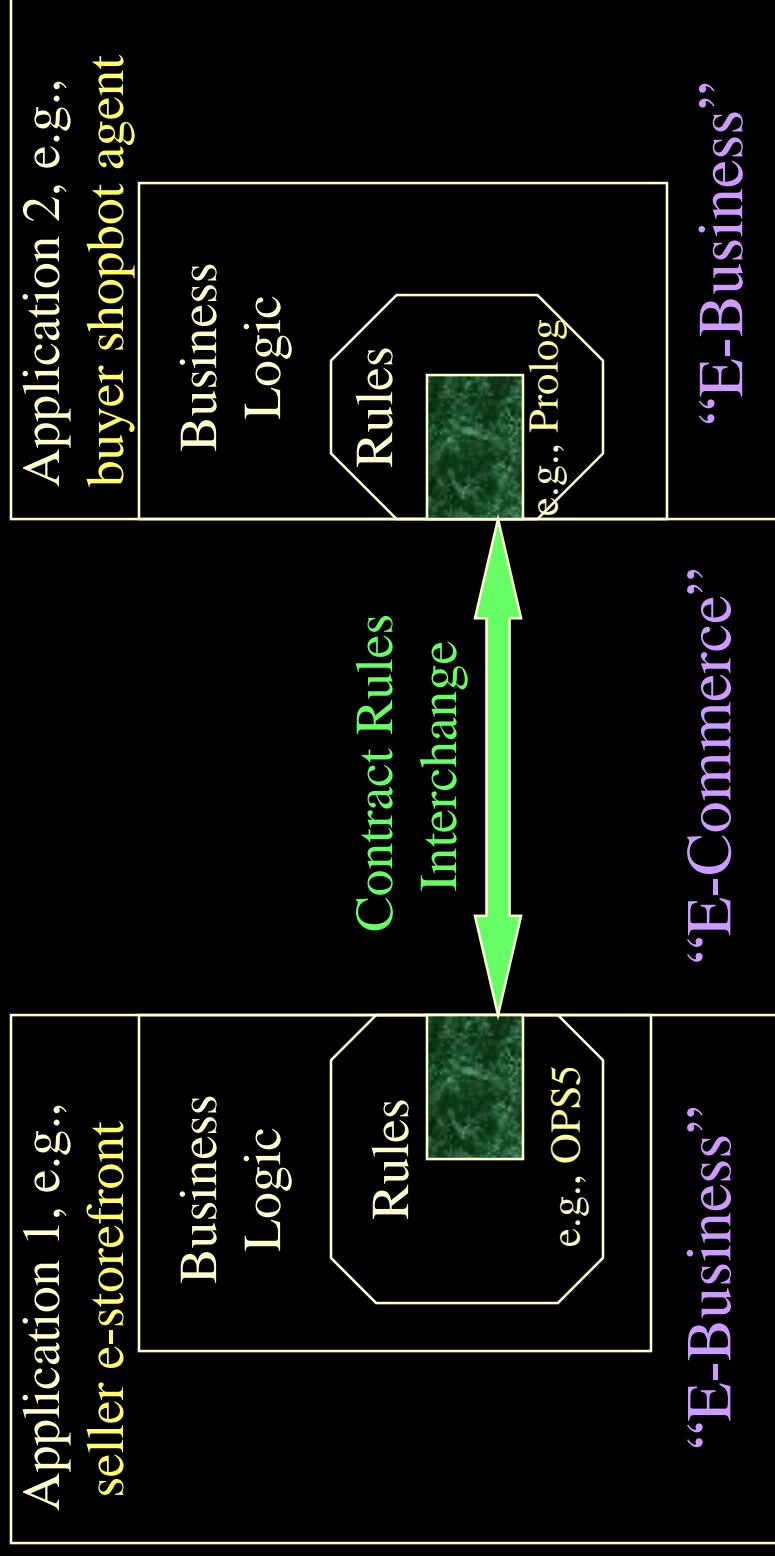
- Communicate:
 - XML, interoperable
 - \Leftrightarrow heterogeneous rule systems / rule-based agents
- Execute contract provisions:
 - infer; ebiz actions; authorize; ...
- Modify easily: contingent provisions
 - default rules; modularity;
- Reason about the contract/proposal
 - hypotheticals, test, evaluate

Approach:

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.

Contract Rules across Applications / Enterprises

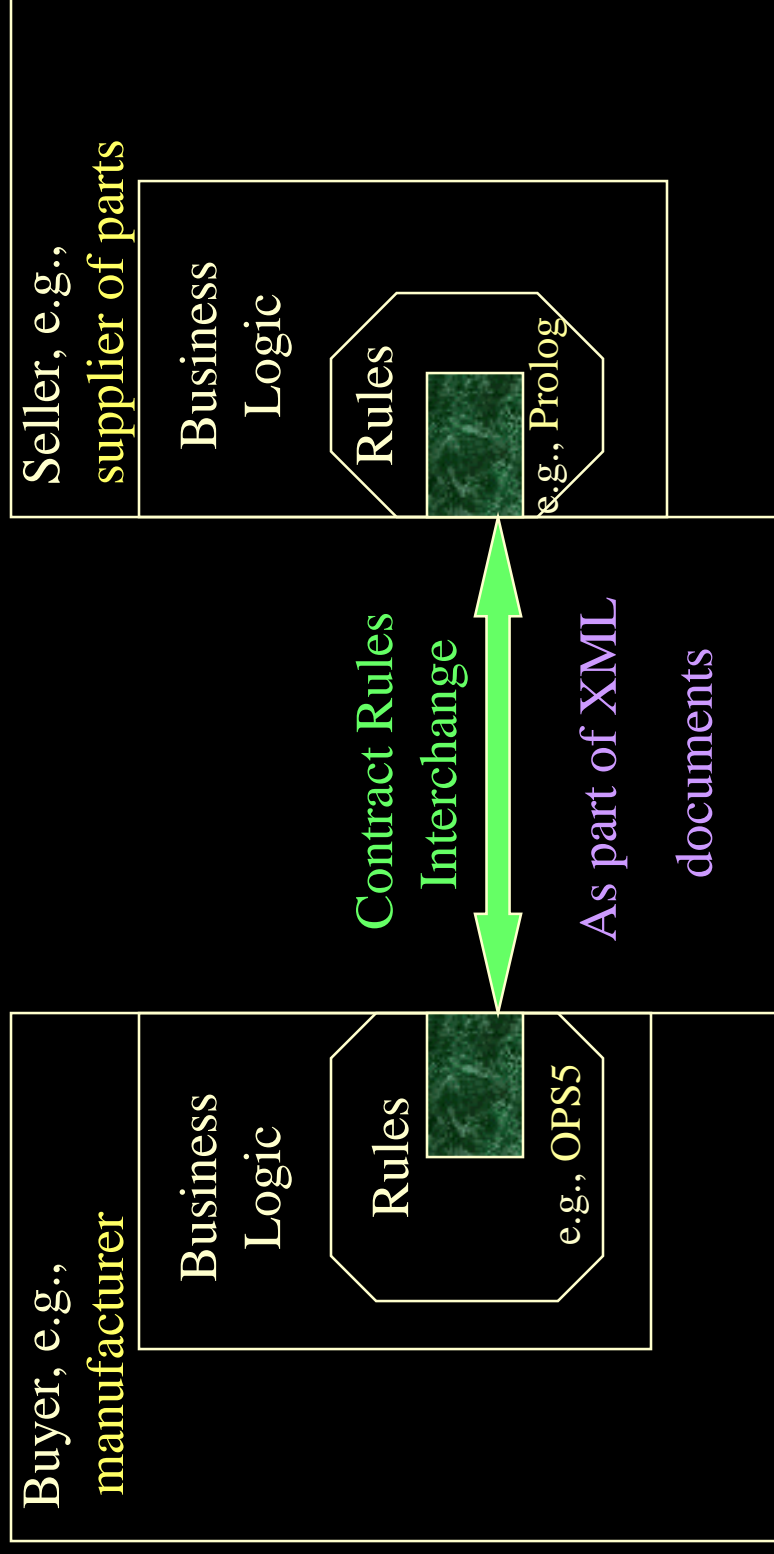


Contracting parties integrate e-businesses via shared rules.

Examples of Rules in Agent Contracts & Deal Making

- Product descriptions
 - Product catalogs: properties, conditional on other properties.
- **Price vs. quantity vs. delivery date.**
 - Discounting, incl. for groups.
- Terms & conditions
 - Service provisions
 - Refunds, cancellations.
 - Surrounding business processes, e.g., **lead time** to order.
- Trust
 - Creditworthiness, authorization, required signatures
- *Buyer Requirements (RFQ, RFP) wrt the above*
- *Seller Capabilities (Sourcing, Qualification) wrt the above*

Contract Rules during Negotiation

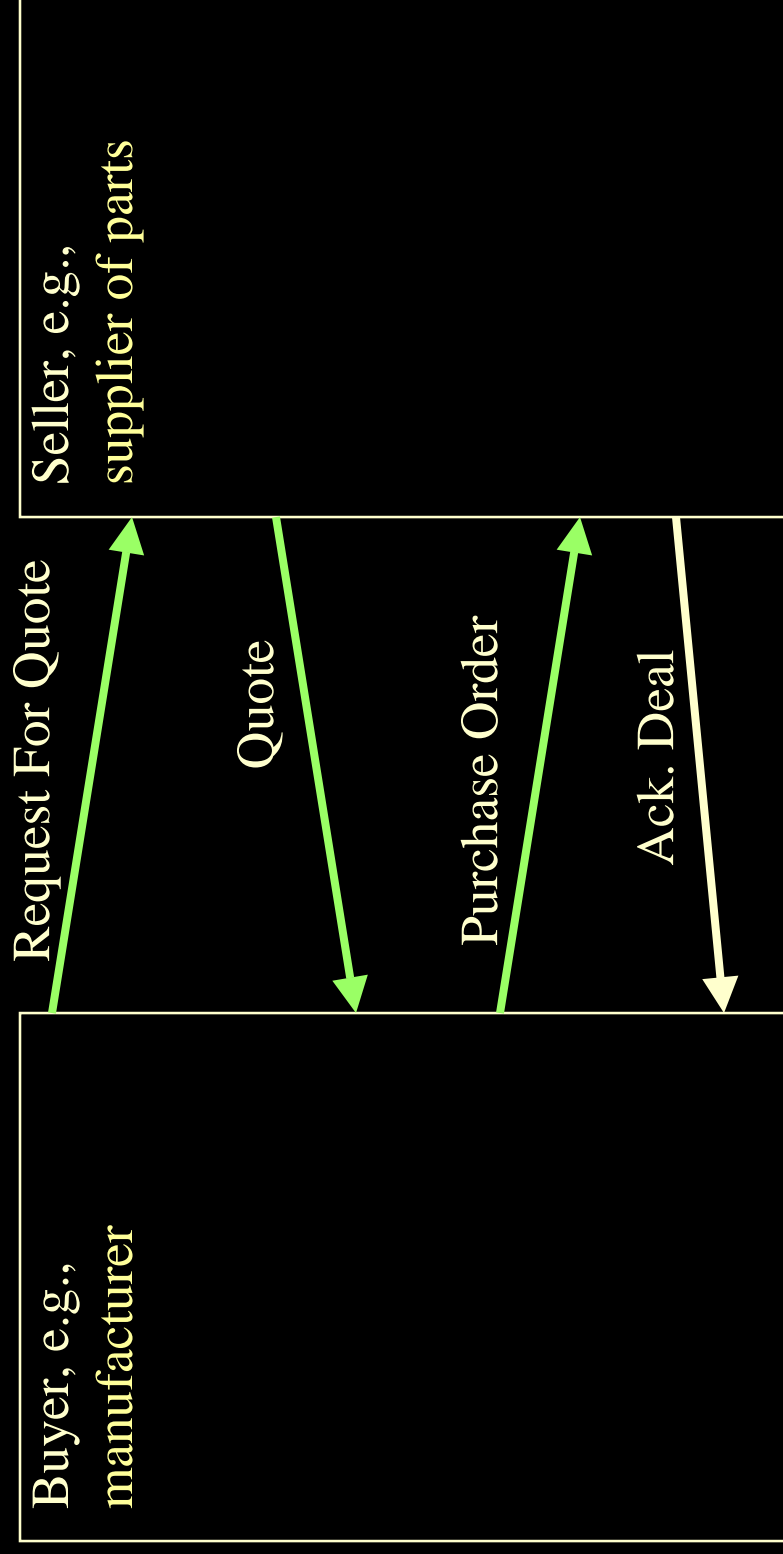


Contracting parties NEGOTIATE via shared rules.

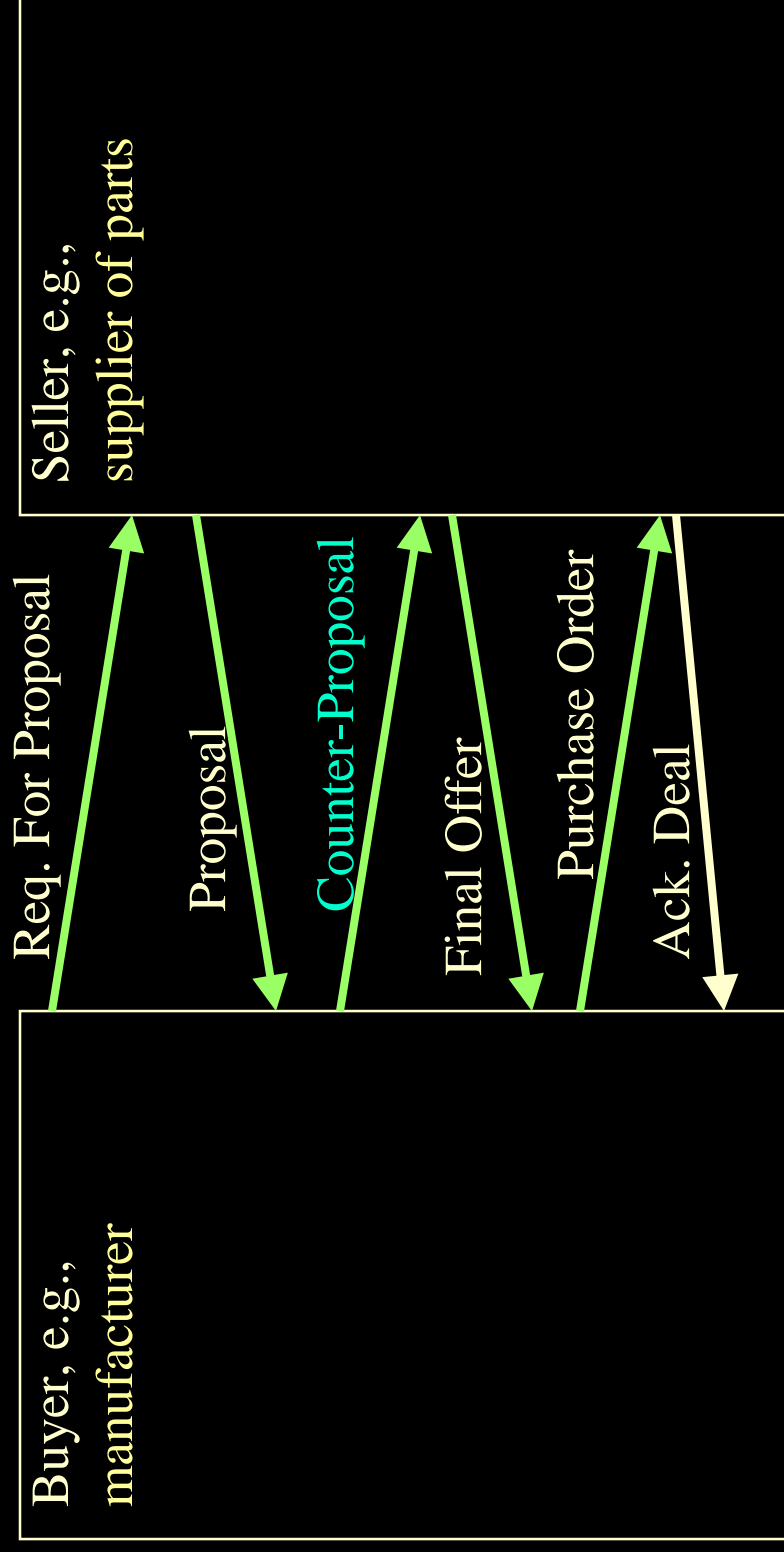
3/26/2001

by Benjamin Groszof copyrights reserved

Exchange of Rules Content during Negotiation: example



Exchange of Rules Content during Negotiation: example



Negotiation Example XML Document: Proposal from supplierCo to manufCo

```
<negotiation_message>  
<message_header>  
  <proposal/>  
  <from> supplierCo </from>  
  <to> ManufCo </to>  
</message_header>  
<rules_content>  
  ...[see next slide]  
</rules_content>  
  ...  
</negotiation_message>
```

Example of similar message document format:

FIPA Agent Communication Markup Language (draft industry standard).

Negotiation Ex. Doc. Rules:

Proposal from supplierCo to manufCo

- ...
<usualPrice> price(per_unit, ?PO, \$60) ←
- purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧
- quantity_ordered(?PO, ?Q) ∧ (?Q ≥ 5) ∧ (?Q ≤ 1000) ∧
- shipping_date(?PO, ?D) ∧ (?D ≥ 24Apr00) ∧ (?D ≤ 12May00).
- <volumeDiscount> price(per_unit, ?PO, \$51) ←
- purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧
- quantity_ordered(?PO, ?Q) ∧ (?Q ≥ 100) ∧ (?Q ≤ 1000) ∧
- shipping_date(?PO, ?D) ∧ (?D ≥ 28Apr00) ∧ (?D ≤ 12May00) .
- overrides(volumeDiscount, usualPrice) .
- ⊥ ← price(per_unit, ?PO, ?X) ∧ price(per_unit, ?PO, ?Y) GIVEN (?X ≠ ?Y).
- ...

Negotiation Ex. Doc. Rules:

Counter-Proposal from *manufCo* to *supplierCo*

- ...
- `<usualPrice> price(per_unit, ?PO, $60) ← ...`
- `<volumeDiscount> price(per_unit, ?PO, $51) ←`
 - `purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧`
 - `quantity_ordered(?PO, ?Q) ∧ (?Q ≥ 5) ∧ (?Q ≤ 1000) ∧`
 - `shipping_date(?PO, ?D) ∧ (?D ≥ 28Apr00) ∧ (?D ≤ 12May00) .`
- `overrides(volumeDiscount, usualPrice) .`
- `⊥ ← price(per_unit, ?PO, ?X) ∧ price(per_unit, ?PO, ?Y) GIVEN (?X ≠ ?Y) .`
- `<aSpecialDeal> price(per_unit, ?PO, $48) ←`
 - `purchaseOrder(?PO, supplierCo, manufCo) ∧`
 - `quantity_ordered(?PO, ?Q) ∧ (?Q ≥ 400) ∧ (?Q ≤ 1000) ∧`
 - `shipping_date(?PO, ?D) ∧ (?D ≥ 02May00) ∧ (?D ≤ 12May00) .`
- `overrides(aSpecialDeal, volumeDiscount) .`
- `overrides(aSpecialDeal, usualPrice) .`
- ...

Simply

added

rules!

In XML: Business Rules Markup Language

- `<clp>`
- `<erule rulelabel="usualPrice">`
- `<head>`
- `<cliteral>`
- `<predicate name="price" arity="3" />`
- `<larglist>`
- `<ifunction name="per_unit" />`
- `<variable name="PO" />`
- `<function name="$60" />`
- `</larglist>`
- `</cliteral>`
- `</head>`
- `<body> ... (see next page) </body>`
- `</erule>`
- ...
- `</clp>`

Business Rules Markup Language for Negotiation Example (continued)

- `<body>`
- `<andb>`
- `<fcliteral>`
- `<predicate name="purchaseOrder" arity="3"/>`
- `<larglist>`
- `<variable name="PO"/>`
- `<lfunction name="supplierCo"/>`
- `<variable name="AnyBuyer"/>`
- `</larglist>`
- `</fcliteral>`
- `<fcliteral>`
- `...`
- `</fcliteral>`
- `...`
- `</andb>`
- `</body>`

3/26/2001

by Benjamin Grosfod copyrights reserved

Commercial Implementation & Piloting

- **IBM CommonRules:** AlphaWorks Java library
 - implements rule-based capabilities:
 - XML inter-operability; prioritized conflict handling
- **Rule Markup Language:** nascent industry standards effort
 - XML Knowledge Representation (KR) → make the Web be “Semantic”
 - KR: **Situated Courteous Logic Programs in XML**
- EECOMS industry consortium including Boeing, Baan, TRW, Vitria, IBM, universities, small companies
 - \$29Million 1998-2000; 50% funded by NIST ATP
 - application piloted
 - contracting & negotiation; authorization & trust

3/26/2001

by Benjamin Grosfod copyrights reserved

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.
- 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.*)

Criteria for Contract Rule Representation

- 1 • *High-level*: Agents reach common understanding; contract is easily modifiable, communicatable, executable.
- 2 • Inter-operate: heterogeneous commercially important rule systems.
• Expressive power, convenience, natural-ness.
• ... but: computational tractability.
- 3 • 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

Also Currently Being Developed in the world today

- Delegations between agents
- XML Ontologies (Vocabularies)
 - knowledge representation: infer with definitional knowledge
 - specific domain/industry vocabularies
- DARPA Agent Markup Language: ontologies, rules
- Industry Standards:
 - Web
 - Agents, Business Processes, Workflow
 - E-Commerce
 - Industry-Specific
 - *Legal XML*
- *Law: Electronic Signatures, ...*

Applications: Current and Visions

- product and contract/deal descriptions
- negotiation
- authorization
- automating legal reasoning and processes
- evidence
- regulations
- **Alternative Dispute Resolution**
- **adjudication, legal decision-making**
- ***?pointers?***

- Thanks!
- Questions?
- Comments? Pointers?
- For More Info:
 - <http://www.mit.edu/~bgrosof/>
 - links to <http://www.research.ibm.com/rules/>

OPTIONAL SLIDES FOLLOW

3/26/2001

by Benjamin Grosof copyrights reserved

Delegation Logic: Goal and Basic Approach

- Our goal: Develop a language that
 - can represent, with significant expressive power, policies and credentials for authorization in Internet scenarios
 - can provide mechanisms for delegation
 - has a clear declarative semantics
- Our approach: Delegation Logic (DL): multi-agent logic programs with *delegation to complex delegates*
 - D1LP: extends negation-free OLP \Rightarrow with delegation
 - D2LP: extends Courteous LP \Rightarrow with delegation
 - Tractable “Delegation compiler” similar to courteous compiler.
- Collaborators: Ninghui Li (NYU \rightarrow Stanford), Joan Feigenbaum (ATT \rightarrow Yale)

Delegation Logic (DILLP) Example: accessing medical records

- **Problem:** Hospital HM to decide: requester Alice authorized for patient Peter?
- **Policies:** HM will authorize only the patient's physician. HM trusts any hospital it knows to certify the physician relationship. Two hospitals together can vouch for a 3rd hospital.
 - HM says **authorized**(?X, read(medRec(?Y))) if HM says inRole(?X, **physic**(?Y)).
 - HM **delegates inRole**(?X, **physic**(?Y))^1 to threshold(1,?Z, HM says inRole(?Z,hosp)).
 - HM **delegates inRole**(?H,hosp)^1 to threshold(2 , ?Z, HM says inRole(?Z,hosp)).
- **Facts:** HC certifies Alice is Peter's physician. HM knows two hospitals HA and HB. HA and HB each certify HC as a hospital.
 - HC says inRole(Alice, **physic**(Peter)). HA says inRole(Joe, **physic**(Sue)).
 - HM says inRole(HA,hosp). HM says inRole(HB, hosp).
 - HA says inRole(HC,hosp). HB says inRole(HC, hosp).
- **Conclusion:** HM says **authorized**(Alice, read(medRec(Peter))). *Joe NOT authorized.*

Launch Vector: My Background

E-Commerce Agents, Rules: Techno + Biz

- Harvard BA math econ & mgm sci
- startups
- Stanford CS (Computer Science) PhD in AI
- IBM Watson Research: IA for EC
 - Led Intelligent Agents, Business Rules for E-Commerce
- MIT Sloan: Information Technology group
- Technology end of B-school IT world
- CS + Business Perspective (cf. Industry, cf. B-school):

theory theory



*practical
theory
+ pilot app's*

- how/where the technology is useful, important
- business value; implications for processes & strategies
- market evolutions; innovation paths; organizational changes

Background in Law-related Research

- Overall: formally represent policies and info as rules
- Evidential Reasoning: probabilistic, fuzzy, ...
- Bureaucratic Processes as domain
 - pioneer within AI knowledge representation community
- Argumentation with rule-based beliefs:
 - **efficient** algorithms
 - theory
 - bridge to commercially practical rule-based/database systems
- Contracting & Negotiation, Authorization & Trust
- *Invited Speaker at 2001 International Conference on AI & Law*:
 - “Automating Law in the Small: Contracts, Regulations, and Prioritized Argumentation”