Semantic Rules for Policies and Services on the Web: Techniques, Business Applications, and Standards

Prof. Benjamin Grosof

MIT Sloan School of Management

Information Technologies group

http://ebusiness.mit.edu/bgrosof

Invited Presentation (1-hour) at (alphabetically):

- *Google Labs, Mountain View, CA, USA, Dec. 12, 2005, hosted by Peter Norvig*
- *IBM Almaden Research Center, San Jose, CA, USA, Dec. 13, 2005, hosted by Kiran Mehta and Ronald Fagin*
- *Oracle Corp., Redwood City, CA, USA, Dec. 7, 2005, hosted by Ashok Malhotra and Daniela Florescu*

Quickie Bio of Presenter Benjamin Grosof

- MIT Sloan professor since 2000
- 12 years at IBM T.J. Watson Research; 2 years at startups
- PhD Comp Sci (AI), Stanford;   BA Applied Math Econ, Harvard
- Leader for 3 major software projects: 1 open source, 2 commercial
- Semantic+web+services   is main research area:
  – Rules as core technology
  – Business Applications, Implications, Strategy:
    - e-contracting incl. ad,shop,monitor; trust; finance; …
  – Overall knowledge representation, e-commerce, intelligent agents
    - Learning. Hybrid techniques with onto, prob, induction.
- Co-Founder, Rule Markup Language Initiative – the leading emerging standards body & design in semantic web rules (http://www.ruleml.org)
- Area Editor, Semantic Web Services Initiative – which coordinates world-wide SWS research and early standards (http://www.swsi.org)

Resources

- This slideset
  - Some of it will only be skimmed in this presentation
- Author's website (http://ebusiness.mit.edu/bgrosof)

  … see especially there:
  - Recent talks (including this one soon), not just papers
  - ISWC-2005 Tutorial slideset
  - SweetRules toolset (http://sweetrules.projects.semwebcentral.org)

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Semantic Rules News

News recently:

- Fundamental theory and technique breakthroughs, e.g.:
  - Declarative logic programs (LP) basis for interoperability, then webized → RuleML standards design (2001-)
  - Courteous LP prioritized defaults, robust modular merging
  - Description LP ontology integration
  - Production LP interoperability+semantics for production rules, declarative procedural attachments for actions and queries
  - SweetRules V2 open source toolset platform (2004-)
- Large US, EU research projects (DAML, WSMO) focus on rules   (DARPA Agent Markup Language;   Web Service Mediation Ontology)

Semantic Rules News (cont.'d)

News recently:

- W3C forms Rule Interchange Format WG, full standards effort, after holding a Workshop (2005)

- OMG forms standards efforts on production rules, rule management

- Semantic Web Services Framework design (2005) focuses on rules

- Rule-based Policy area heats up in web services, semantic web, incl. at Oasis.   Oasis forms Semantic Execution Env. standards effort (2005).

- Semantic web rules workshop series becomes full research conference (RuleML-2005)   colocated with ISWC

- Gartner etc. reports on rules sector

Semantic Rules: Differences from Rules in the 1980's / Expert Systems Era

- Get the <u>KR</u> right   (knowledge representation)
  - More <u>mature</u> research understanding
  - <u>Semantics</u> independent of algorithm/implementation
  - <u>Cleaner</u>; avoid general programming/scripting language capabilities
  - Highly <u>scaleable</u> <u>performance</u>; better algorithms; choice from interoperability
  - Highly <u>modular</u> wrt updating; use prioritization
  - → Highly <u>dynamic</u>, <u>scaleable</u> <u>rulebase authoring</u>: distributed, integration, partnering

- Leverage <u>Web</u>, esp. XML
  - Interoperable syntax
  - Merge knowledge bases

- Embeddable
  - Into <u>mainstream</u> software development environments (Java, C++, C#); not its own programming language/system (cf. Prolog)

- Knowledge <u>Sharing</u>: intra- or inter- enterprise
- <u>Broader</u> set of Applications

"Semantic"

- "Semantic" in "semantic rules" and "semantic web" means:
  - 1. Knowledge-based
  - … and …
  - 2. Having meaning independent of algorithm and implementation
  - I.e., equipped with an interoperable conceptual abstraction
  - … based on <u>declarative</u> <u>knowledge representation</u> (<u>KR</u>)
  - (vs. procedural, dependent on inferencing control strategy, inferencing engine)

Concept of Knowledge Representation (KR)

- A knowledge representation S is defined as a triple (LP, LC, |=), where:

  - LP is a formal language of sets of premises (i.e., premise expressions)

  - LC is a formal language of sets of conclusions (i.e., conclusion expressions)

  - |= is the <u>entailment</u> relation.

    - Conc(P,S) stands for the set of conclusions that are entailed in KR S by a set of premises P

    - We assume here that |= is a functional relation.

- Heritage of KR concept: AI, DB areas of comp sci; earlier: logic from math, phil.; programming languages foundations

Example of Entailment: Mortality

- In First-Order Logic (FOL) KR:
    - Let P be the premises:
    - $\forall ?X.\ human(?X) \Rightarrow mortal(?X).$
    - human(Socrates).

    - In FOL, P entails (among others) the conclusion:
        - mortal(Socrates).

    - Notation:
        - "$\forall$" means "for all".
        - "?" Prefixes a logical variable.

Example of Entailment: Sunday Stroll

- In Bayesian Probability KR:
  - Let P be the premises:
    - prob(rainySunday) = 0.4.
    - prob(funSunday | rainySunday)   = 0.3.
    - prob(funSunday | ¬rainySunday) = 0.9.
  -
  - In this KR, P entails (among others) the conclusion:
    - prob(funSunday) = 0.66.

Example of Entailment: Discounting

- In the Courteous Logic Programs KR (e.g., RuleML):

  Let P be the premises:

  - {loyald}   discount(?cust, RamadaHotel, 10percent)
               ← memberOf(?cust, AAA).
  - {seniord} discount(?cust, RamadaHotel, 25percent)
               ← age(?cust, ?x) and greaterThan(?x, 64).
  - overrides(seniord, loyald).

  - ⊥ ← discount(?c, ?h, ?y) and discount(?c, ?h, ?z) | (?y ≠ ?z).
  - memberOf(Faisal, AAA).
  - age(Faisal, 72).

  - In this KR, P entails (among others) the conclusion:
        discount(Faisal, RamadaHotel, 25percent).

Example of Discounting, cont.'d

In the more general Production Logic Programs KR:

Suppose one adds the rule:

– @emailCouponAd(?cust, RamadaHotel, ?x)

$\leftarrow$ discount(?cust, RamadaHotel, ?x).

Then P entails the action (i.e., sanctions a call to an attached procedure):

@emailCouponAd(Faisal, RamadaHotel, 25percent).

RuleML Example: Markup and Tree

'The **discount** for a *customer* buying a *product* is **5.0 percent**
if the *customer* is **premium** and the *product* is **regular**.',
discount(?customer,?product,"5.0 percent") ← premium(?customer) ∧
regular(?product);

```
<imp>
  <head>
    <atom>
     <opr><rel>discount</rel></opr>
     <tup><var>customer</var>
          <var>product</var>
          <ind>5.0 percent</ind></tup>
    </atom>
  </head>
  <body>
    <and>
      <atom>
        <opr><rel>premium</rel></opr>
        <tup><var>customer</var></tup>
      </atom>
      <atom>
        <opr><rel>regular</rel></opr>
        <tup><var>product</var></tup>
      </atom>
    </and>
  </body>
</imp>
```



```
tup    is an ordered tuple.
```
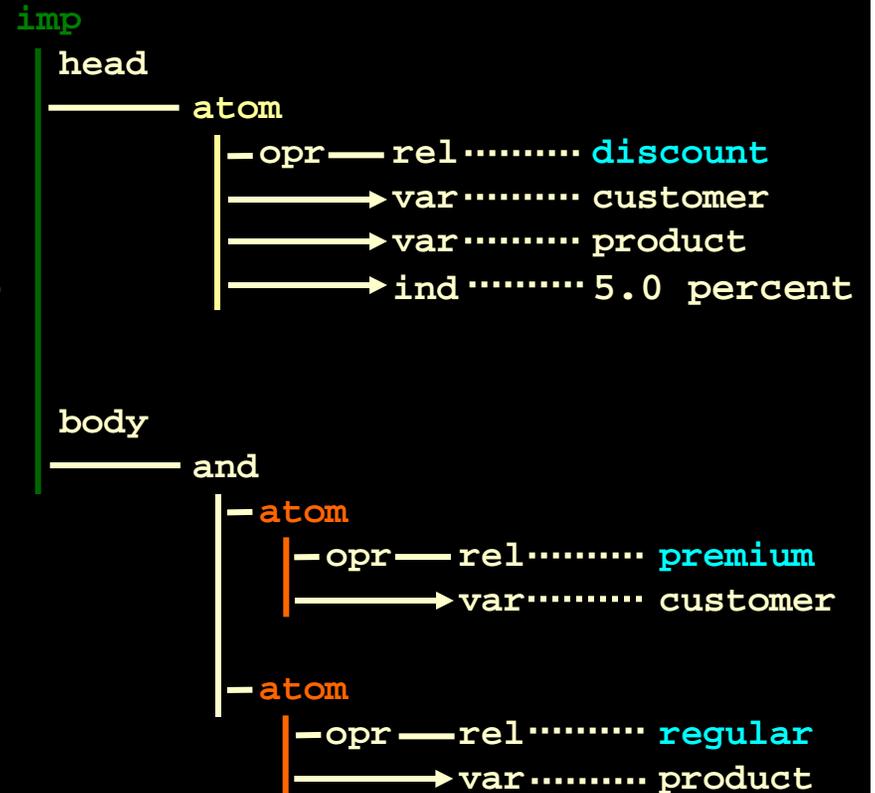
KR: What's the Game?
Desiderata

- Expressiveness: what can be said
  - useful, natural, complex enough

- Syntax: encoding data format -- e.g., in XML
  - easy enough to edit and communicate, by computers and by humans

- Semantics: principles of sanctioned inference, independent of reasoning algorithms:
  - clear, useful, natural, and understandable enough

- Computational Tractability (esp. worst-case): scale up in a manner qualitatively similar to relational databases: computation cycles go up as a polynomial function of input size

- Reasoning algorithms (compute the entailed conclusions):
  - sound (correct), complete, efficient, clear, and simple enough to engineer

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
 - Cheaper, Faster, Better; EAI and B2B; Early Verticals

The Semantic Web

- The 1st generation, the Internet, enabled disparate machines to exchange data.

- The 2nd generation, the World Wide Web, enabled new applications on top of the growing Internet, making enormous amounts of information available, in <u>human-readable</u> form, and allowing a revolution in new applications, environments, and <u>B2C</u> e-commerce.

- The next generation of the net is an "agent-enabled" resource (the "Semantic Web") which makes a huge amount of information available in <u>machine-readable</u> form creating a revolution in new applications, environments, and <u>B2B/EAI</u> e-commerce.

…by enabling "agent" communication at a Web-wide scale.
- "Agent" = <u>knowledge-based</u> application

Next Generation Web

Semantic Web Services

Semantic Web techniques

Automated Knowledge Bases

Rules (RuleML)

Ontologies (OWL)

Databases (SQL, XQuery, RDF)

Web Services techniques

API's on Web

(WSDL, SOAP)

Two interwoven aspects:
Program: Web Services
Data: Semantic Web

XML

First Generation Web
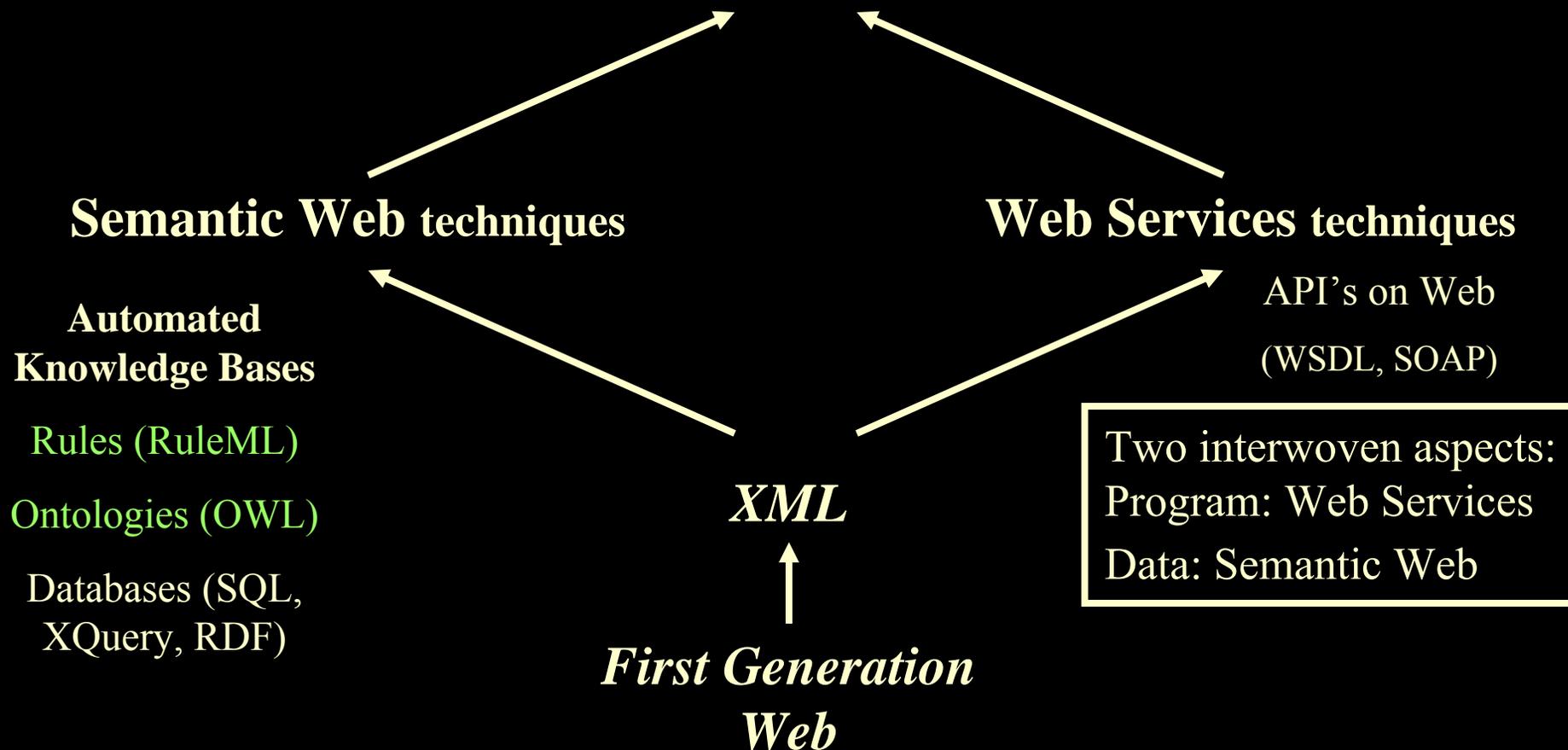
Vision of Evolution: Agents in Knowledge-Based E-Markets

Coming soon to a world near you:...

- billions/trillions of agents (= k-b applications)

- ...with smarts: knowledge gathering, reasoning, economic optimization

- ...doing our bidding
  - but with some autonomy

- *A 1st step: ability to communicate with sufficiently precise shared meaning... via the SEMANTIC WEB*

Semantic Web: concept, approach, pieces

- Shared semantics when interchange data   ∴ knowledge

- **Knowledge Representation** (cf. AI, DB) as approach to semantics
  - Standardize KR syntax, with KR theory/techniques as backing

- Web-exposed <u>Databases</u>:   SQL;   XQuery (XML-data DB's)
  - Challenge: share DB schemas via meta-data
  - RDF: "Resource Description Framework" W3C standard
    - Meta-data low-level mechanics: unordered directed graphs (vs. ordered trees)
    - RDF-Schema extension: simple class/property hierarchy, domains/ranges

- <u>Ontology</u> = formally defined vocabulary & class hierarchy
  - <u>OWL</u>: "Ontologies Working Language" W3C standard
    - Subsumes RDF-Schema and Entity-Relationship models
    - Based on Description Logic (DL) KR   ~subset of First-Order Logic (FOL))

- <u>Rules</u> = if-then logical implications, facts   ~subsumes SQL DB's
  - <u>RuleML</u>: "Rule Markup Language" emerging standard
    - Based on Logic Programs (LP) KR   ~extension of Horn FOL
    - Also provide FOL KR

Beware Narrow Usage of "Semantic Web"

- *Some people* use "semantic web" to mean *only*:
    stuff that uses RDF and OWL.

    … E.g., often W3C does this.

- We use the broader sense, as does the overall SW R&D community.

Exploding Research Interest in SW

Since 2002: …

- International Semantic Web Conference (ISWC) formed
  - Grown to 400+ researchers
- Became 2nd largest topic area of the International Conference on the World Wide Web (WWW)
  - (1st is Search, i.e., Google etc.)
- Specialized conferences formed: e.g., RuleML
- Major Research Programs in US and EU
- Professional Societies Chapters formed: e.g., AIS SIG
- Journals formed: e.g., J. Web Semantics
- Several industry standards efforts   (some done)

Big Questions
about the New Generation Web

- What are the critical features/aspects of the new technology?

- What business problems does it help solve?

- What are the likely innovation evolution paths, and associated entrepreneurial opportunities?

Our Research Aspects/Questions about the Semantic Web

- Core technologies: Requirements, concepts, theory, algorithms, standards?
  - <u>Rules in combination with ontologies</u>; probabilistic, decision-/game-theoretic

- Business applications and implications: concepts, requirements analysis, techniques, scenarios, prototypes; strategies, business models, market-level evolution?
  - <u>End-to-end e-contracting, finance, trust</u>; …

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Flavors of Rules Commercially Most Important today in E-Business

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

1. <u>Relational databases, SQL</u>: Views, queries, facts are all rules.
   - XQuery, SPARQL emerging.   SQL99 even has recursive rules.
2. <u>Production rules</u> (OPS5 heritage): e.g.,
   - Fair Isaac, ILOG, Haley, etc.:   rule-based Java/C++ objects.
3. <u>Event-Condition-Action rules</u> (loose family similar to PR), cf.:
   - business process automation / workflow tools.
   - active databases; publish-subscribe.
4. <u>Prolog</u>. *"logic programs" as a full programming language.*

5. *(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.
- Known advantages of rules vs. general code
  - separable business logic, more reusable across app.'s, life cycle
  - good for loose coupling cf. workflow
  - good for representing contingent behavior of services/processes.
  - 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.

Semantic Rules: Use Cases *from our research*

- Contracts/negotiation, advertising/discovery
  - E-procurement, E-selling
  - Pricing, terms & conditions, supplier qualification, …

- Monitoring:
  - Exception handling, e.g., of contract violations
    - Late delivery, refunds, cancellation, notifications
  - Notifications, personal messaging, and other workflow

- Trust Policies: authorization, confidentiality & privacy, security, access control
  - E.g., financial services, health care
    - *Extensive analysis of business case/value*

- Semantic mediation: rule-based ontology translation, context-based information integration

SWS and Rules   Summary

*** SWS Tasks Form 2 Distinct Clusters,**

each with associated Central Kind of Service-description

Knowledge and Main KR

1. <u>Security/Trust</u>, <u>Monitoring</u>, <u>Contracts</u>, <u>Advertising/Discovery</u>, <u>Ontology-mapping Mediation</u>

    - Central Kind of Knowledge: <u>Policies</u>

    - Main KR: <u>Nonmon LP</u>   (rules + ontologies)

2. <u>Composition</u>, <u>Verification</u>, <u>Enactment</u>

    - Central Kind of Knowledge: <u>Process Models</u>

    - Main KR: <u>FOL</u> (axioms + ontologies)

        - + <u>Nonmon LP</u> for ramifications (e.g., cf. Golog)

- Thus RuleML & SWSF specify both Rules, FOL

    – Fundamental KR Challenge: "Bridging" Nonmon LP with FOL

        - SWSF experimental approach based on hypermon. [Grosof & Martin]

Rule-based Semantic (Web) Services

- Rules/LP in appropriate combination with DL as KR, for RSWS
  - DL good for <u>categorizing</u>:   a service overall, its inputs, its outputs

- Rules to describe <u>service process models</u>
  - rules good for representing:
    - <u>preconditions</u> and <u>postconditions</u>, their contingent relationships
    - <u>contingent</u> behavior/features of the service more generally,
      - e.g., exceptions/problems
  - familiarity and naturalness of rules to software/knowledge engineers

- Rules to specify <u>deals about services</u>: cf. e-contracting.

Rule-based Semantic Web Services

- Rules often good to <u>executably specify</u> service process models
  - e.g., business process automation using procedural attachments to perform side-effectful/state-changing actions ("effectors" triggered by drawing of conclusions)
  - e.g., rules obtain info via procedural attachments ("sensors" test rule conditions)
  - e.g., rules for knowledge translation or inferencing
  - e.g., info services exposing relational DBs

- <u>Infrastructural</u>: rule system functionality as services:
  - e.g., inferencing, translation

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Talk Mode: the MIT Firehose

Shortened from a 90-minute talk
⇒ Some skimmed

New Fundamental Rule KR Theory I
that enables Key Technical Requirements for SWS

- 1. Courteous Logic Programs: [Grosof]

    KR to combine rules from many sources, with:
    - Prioritized conflict handling to enable consistency, modularity; scaleably
    - Interoperable syntax and semantics

- 2. Situated Logic Programs: [Grosof]

    KR to hook rules (with ontologies) up to (web) services
    - Rules use services, e.g., to query, message, act with side-effects
    - Rules constitute services executably, e.g., workflow-y business processes

New Fundamental Rule KR Theory II
that enables Key Technical Requirements for SWS

- 3. Reference Ontologies from Rules Via URI Names [Grosof]

- 4. Description Logic Programs: [Grosof, Horrocks, Volz, & Decker]

   KR to combine LP (RuleML) rules on top of DL (OWL) ontologies,

   with:
   - Power in inferencing   (including for consistency)
   - Scaleability of inferencing
   - Approach: Analyze and exploit the Intersection of DL and LP (within FOL)

New Fundamental Rule KR Theory III
that enables Key Technical Requirements for SWS

- ## 5. Courteous Inheritance: [Grosof & Bernstein]
  - OO default inheritance as Courteous LP
  - Used to Leverage Process Handbook, & other Legacy OO Knowledge, to create SW service ontologies

- ## 6. Production Rules as LP: [Grosof]
  - OPS5-heritage production rules as Situated Courteous LP
  - Find and fix fundamental weakness in chaining through negation in Rete-based inferencing
  - Unify commercially most important and fast-growing rule families

- ## 7. Hypermonotonic Reasoning: [Grosof (in-progress)]
  - Unify Nonmon LP KR with FOL KR
  - Nonmon LP as sound & incomplete wrt FOL

New Fundamental Rule KR Theory
that enables Key Technical Requirements for SWS

In 1985-94:

- Prolog interoperable with relational DB; LP extends core-SQL [many]
- Richer logical connectives, quantifiers [Lloyd & Topor]
- "Well Founded" Semantics for Negation-As-Failure [Van Gelder et al; Przmusinski]
- Hilog quasi-higher order expressiveness, meta-syntax flexibility [Kifer et al.]
- Frame syntax cf. F-Logic [Kifer *et al.*]

In 1995-2004:

- Courteous LP: prioritized conflict handling [Grosof]
  - Robust, tractable, modular merging & updating
- Situated LP: hook rules up to services [Grosof]
- Description LP: combine Description Logic ontologies [Grosof *et al.*]
- Courteous Inheritance: combine OO default ontologies [Grosof *et al.*]
- Production Rules as LP: interoperate [Grosof *et al.*]
  - Declarative LP as interoperable core between commercial families [Grosof *et al.*]
- Hypermonotonic Reasoning: combine with FOL [Grosof (in-progress)]

Production Logic Programs:
A New Fundamental Rule KR Approach

In 2005:

- Production extension of LP:
  - actions and tests appear directly within rules (procedural attachments)
  - Generalizes Situated LP a bit, and reformulates it more familiarly
- Theory & algorithms achieving semantic interoperability of
    {core Production Rules} ← → declarative LP
  - Handles negation correctly, by stratifying PR agenda control strategy
  - 1st declarative semantics for Production Rules

- Combines with all the other features: Courteous, …

- → "Production LP" as umbrella LP KR approach

Venn Diagram: Expressive Overlaps among KR's

First-Order Logic

Description Logic

Horn Logic Programs

Logic Programs

Description Logic Programs

(Negation As Failure)

(Procedural Attachments)
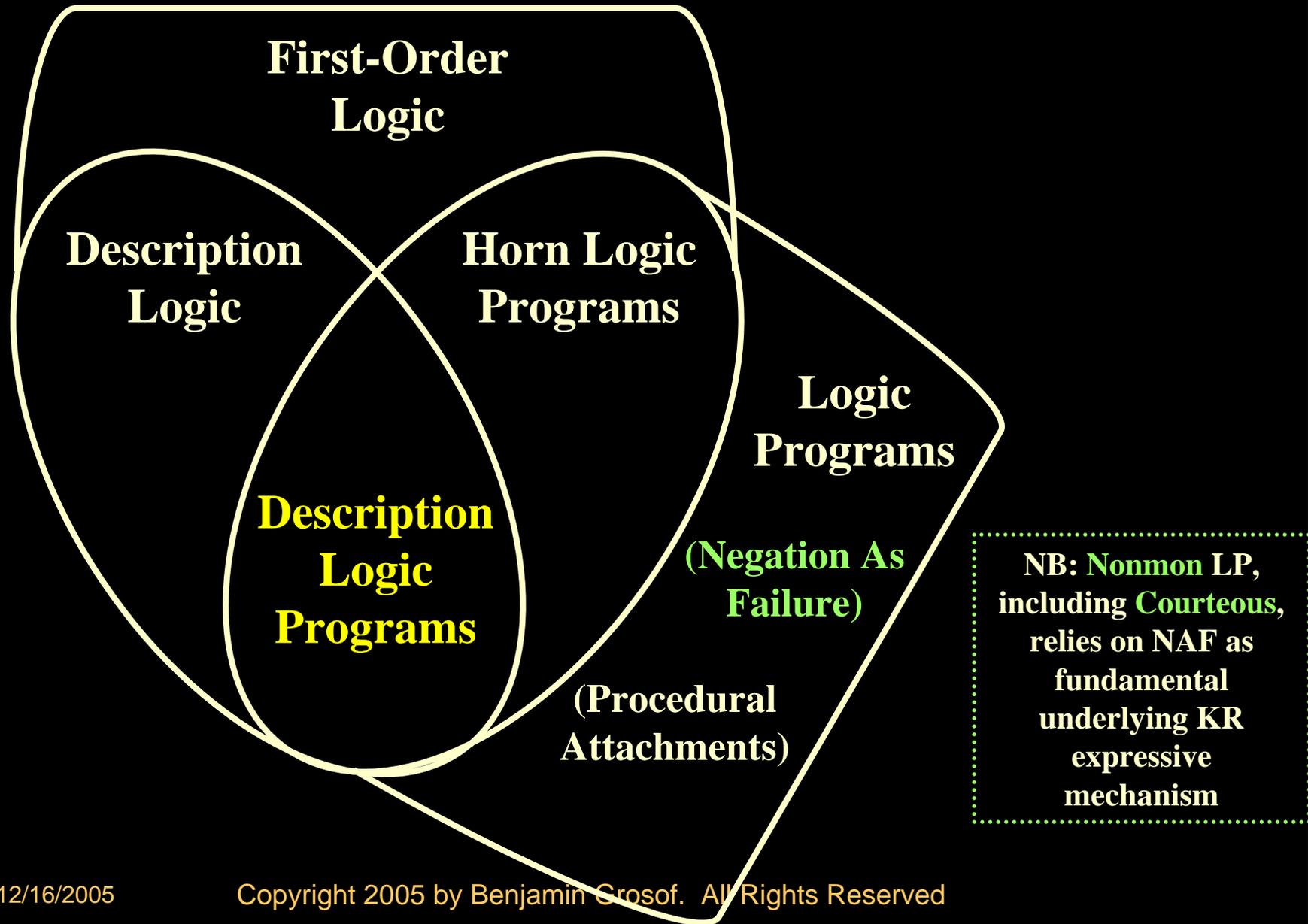
Criteria for Contract Rule Representation

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.
- Modularity and locality in revision.

3

- Declarative semantics.
- Logical non-monotonicity: default rules, negation-as-failure. } **OLP**
  - essential feature in commercially important rule systems.
- Prioritized conflict handling.
- Ease of parsing. ⟶ **Courteous**
- Integration into Web-world software engineering. } ⟶ **XML**
- Procedural attachments.
- ⟶ **Situated**

New Analysis:
Key Technical Requirements for SWS

- 1. Combine rules with ontologies, from many web sources, with:
    - Rules on top of ontologies
    - Interoperability of heterogeneous rule and ontology systems
    - Power in inferencing
    - Consistency wrt inferencing
    - Scaleability of inferencing

- 2. Hook rules (with ontologies) up to web services
    - Ex. web services: enterprise applications, databases
    - Rules use services, e.g., to query, message, act with side-effects
    - Rules constitute services executably, e.g., workflow-y business processes
    - Rules describe services non-executably, e.g., for discovery, deal negotiation
    - On top of web service process models, coherently despite evolving messiness

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Some Answers to:
"Why does SW Matter to Business?"

- 1. "Death. Taxes. Integration." - They're always with us.

- 2. "Business processes require communication between organizations / applications." - Data and programs cross org./app. boundaries, both intra- and inter- enterprise.

- 3. "It's the *automated knowledge* economy, stupid!" - The world is moving towards a knowledge economy. And it's moving towards deeper and broader automation of business processes. The first step is automating the use of <u>structured</u> knowledge.
  - Theme: *reuse* of knowledge across multiple tasks/app's/org's

Strategic Business Foci in our SW Research

- Knowledge-based Services Engineering: intra- and inter- enterprise

- Target "killer app" known for 30 years: do better job of EDI

- Challenges:
  - Ease of development, deployment ↑
  - Reuse of knowledge ↑
  - ⇒ life cycle costs ↓ , agility ↑

- Starting with: Policies
  - Using recent theory breakthroughs in semantic rules
  - E.g., for end-to-end contracting and authorization (incl. security)

- Starting with: EAI as well as B2B

SW Rules: Use Cases *from our research*

- Contracts/negotiation, advertising/discovery
  - E-procurement, E-selling
  - Pricing, terms & conditions, supplier qualification, …

- Monitoring:
  - Exception handling, e.g., of contract violations
    - Late delivery, refunds, cancellation, notifications
  - Notifications, personal messaging, and other workflow

- Trust Policies: authorization, confidentiality & privacy, security, access control
  - E.g., financial services, health care
    - *Extensive analysis of business case/value*

- Semantic mediation: rule-based ontology translation, context-based information integration

EECOMS Example of SCM Policy Rules: Ordering Lead Time

- 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 > A.

Courteous LP's:
Ordering Lead Time Example

{leadTimeRule1} orderModificationNotice(?Order,14days)

$\leftarrow$ preferredCustomerOf(?Buyer,?Seller) $\wedge$

purchaseOrder(?Order,?Buyer,?Seller) .

{leadTimeRule2} orderModificationNotice(?Order,30days)

$\leftarrow$ minorPart(?Buyer,?Seller,?Order) $\wedge$

purchaseOrder(?Order,?Buyer,?Seller) .

{leadTimeRule3} orderModificationNotice(?Order,2days)

$\leftarrow$ preferredCustomerOf(?Buyer,?Seller) $\wedge$

orderModificationType(?Order,reduce) $\wedge$

orderItemIsInBacklog(?Order) $\wedge$

purchaseOrder(?Order,?Buyer,?Seller) .

overrides(leadTimeRule3 , leadTimeRule1) .

$\perp \leftarrow$ orderModificationNotice(?Order,?X) $\wedge$

orderModificationNotice(?Order,?Y) | (?X $\neq$?Y) .

Transforming Business Reporting

PRESS ROOM   EVENTS   CONTACT

Welcome to XBRL International

16 - 17 NOVEMBER 2004   10th International
BRUSSELS   CONFERENC

Financial Reporting Goes Global – XBRL and IFRS Working Together

For more information, please visit the Conference Website register today.

XBRL is a language for the electronic communication of business financial data which is set to revolutionise business reporting arou the world. It provides major benefits in the preparation, analysis communication of business information. It offers cost savings, gr efficiency and improved accuracy and reliability to all those involv in supplying or using financial data.

XBRL stands for eXtensible Business Reporting Language. It is o a family of "XML" languages which is becoming a standard means communicating information between businesses and on the inter

XBRL is being developed by an international non-profit consortiu approximately 250 major companies, organisations and governme agencies. It is an open standard, free of licence fees. It is already being put to practical use in a number of countries and implementations of XBRL are growing rapidly around the world.

This site provides information about the nature, uses and benefit XBRL. It explains how individuals and companies can join the eff move forward and make use of the language.

Left navigation menu:

WHAT IS XBRL
NEWS ABOUT XBRL
THE SHOWCASE
XBRL IN ACTION
XBRL AND BUSINESS
TECHNICAL INFORMATION
EDUCATION AND TRAINING
ABOUT THE ORGANISATION
HOW TO JOIN
MEMBERS' AREA
NEWS FOR MEMBERS
TECHNICAL DEVELOPMENT
WORKING GROUPS
SUPPLY CHAINS

PRESS ROOM EVENTS CONTACT US JUR

Equational Ontological Conflicts

Gross Profit = Net Sales – Cost of Goods

Gross Profit = Net Sales – Cost of Goods – Depreciation

Price = Nominal Price + Shipping

Price = Nominal Price + Shipping + Tax

" heterogeneity in the way data items are *calculated* from other data items *in terms of definitional equations*"

Slide also by A. Firat and S. Madnick

Comparing Prices From Multiple Vendors/Sources using ECOIN

Price: Nominal
Product Code: Numeric

Price Equations

pokemon	13.3
starwars	30.1

Context Mediator

Price:Nominal + Tax+Shipping
Product Code: Alpha

Price:Nominal + Tax
Product Code: Numeric

Query

Prices of Products Cheaper in eToys

eToys

pokemon	17
starwars	45

Kid's World

123456	20
234567	40
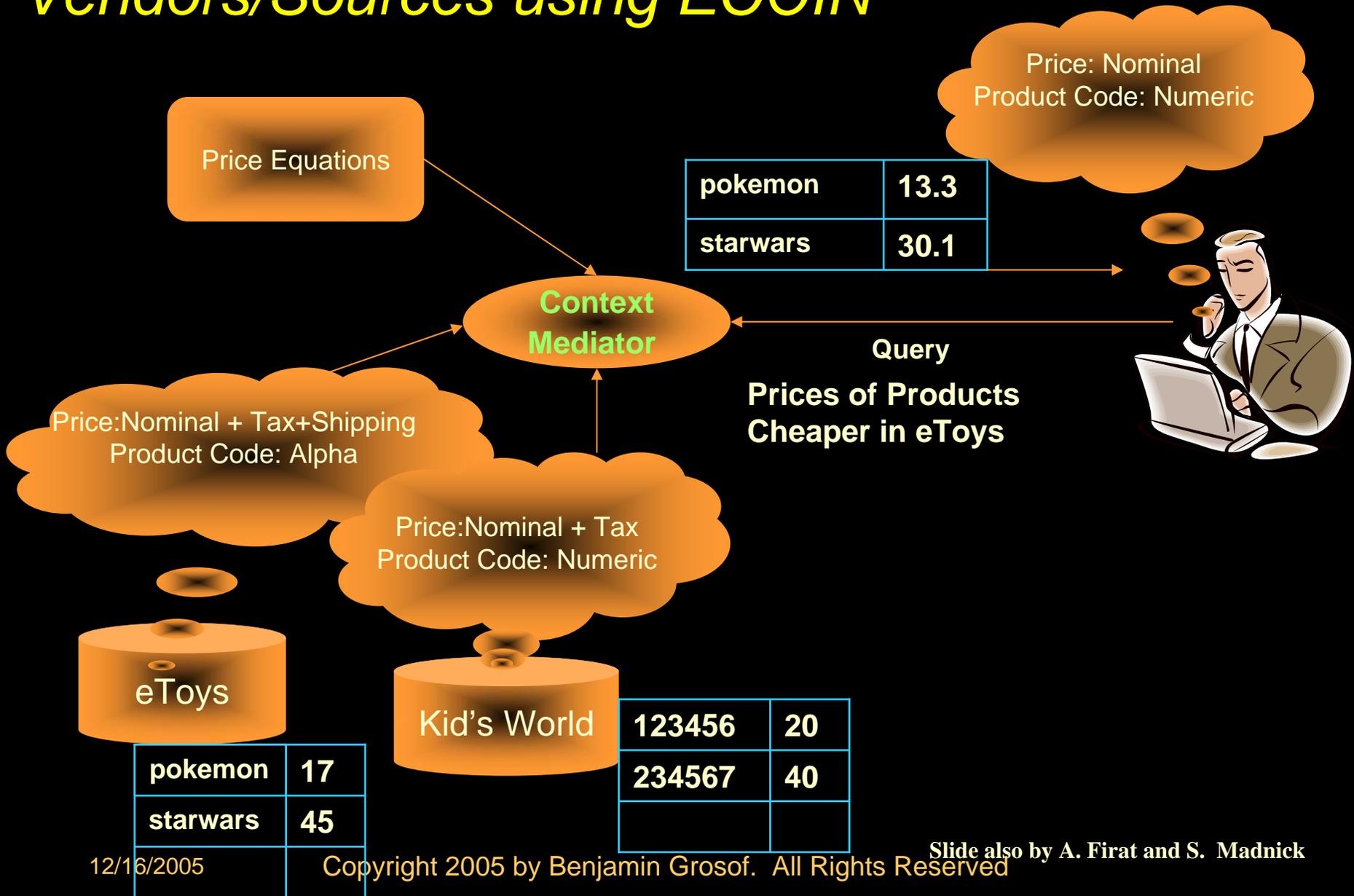
Approach: ECOIN

- Extended <u>CO</u>ntext <u>IN</u>terchange, developed at MIT Sloan

  - *[Firat, Madnick, & Grosof] (Best Paper Award WITS-2002)*

- Context-based loosely-coupled integration

  Extends the Context Interchange (COIN) framework also developed at MIT

- Symbolic Equation Solving using Constraint Logic Programming

  Integrates symbolic equation solving techniques with abductive logic programming

- *In-progress:* Utilizing RuleML and OWL in ECOIN

Slide also by A. Firat and S. Madnick

End-to-End E-Contracting Tasks

- Discovery, advertising, matchmaking
  - Search, sourcing, qualification/credit checking
- Negotiation, bargaining, auctions, selection, forming agreements, committing
  - Hypothetical reasoning, what-if'ing, valuation
- Performance/execution of agreement
  - Delivery, payment, shipping, receiving, notification
- Problem Resolution, Monitoring
  - Exception handling

SweetDeal 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 during Negotiation

Buyer, e.g., manufacturer

Seller, e.g., supplier of parts

Business Logic

Business Logic

Rules

Rules

Contract Rules Interchange

e.g., OPS5

e.g., Prolog

As part of XML

documents
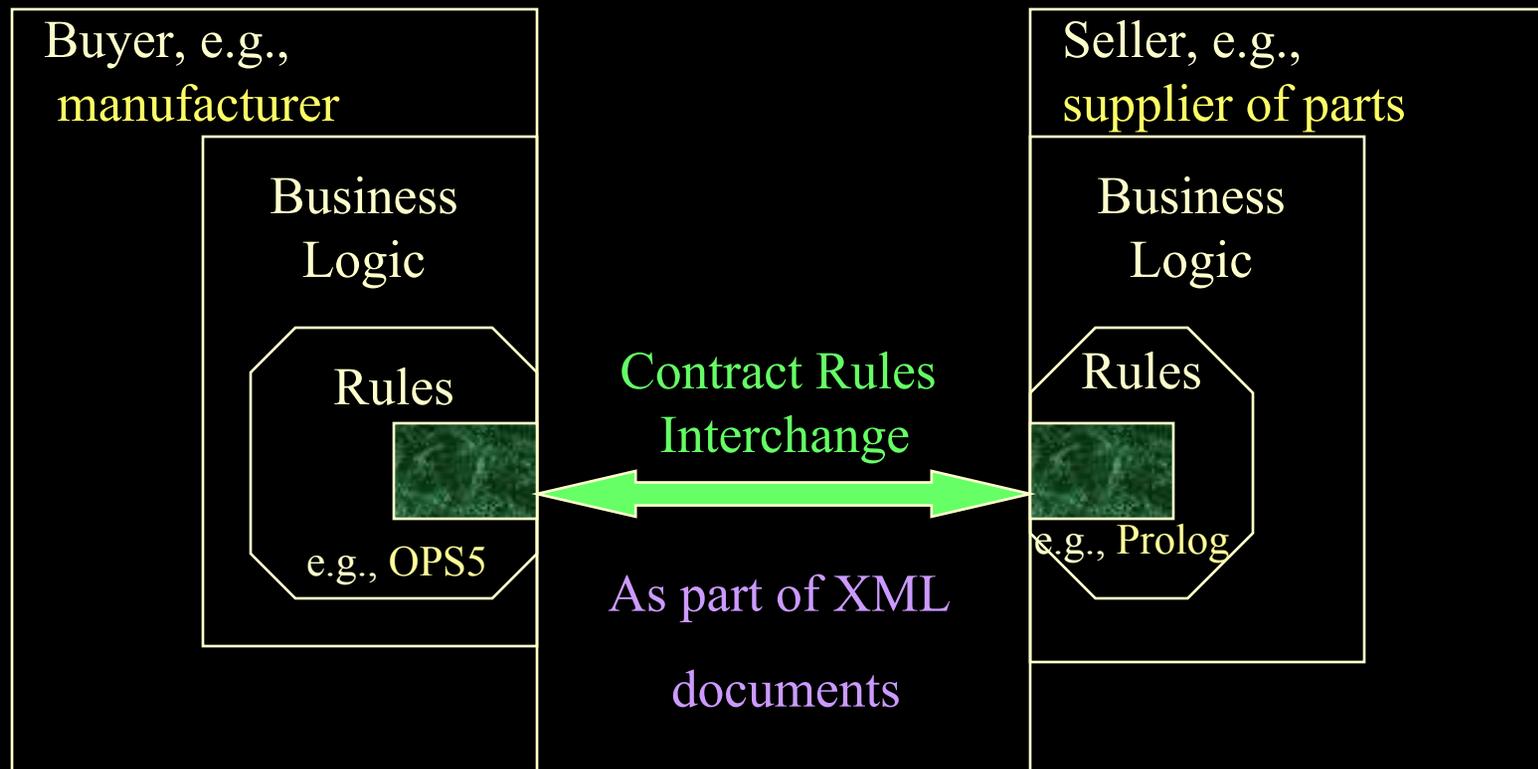
Contracting parties NEGOTIATE via shared rules.

Examples of Contract Provisions
Well-Represented by Rules
in Automated 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, <u>service</u> provisions

- Trust
  - Creditworthiness, authorization, required signatures

- *Buyer Requirements (RFQ, RFP) wrt the above*

- *Seller Capabilities (Sourcing, Qualification) wrt the above*

Exchange of Rules Content during Negotiation: example

Buyer, e.g., manufacturer	Req. For Proposal →	Seller, e.g., supplier of parts

Req. For Proposal

Proposal

Counter-Proposal

Final Offer

Purchase Order

Ack. Deal
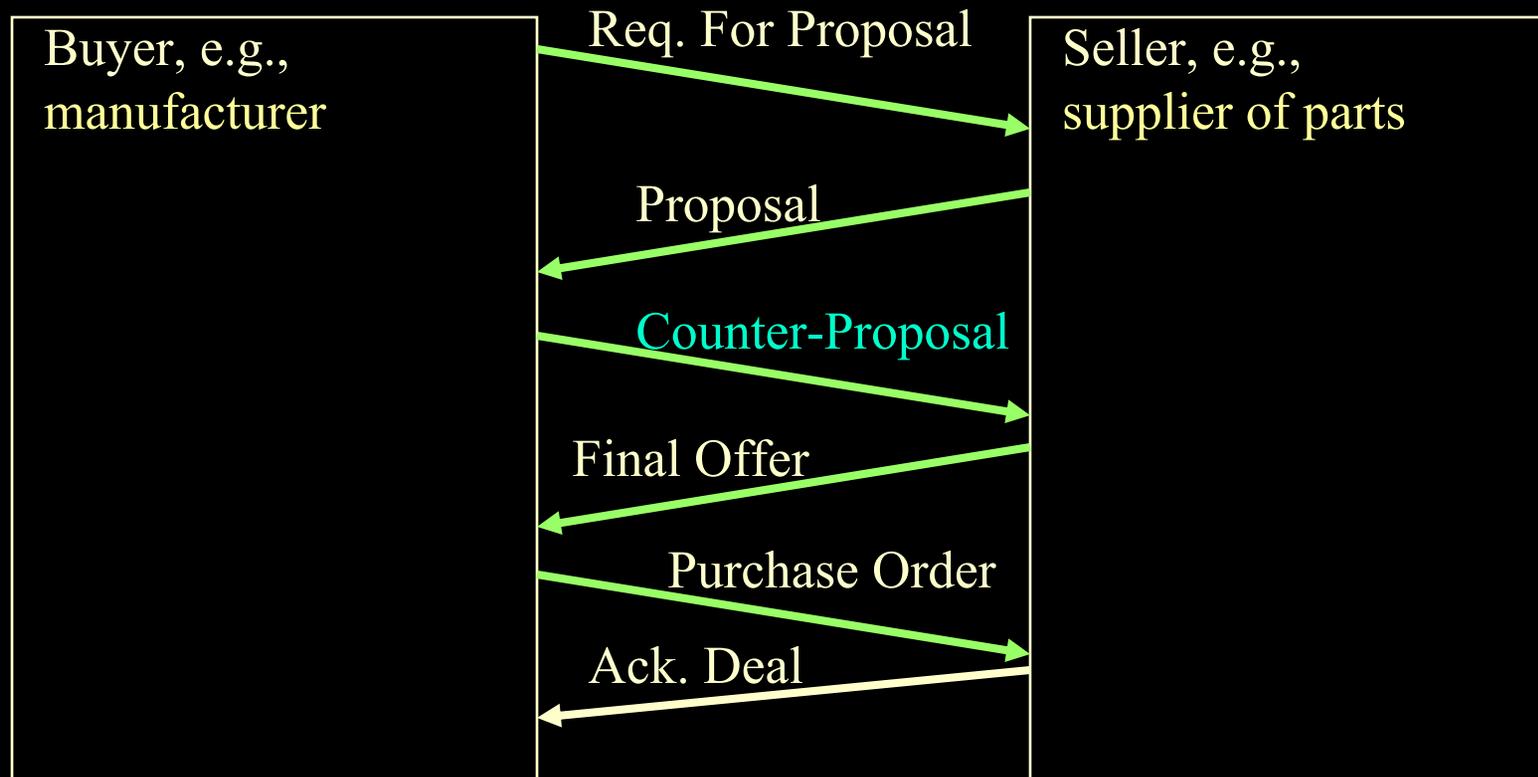
Example: E-Contract Proposal from supplierCo to manufCo

- …

  {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!**

XML Encoding of Rules in   RuleML

- <rulebase>
-   <imp>
-    <rlab>usualPrice</_rlab>
-    <head>
-     <atom>
-      <opr><rel>price</rel></_opr>
-      <ind>per_unit</ind>
-      <var>PO</var>
-      <ind>$60</ind>
-     </atom>
-    </head>
-   <body>   … *(see next page)*   </_body>
-   </imp>
- …
- </rulebase>

SweetDeal V2   Demo Outline

- SweetDeal E-Contracting Application using SweetRules (supply chain)
  - SCLP RuleML that includes OWL ontologies
  - Contract proposals/final-agreements are SCLP RuleML rulebases that reference/include OWL ontologies
  - Humans edit & communicate, supported by automated agents
  - Proposal evaluation supported by inferencing
  - Agreed business process is executable via inferencing+action

What Can Be Done with the Rules *in contracting, & negotiation, based on our SweetDeal approach to rule representation*

- Communicate: with deep shared semantics
  - via RuleML, inter-operable   with same sanctioned inferences
  - ⇔ <u>heterogeneous</u> rule/DB systems / rule-based applications ("agents")

- Execute contract provisions:
  - infer;   <u>ebiz actions</u>;   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)*

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Overview of RuleML Today I

- RuleML Initiative (2000--)   http://www.ruleml.org
  - Dozens of institutions (~35), researchers; esp. in US+Canada, EU
- Mission priorities:
  1. Enable semantic exchange of rules/facts between most commercially important rule systems
     - Production rules, relational databases, Prolog, Event-Condition-Action rules
  2. Synergize with RDF, OWL (& other relevant web standards as arrive)
  3. Enable rule-based semantic web services, e.g., policies
- Standards specification: current version V0.8+
  - 1st version 2001; basic now fairly stable
- Logical Knowledge Representation at core of semantics
  - Declarative Logic Programs (LP) & First Order Logic (FOL) … Webized
  - Firm foundations in decades of R&D theory, algorithms, implementations

Overview of RuleML Today II

- A number of tools (~60 engines, translators, editors), demo applications. E.g., SweetRules open source platform.
- Very influential & lots of mindshare in cutting edge R&D community. 20,000+ Google Hits (as of Mar. 2004)
- Annual International Scientific Workshop since 2002
- Cooperating closely with the leading umbrella Web standards organizations and SW research efforts:
  - OMG – providing markup and semantics for production rules meta-model.
  - W3C – providing technical approach and industry partners for the new Rule Interchange Format working group.
  - Discussions well underway to launch Oasis standards working group.
  - Encouraged (and funded in part) by DARPA
  - Collaborating with Semantic Web Services Initiative (SWSL), Web Services Mediation Language (WSML) & REWERSE in EU

SweetRules Overview I

Key Ideas:

– Unite the commercially most important kinds of rule and ontology languages via a a new, common KR (Production LP) in a new standardized syntax (RuleML), including to cope with *heterogeneity* resolve contradictory *conflicts*.

   • Capture most of the useful expressiveness, interoperably and scalably.

– Combine a large *distributed* set of rule and ontology knowledge bases that each are *active:* each has a different *associated engine* for reasoning capabilities (inferencing, authoring, and/or translation).

– Based on recent fundamental KR theory advances, esp. Situated Courteous Logic Programs (SCLP) and Description Logic Programs.

   • Including semantics-preserving translations between different rule languages/systems/families, e.g., Situated LP ↔ production rules

Application Areas (prototyped scenarios):

– Policies and authorizations; contracting, supply chain management; retailing, customer relationship management; business process automation and e-services; financial reporting and information; etc.

Distributed Active Knowledge Bases

• heterogeneous rules / ontologies
• with associated inferencing, authoring, translation capabilities

New Integration Capabilities

Inferencing + Translation

Authoring + Testing

Reasoning Capabilities to Support Applications
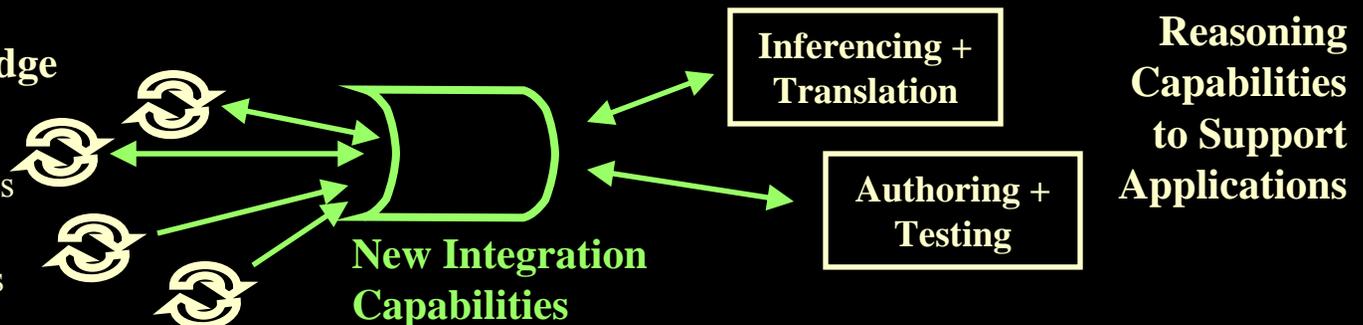
SweetRules   Overview II

- Concept and Architecture: Open Source Tools Platform for SW Rules and RuleML.   http://sweetrules.projects.semwebcentral.org (2004-)
  - Multi-institutional collaboration led by MIT Sloan, with 12+ other co.'s / univ.'s
- Capabilities:
  - Translation and interoperability between heterogeneous rule systems (forward- and backward-chaining) and their rule languages/representations of the most commercially important flavors (relational database / Prolog and production rules / event-condition-action)
  - Inferencing including via translation between rule systems
  - Authoring, Analysis, and testing of rulebases
  - Open, lightweight, extensible, pluggable architecture overall
  - Merge knowledge bases
    - Combine rules with ontologies, incl. OWL, OO default inheritance
  - Focus on kinds of rule systems that are commercially important
    - E.g., Jess production rules, XSB Prolog, IBM Common Rules, HP Jena, …
  - Highly scaleable performance by piggybacking on mature commercial implementations (e.g., Jess, XSB)
  - Automatically composes translators, inference engines

SweetRules Goals

- <u>Research vehicle</u>: embody ideas, implement application scenarios (e.g., contracting, policies)
  - Situated/Production Courteous Logic Programs (SCLP) KR
  - Description Logic Programs (DLP) KR which is a subset of SCLP KR
  - RuleML/SWRL

- <u>Proof of concept</u> for feasibility, including of <u>KR algorithms</u> and <u>translations</u> between heterogenous families of rule systems
  - Encourage others: researchers; industry esp. vendors
  - Scaleability; practicality

- <u>Catalyze/nucleate</u> SW Rules communal efforts on:
  - Tools, esp. open-source
  - Application scenarios / use cases, esp. in services

SweetRules   Context and Players

- Part of SWEET = "<u>S</u>emantic <u>WE</u>b <u>E</u>nabling <u>T</u>ools" (2001 –)
  - Other parts: … these use SweetRules …
    - SweetDeal for e-contracting
    - SweetPH for Process Handbook ontologies
- <u>Cross-institutional. Collaborators invited!</u>
  - Originated and coordinated by MIT Sloan since 2001
  - Code base: Java, XSLT; convenience shell scripts (for testing drivers)
  - Code by MIT, UMBC, BBN, Stanford, U. Zurich
  - Cooperating other institutions: U. Karlsruhe, IBM, NRC/UNB, SUNY Stonybrook, HP, Sandia Natl. Labs; RuleML Initiative
    - Collaboration on design of code by Stanford, U. Karlsruhe
  - Uses code by IBM, SUNY Stonybrook, Sandia Natl. Labs, HP, Stanford, Helsinki
  - Many more are good targets: subsets of Flora-2, cwm, KAON, JTP, SWI Prolog, Hoolet, Triple, DRS, ROWL, ...

- Fundamental KR: Situated Courteous Logic Programs (SCLP)   KR = Knowledge Representation

  - Horn

  - + Negation-As-Failure (<u>NAF</u>) = <u>Ordinary</u> LP

  - + <u>Courteous</u> prioritized conflict handling

    - overrides relation on rule labels, classical negation, mutex integrity constraints

  - + <u>Situated</u> sensing & effecting

    - Invoke external procedural attachments

    - Sensing = <u>tests/queries</u>; e.g., built-ins

    - Effecting = side-effectful <u>actions</u>, triggered by conclusions

SweetRules V2.0   Translators Graph

KIF **(FOL -subset)**

Courteous Compiler

CommonRules
(fwd. SCLP)

RuleML
(SCLP)

Jess/CLIPS
(prodn. ≡ fwd. SOLP)

XSB (bkw. OLP)

Smodels (fwd. OLP)

Process Handbook
(OO/frame def.-inh)

Jena-2
(fwd. Horn LP)

OWL (-DLP)
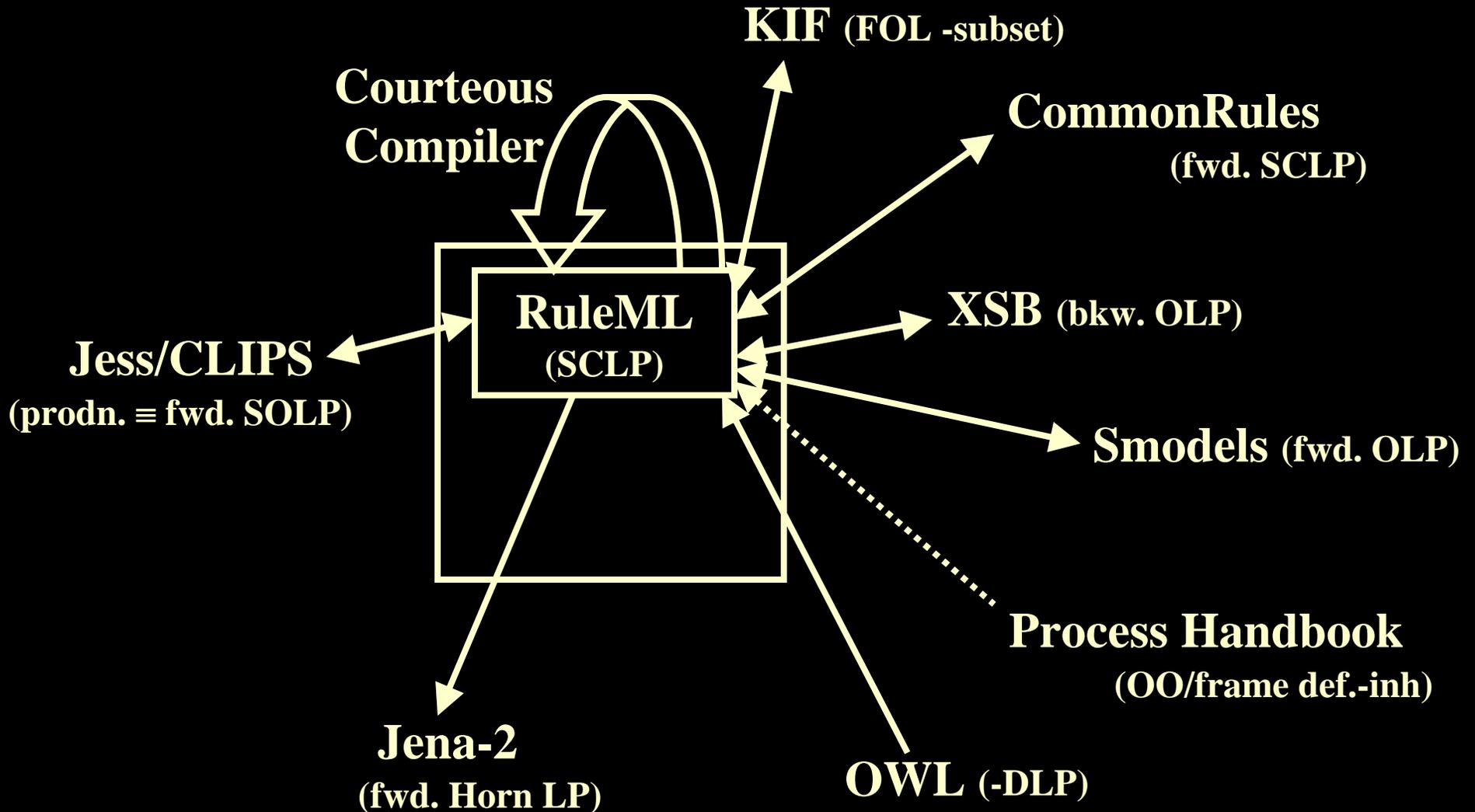
SweetRules Inferencing Capabilities: Overview

- Inferencing engines in RuleML/SWRL via translation:
  - Indirect inferencing:
    1. translate to another rule system, e.g., {XSB, Jess, CommonRules, or Jena}
    2. run inferencing in that system's engine
    3. translate back
  - Can use composite translators

SweetRules V2.0+: *Indirect Inferencing Engines*

Key: ↑ =
SweetRules
raises **power**

Courteous
Compiler

KIF **(FOL -subset)**

CommonRules
(fwd. SCLP)

↑fwd. SCLP & bkw. CLP
XSB **(bkw. OLP)**

↑fwd. SCLP
Jess/CLIPS
(prodn. ≡ fwd. SOLP)

RuleML
(SCLP)

SWRL
(Horn)

Smodels (fwd. OLP)

Process Handbook
(OO/frame def.-inh)

↑+ SWRL built-ins
Jena-2
(fwd. Horn LP)

OWL (-DLP)
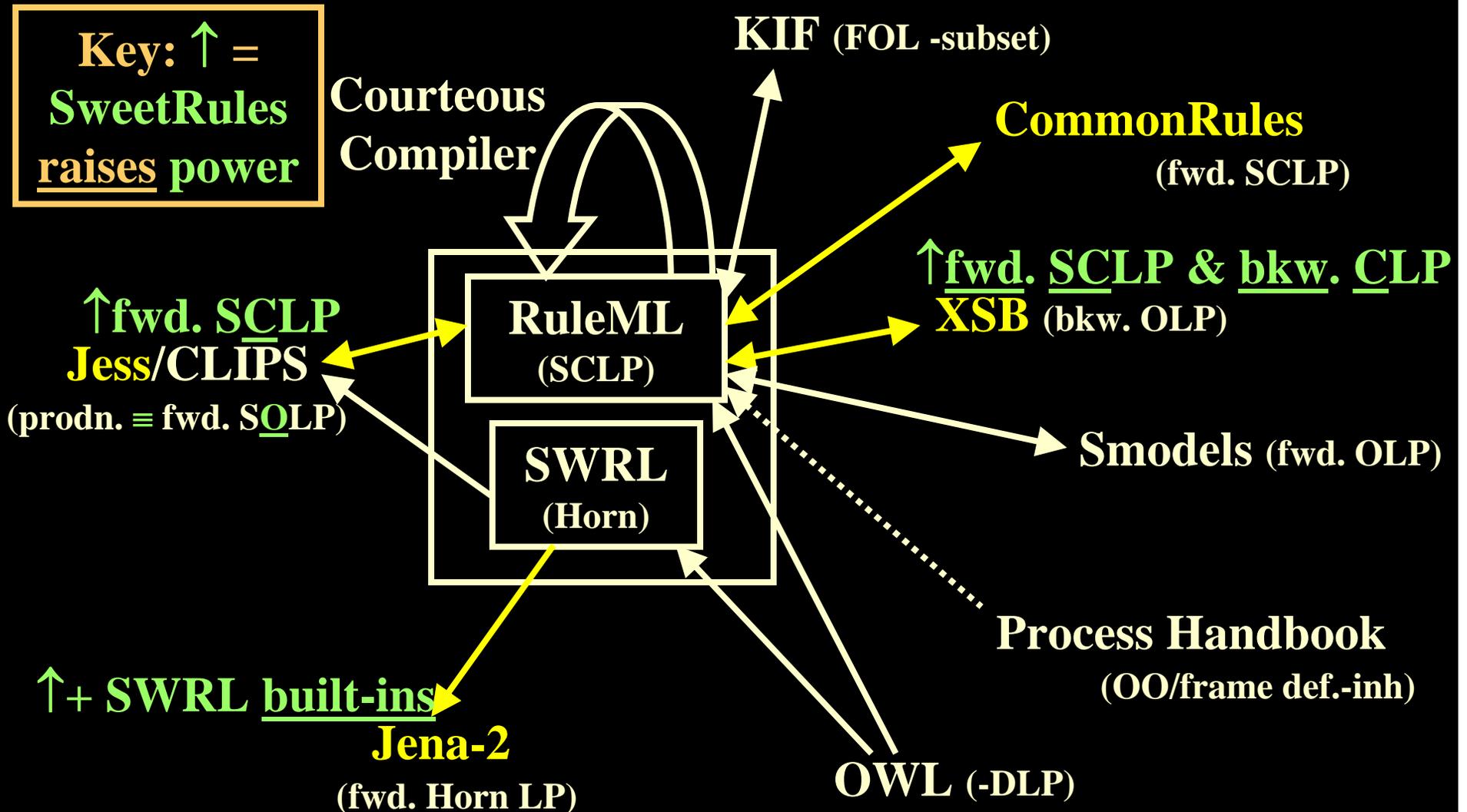
Novel Capabilities of SweetRules   I

- 1st Semantic interoperability between Production Rules and declarative LP
  - 1st semantic treatment of Production Rules
- 1st for several particular kinds of semantic interoperability between heterogeneous commercially important kinds of rules -- and ontologies too, e.g. …
- 1st: Production Rules,
- + Prolog (and thus essentially core SQL),
- + OWL/RDF (via Description LP approach)
- 1st: Via an emerging standards design for semantic rules on web: RuleML
- 1st: Supports WSDL actions in semantic rules – a true rule-based semantic web service system

Novel Capabilities of SweetRules II

- 1st: tool set <u>platform</u> for semantic rules on web
  - & in open source
- 1st: Based on Production LP KR approach, overall
- 1st: Inclusion merging for heterogeneous rulebases and ontology KBs, via such interchange language
- 1st: Indirect inferencing: design pattern and detailed design/implem.
- 1st: Dozens of particular translators, e.g., Jess, XSB, OW/RDF, CommonRules, Jena, KIF, Process Handbook
  - Pluggable and automatically composed
- 1st: Supports expressively powerful RuleML-based interoperability and inferencing
  - Courteous prioritized conflict handling
  - + Situated procedural attachments for actions and tests/queries -- cf. PR, + geeneralized
- 1st wrt several nonmon algorithms & capabilities:
  - Courteous + unrestricted non-stratified negation
  - Stratified negation in production rules
  - Non-stratified negation via production rules
  - Courteous extension of Prolog, Production Rules
- 1st wrt several procedural attachment algorithms & capabilities:
  - Actions extension of Prolog-based engine

Summary of SweetRules

- SweetRules V2: tool set platform
  - Supports expressively powerful RuleML-based interoperability and inferencing
  - and also SWRL. Basis: declarative Logic Programs KR at heart, + some FOL
  - Description LP technique for merging restricted OWL-DL into LP
  - Courteous LP prioritized conflict handling
  - Unrestricted (scoped) default negation
  - Production LP / Situated LP procedural attachments for actions and tests/queries cf. PR
    - has generalization to permit unbound such queries
  - Indirect inferencing: translate, infer in another rule system, translate back
  - 1st interoperability between Production Rules and declarative LP
  - Based on Production LP KR approach, overall
  - Translation/inferencing in Jess, XSB, OWL/RDF, CommonRules, Jena, KIF, more
  - Inclusion merging of heterogeneous rulebases and ontology knowledge bases
  - Dozens of translators, pluggable and automatically composed
  - Supports WSDL actions – a true rule-based semantic web service system

Objectives for Integrating Distributed SW Rules and Ontologies, Motivating SweetRules and its underlying theory+standards

BEFORE

AFTER

Contradictory conflict is globally contagious, invalidates all results.

Contradictory conflict is contained locally, indeed tamed to aid modularity.

Knowledge integration tackling the 5 D's (diversity, distributedness, disagreement, dynamism, & delay) is labor-intensive, slow, costly.
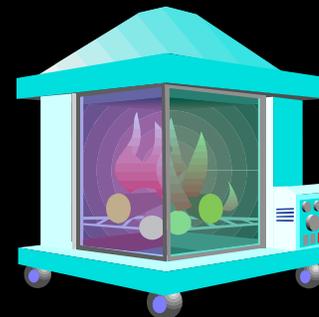
Knowledge integration is highly automated, faster, cheaper.
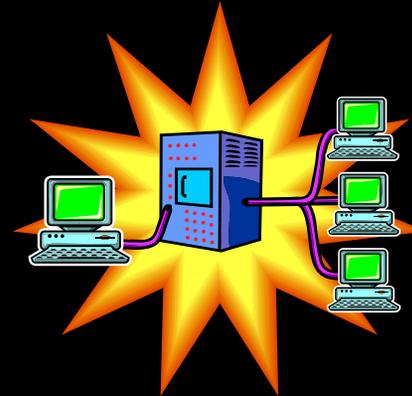
SweetRules: Use Cases Overview

- Contracts/negotiation, advertising/discovery
  - E-procurement, E-selling
  - Pricing, terms & conditions, supplier qualification, …

- Monitoring:
  - Exception handling, e.g., of contract violations
    - Late delivery, refunds, cancellation, notifications
  - Notifications, personal messaging, and other workflow

- Trust Policies: authorization, confidentiality & privacy, security, access control
  - E.g., financial services, health care
    - *Extensive analysis of business case/value*

- Semantic mediation: rule-based ontology translation, context-based information integration

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Advantages of Standardized SW Rules for Policies, e.g., Authorization/Security

- Easier Integration: with rest of business policies and applications, business partners, mergers & acquisitions
  - Enterprise integration, B2B
- Familiarity, training
- Easier to understand and modify by humans.
- Quality and Transparency of implementation in enforcement
  - Provable guarantees of behavior of implementation
    - Scaleability;   consistency, completeness, correctness
- Reduced Vendor Lock-in
- Expressive power
  - Principled handling of conflict, negation, priorities
- $\Rightarrow$ Agility, change management $\uparrow$

Advantages of SW Rules, cont'd:
Loci of Business Value
in Policy Management

- Reduced system dev./maint./training costs
- Better/faster/cheaper policy admin.
- Interoperability, flexibility and re-use benefits
- Greater visibility into enterprise policy implementation $\Rightarrow$ better compliance
- Centralized ownership and improved governance by Senior Management
- Rich, expressive policy management language allows better conflict handling in policy-driven decisions
- Strategic agility, incl. wrt business model

SWS Adoption Roadmap:
Some Strategy Considerations

- ## "Death. Taxes. Integration."

- Expect see beginning in a lot of B2B interoperability or
  heterogeneous-info-integration intensive (e.g., finance, travel)
  - Actually, probably 1st intra-enterprise, e.g., <u>EAI</u>

- Reduce costs of <u>communication</u> in procurement, operations, customer service, supply chain ordering and logistics

- <u>Agility</u>/speed/flexibility in business processes, supply chains

- "Killer app" target known for 30 years: do better job of EDI

Prospective SW Early Adopters: Areas by Industry or Task

- *We've discussed a number of industry or task areas:*
  - Manufacturing supply chain, procurement, pricing, selling, e-tailing, financial/business reporting, authorization/security/access/privacy policies, health records, credit checking, banking, brokerage, contracts, advertising, …

- Others:
  - travel "agency", i.e.: tickets, packages
    - See Trading Agent Competition, [M.Y. Kabbaj thesis]
  - military intelligence (e.g., funded DAML)

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Some Technical Directions for Research I

Core technology:

- Knowledge Services integration: pull/push/merge via web
  - Integrate with DBMS: XQuery; algorithms; SPARQL
  - Incremental Reasoning: Events, Updates; algorithms
- Integrate with Probabilistic and Induction
  - Integrate with Text and Unstructured info
  - Decision-theoretic, game-theoretic; data mining; OO/FOL Bayesian; dependency and locality
- LP KR other extensions: existentials, higher-order, equality, reification, …
- Hypermonotonicity: analysis of LP, merging; new KR's incl. disjunctive
- Constraints: satisfaction, optimization

Some Technical Directions for Research II

Applications esp. in policies and services:

- Trust policies for security, confidentiality, privacy, access control
- E-Contracting end-to-end reuse, power: incl.
  - advertising/discovery, shopping, contracts/negotiation, business process monitoring, regulations
- Policy Ontology, Services Ontologies, Relationship to C++/Java/C# Inheritance
- Web Services "Policy Management", "Contracts"
- Add semantics to existing standards: XBRL, XACML, ebXML, RosettaNet, EDI
- Biomedical: patient records, drug discovery, treatment safety
- Marketing, intelligence, supply chain, financial reporting, travel
- Financial sector, overall
- Business Value Analysis, Strategy, Roadmapping

Outline

- Intro: What are Semantic Rules, Semantic Web Services
  - Knowledge Representation; in XML; with Web Services

- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform

- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …

- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Resources

- This slideset
  - Some of it will only be skimmed in this presentation
- Author's website (http://ebusiness.mit.edu/bgrosof)

    ... see especially there:
  - Recent talks (including this one soon), not just papers
  - ISWC-2005 Tutorial slideset
  - SweetRules toolset (http://sweetrules.projects.semwebcentral.org)