

What's My Deal? Contract Communications in XML Agent Marketplaces

*Invited Presentation for
Center for eBusiness @ MIT Annual Conference
April 19, 2002, Cambridge, MA USA*

Prof. Benjamin Grosf

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

Outline

- *Aiming to be provocative*
- 1. Intro: XML, Semantic Web
 - example: e-signatures: deeper issues & opportunities
- 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

SweetDeal system

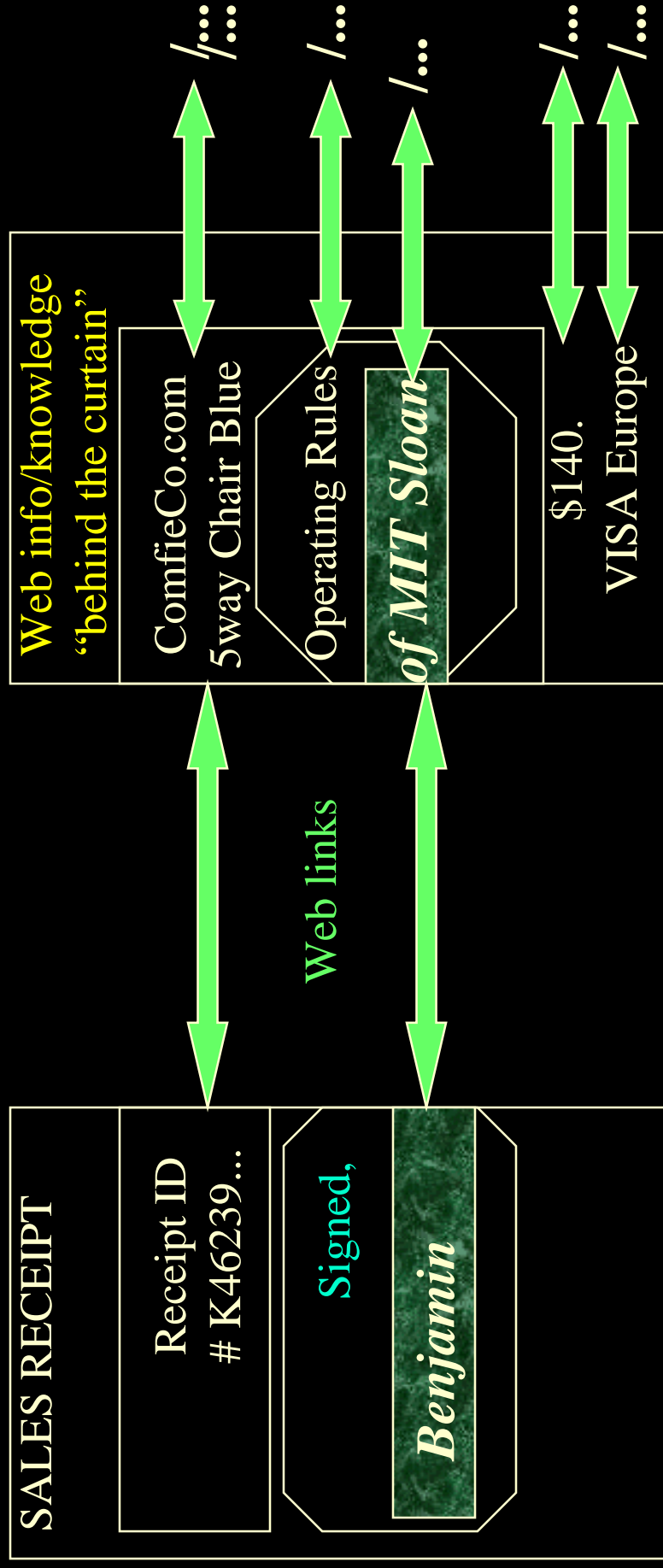
- 5. Discussion: *Directions for the Glorious Future*

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



The Web is becoming XML → the Semantic Web

- XML (vs. HTML) offers much greater capabilities for structured detailed descriptions that can be processed automatically.
 - Eases application development effort for assimilation of data in inter-enterprise interchange
 - A suite of open standards both current and emerging
 - ... including for knowledge-level SEMANTICS
- *Soon, Agents will Talk according to these standards...*
 - ∴ potential to revolutionize interactivity in Web marketplaces
 - B2B, ...

What's Doable Today in rule-based agent contracting, based on our approach to rule representation (“SweetDeal”)

- Communicate: with deep shared semantics
 - XML, inter-operable with same sanctioned inferences
 - \leftrightarrow heterogeneous rule systems / rule-based agents
- Execute contract provisions:
 - infer; ebiz actions; authorize; ...
- Modify easily: contingent provisions
 - default rules; modularity; exceptions, overriding
- Reason about the contract/proposal
 - hypotheticals, test, evaluate; tractably
 - (also need “solo” decision making/support by each agent)

Approach:

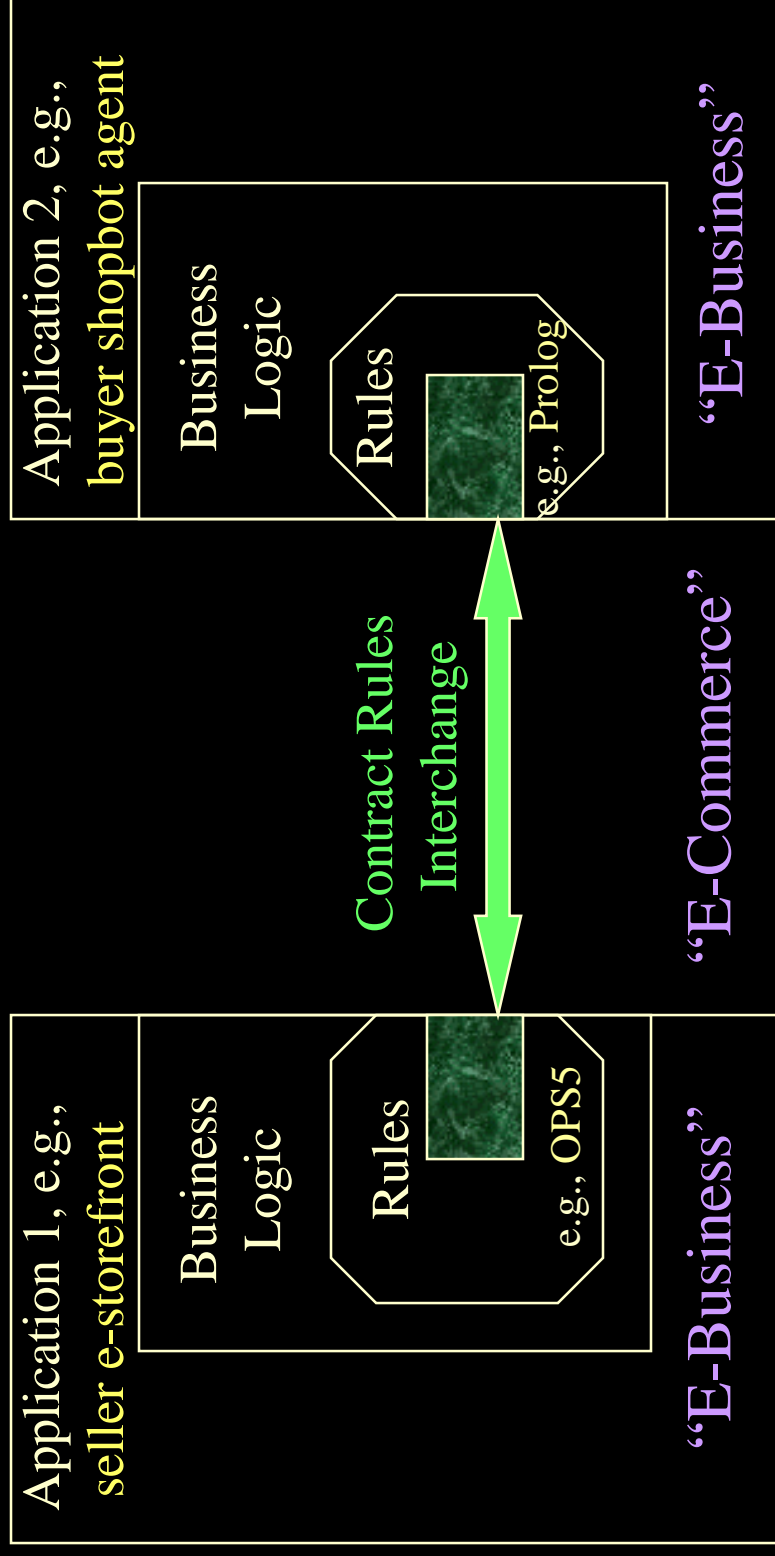
Rule-based Contracts for E-commerce

- 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.

Our SweetDeal System

- SWEET = Semantic WEB Enabling Technology
 - software components, theory, approach
 - pilot application scenarios, incl. [contracting](#) ([SweetDeal](#))
- Uses/contributes *emerging standards* for XML and knowledge representation:
 - RuleML (co-founder)
 - WebOnt ontologies (W3C)
- Uses *repositories* of business processes and contracts
 - MIT Process Handbook (Sloan IT)
 - legal/regulatory sources: law firms, ABA, CommonAccord, ... *Suggestions welcome!!*

Contract Rules across Applications / Enterprises

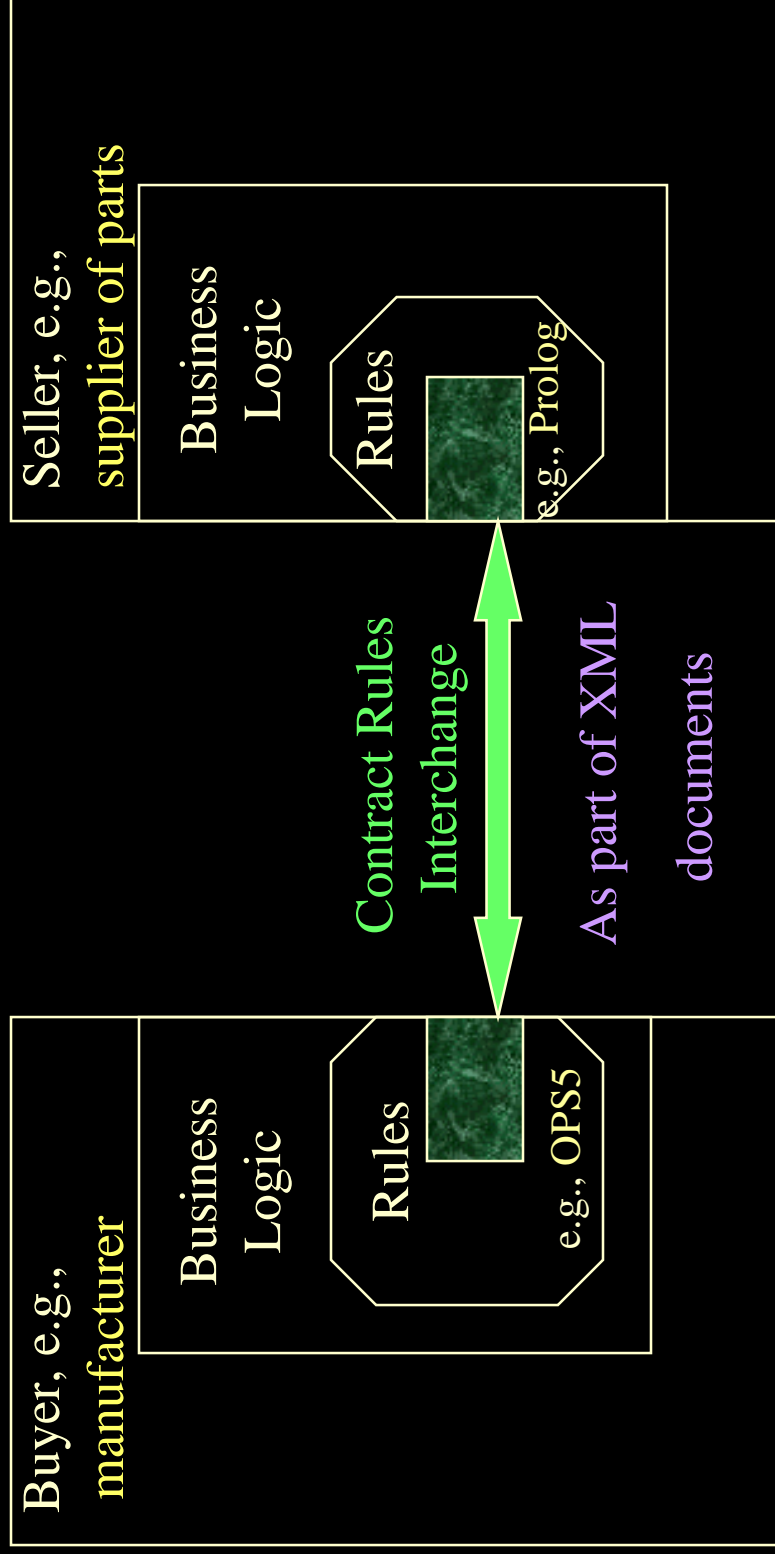


Contracting parties integrate e-businesses via shared rules.

Examples of Contract Provisions Well-Represented by Rules in Agent Deal Making

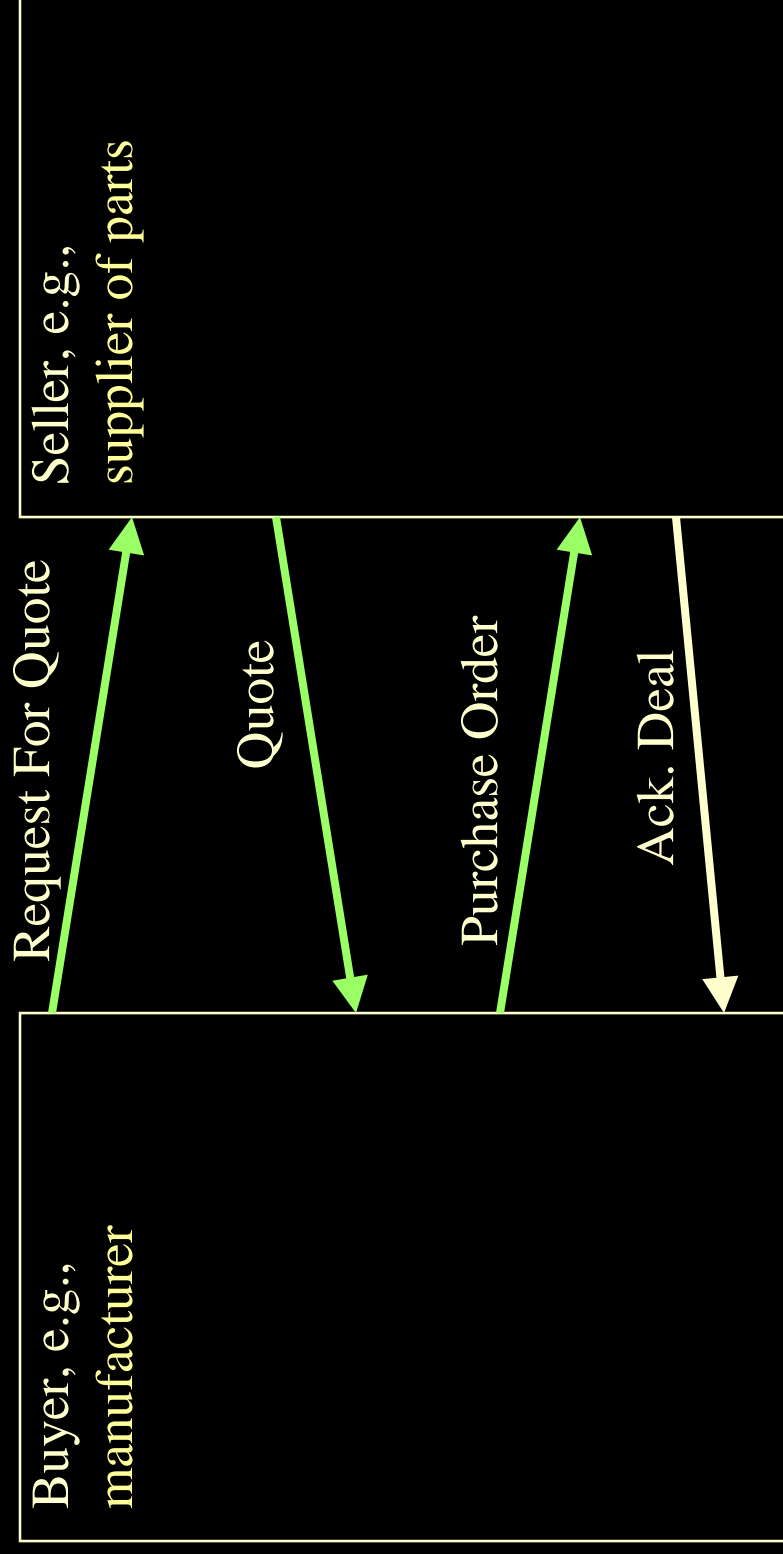
- Product descriptions
 - Product catalogs: properties, conditional on other properties.
- Pricing dependent upon: delivery-date, quantity, group memberships, umbrella contract provisions
- Terms & conditions: refund/cancellation timelines/deposits, lateness/quality penalties, ordering lead time, shipping, creditworthiness, biz-partner qualification, service provisions
- Trust
 - Creditworthiness, authorization, required signatures
- *Buyer Requirements (RFQ, RFP) wrt the above*
- *Seller Capabilities (Sourcing, Qualification) wrt the above*

Contract Rules during Negotiation

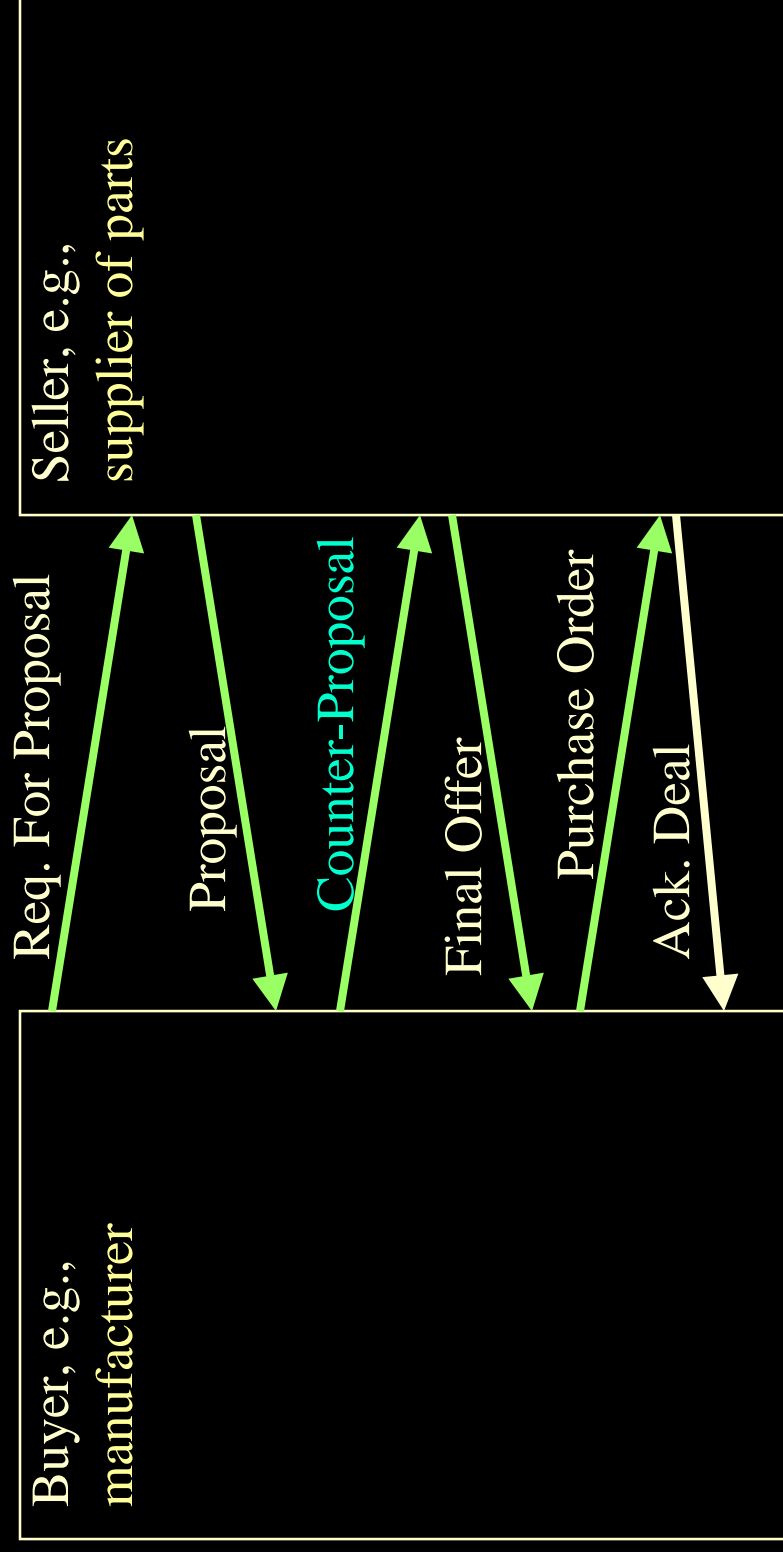


Contracting parties NEGOTIATE via shared rules.

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!

Negotiation Example -- XML Encoding of Rules in RuleML

- `<rulebase>`
- `<imp>`
- `<_rlab>usualPrice</_rlab>`
- `<_head>`
- `<clit>`
- `<_opr><rel>price</rel></_opr>`
- `<ind>per_unit</ind>`
- `<var>PO</var>`
- `<ind>$60</ind>`
- `</clit>`
- `</_head>`
- `<_body> ... (see next page) </_body>`
- `</imp>`
- `...`
- `</rulebase>`

Negotiation Example -- XML Encoding of Rules in RuleML, Continued

- `<_body>`
- `<andb>`
- `<fclit>`
- `<_opr><rel>purchaseOrder</rel></_opr>`
- `<var>PO</var>`
- `<ind>supplierCo</ind>`
- `<var>AnyBuyer</var>`
- `</fclit>`
- `<fclit>`
- ...
- `</fclit>`
- ...
- `</andb>`
- `</_body>`

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

4/30/2002

by Benjamin Groszof 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, incl. Web services
 - Agents, Business Processes, Workflow
 - E-Commerce: ebXML, ...
 - Industry-Specific
 - *Legal XML*
- *Law: Electronic Signatures, ...*
- *Reusable Contract doc's on Web: CommonAccord, our work, ...*

Our Current Research Directions

- SweetRules: prototyping of SCLP RuleML inferencing, translation, **knowledge integration**
 - ontologies; justifications; queries; using Web protocols to invoke procedural actions; Jess.
- In SweetDeal, using SweetRules: deals about **Web services**, using MIT Process Handbook; integrating shallowly automated legal text; reputations; exception handling and management of risk contingencies; financial services; P3P privacy policies; and distributed trust management incl. for security authorization.
- *I.e.*, **Business Intelligence on the Semantic Web**

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

Speaker Bio

- Benjamin Grosf is Douglas Drane Assistant Professor in Information Technology at MIT Sloan School of Management. His research is to create and study knowledge-based information technology for e-commerce applications, including high-level business/agent communication, information integration, contracts/negotiation, trust, product descriptions, business rules/policies, Web services, and e-marketplaces. The pioneer of inter-operable XML business rules and of their application to contracting, he co-leads the RuleML emerging industry standards effort on inter-operable XML/RDF rules. He is PI currently for a project in the DARPA Agent Markup Language (DAML) initiative, to create Semantic Web technology and explore its business applications.
- Previously, he was a senior research scientist at IBM T.J. Watson Research Center (12 years there), where most recently he conceived and led IBM CommonRules (V3.0 currently on IBM alphaWorks) and co-led its application piloting for rule-based XML agent contracting in EECOMS, a \$29Million NIST industry consortium project on manufacturing supply chain management. His notable technical contributions also include fundamental advances in rule-based security authorization, conflict handling for rules, rule-based intelligent agents, and integration of rules with machine learning. He is author of over 30 refereed publications, two major software releases, and a patent. His recent service includes co-chairing the AAAI (National Conference on Artificial Intelligence) Workshops on AI in E-Commerce (1999) and Knowledge-Based E-Markets (2000). His background includes 2 years in software startups, PhD in Computer Science (specialty AI) from Stanford University, and BA in Applied Mathematics from Harvard University.

4/30/2002

by Benjamin Grosf copyrights reserved

OPTIONAL SLIDES FOLLOW

4/30/2002

by Benjamin Grosof copyrights reserved

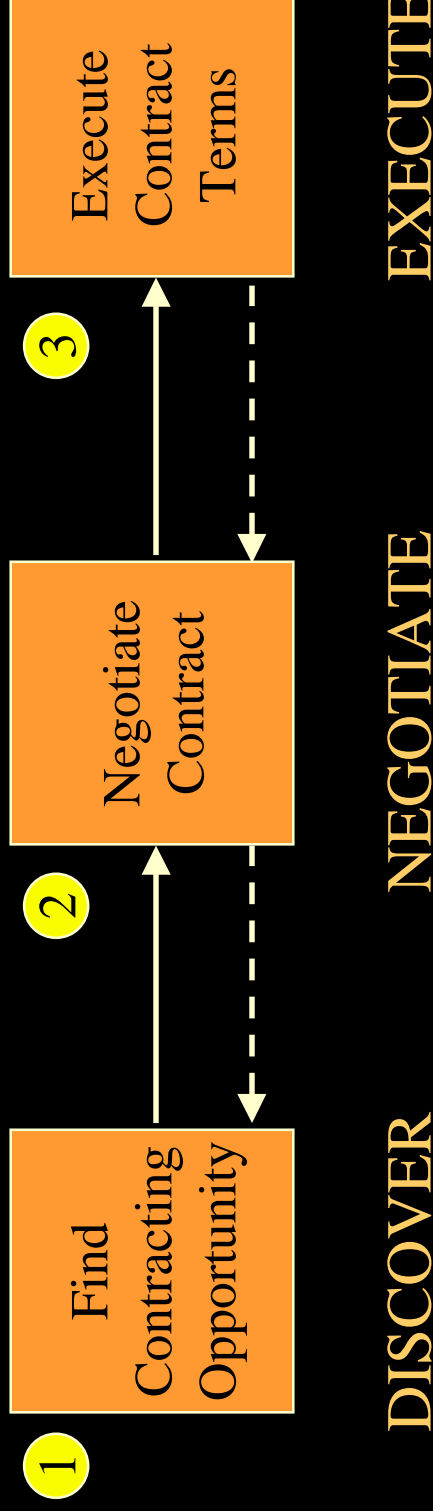
Intelligent Agents in Web E-Commerce

- *Today:* especially in the discovery phase of shopping
 - sales agents: recommend products, target ads
 - buyer agents: find vendors; compare offers on price, delivery, and availability
- *Coming soon to a world near you:...*
 - billions/trillions of agents
 - ...with smarts: knowledge gathering, reasoning, economic optimization
 - ...**doing our bidding**
 - but with some autonomy

Automating Contracting

- “Contract” in broad sense: = offering or agreement.
- “Automate” in deep sense: =
 - 1. Communicatable automatically.
 - 2. Executable within appropriate context of contracting parties’ business processes.
 - 3. Evaluable automatically by contracting parties.
 - “reason about it” .
 - 4. Modifiable automatically by contracting parties.
 - negotiation, auctions.

Contracting 1-2-3



- Applies to any contracting, electronic or not.
- May iterate or interleave these steps.
- Boundaries not necessarily sharp.