Slideset 3 of "E-Commerce Applications of Semantic Web Services"

Benjamin Grosof

MIT Sloan School of Management,

http://ebusiness.mit.edu/bgrosof

EC '04 Conference Tutorial

5th ACM Conference on Electronic Commerce, May 17, 2004, New York City

Version Date: April 29, 2004

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Web Service -- definition

- *(For purposes of this talk:)*
- A procedure/method that is invoked through a Web protocol interface, typically with XML inputs and outputs
  - Add the flexibility of XML to the concepts of RPC
  - XML Tools support extra functionality required
- Purpose: Program integration across application and organizational boundaries
  - Needs commercial semantics

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 Databases: SQL; XQuery (XML-data DB's)
  - Challenge: share DB schemas via meta-data
- RDF: "Resource Description Framework" W3C proposed standard
  - Meta-data lower-level mechanics: unordered directed graphs (vs. ordered trees)
  - RDF-Schema extension: simple class/property hierarchy, domains/ranges
- Ontology = formally defined vocabulary & class hierarchy
  - OWL: "Ontologies Working Language" W3C proposed standard
    - Subsumes RDF-Schema and Entity-Relationship models
    - Based on Description Logic (DL) KR ~subset of First-Order Logic (FOL))
- Rules = if-then logical implications, facts ~subsumes SQL DB's
  - RuleML: "Rule Markup Language" emerging standard
    - Based on Logic Programs (LP) KR ~extension of Horn FOL

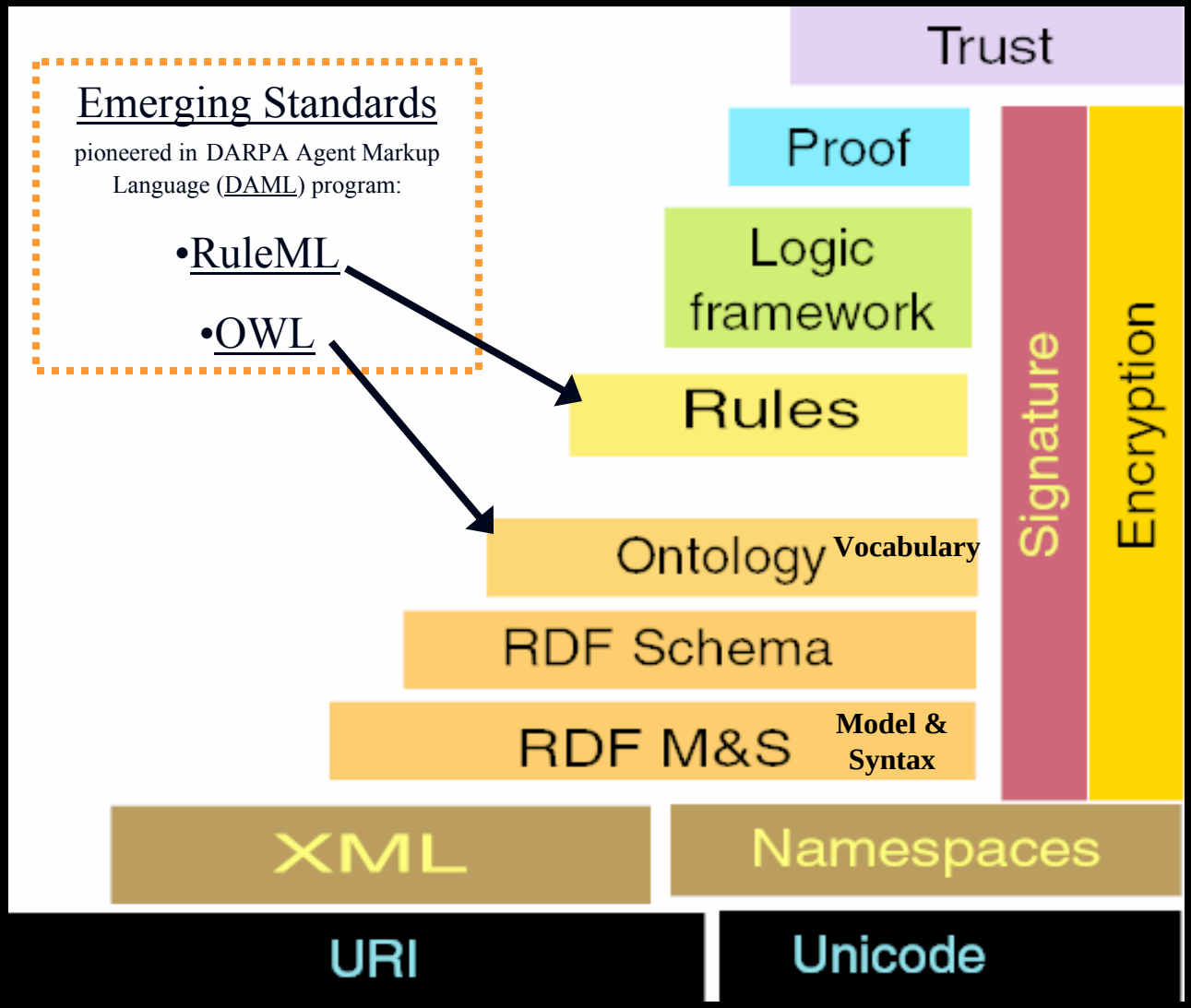
Some Semantic Web Advantages for Biz

- Builds upon XML's much greater capabilities (vs. HTML*) for structured detailed descriptions that can be processed automatically.
  - Eases application development effort for assimilation of data in inter-enterprise interchange
- Knowledge-Based E-Markets -- where Agents Communicate

  (Agent = knowledge-based application)
  - ∴ potential to revolutionize *interactivity* in Web marketplaces: B2B, …
- Reuse same knowledge for multiple purposes/tasks/app's
  - Exploit declarative KR; Schemas

- * new version of HTML itself is now just a special case of XML

W3C Semantic Web "Stack": Standardization Steps



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

SW: Research Players

- US: DARPA Agent Markup Language Program (DAML) program
- EU: OntoWeb program
- @MIT:
  - Sloan IT group: Grosof, Madnick, *et al.*
  - LCS / W3C advanced-dev.: Berners-Lee, *et al.*
- Number of companies:
  - HP, IBM, Adobe, Oracle, …
- 500+ basic researchers now working largely on it.
  - Research community has grown rapidly from a handful in 1999.

SW: Standards Players

- US-EU Joint Committee:
  - Early standards drafting
  - 1st focus: ontologies: DAML+OIL → W3C OWL
  - 2nd focus (current): rules: RuleML
- W3C: Semantic Web Activity
- Oasis: various incl. Security
- New efforts (currently in formation):
  - US-EU Joint Committee on Semantic Web Services
  - ISO: CommonLogic first-order logic (formerly KIF)

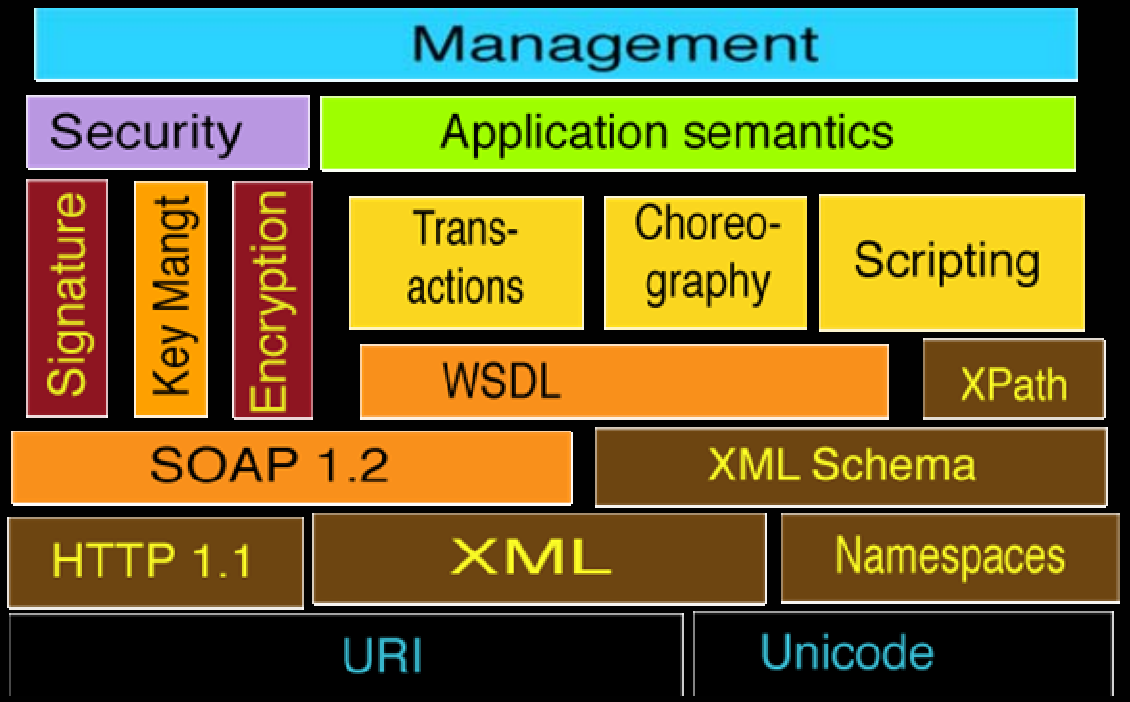
SW-Related: XQuery (XML Database Query Language)

- Goals:
  - a data model for generic "natively" XML documents,
  - a set of query operators on that data model,
  - and a query language based on these query operators
  - Queries operate on single documents or fixed collections of documents.
- What SQL is for relational databases, XQuery is for collections of XML docs. It's a W3C standard.
- Oracle, IBM, Microsoft, etc. already support some
  - Did not take off quickly – complex spec.
  - Now in major development.
  - Being pushed strongly to customers for 2006+ horizon as next major generation of enterprise data management tool.

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Web Services Stack outline



NOTES:

WSDL is a Modular Interface spec
SOAP is Messaging and Runtime
Also:
- UDDI is for Discovery
- BPEL4WS, WSCI, …
  are for transactions
- Routing, concurrency, …

Diagram courtesy Tim Berners-Lee: http://www.w3.org/2004/Talks/0309-ws-sw-tbl/slide6-0.html

WS Stack: some Acronym Expansion

- SOAP = simple protocol for XML messaging
- WSDL = protocol for basic invocation of Web Services, their input and output types in XML
- Choreography = higher-level application interaction protocols in terms of sequences of exchanged message types, contingent branching
  - There's now a W3C Working Group
- "Agreement" here = agreement between invoker and provider of the service, described at knowledge level
- *Overall: in 2001-2002 lots of proprietary jockeying and de-facto mode testing/pressuring of the open-consortial standards bodies (e.g., of W3C) "riding the tiger". Then more via W3C, Oasis starting in 2003.*

WS Players

- Basically, all the major software vendors
  - Biggies: Microsoft, IBM, Oracle, Sun, SAP, …
  - Webserver/XML ebiz space: BEA, CommerceOne, Ariba, …
  - Niche offerings, e.g., travel agent services, weather, …
- Standards bodies: W3C; Oasis incl. Security
- Overall: lots of proprietary jockeying and *de-facto* mode testing/pressuring of the open-consortial standards bodies (e.g., of W3C) "riding the tiger"
- Still low-level in terms of application abstractions

INSERT HERE:
a look at Amazon.com -> Sell Your Stuff -> Web Services: thousands of developers

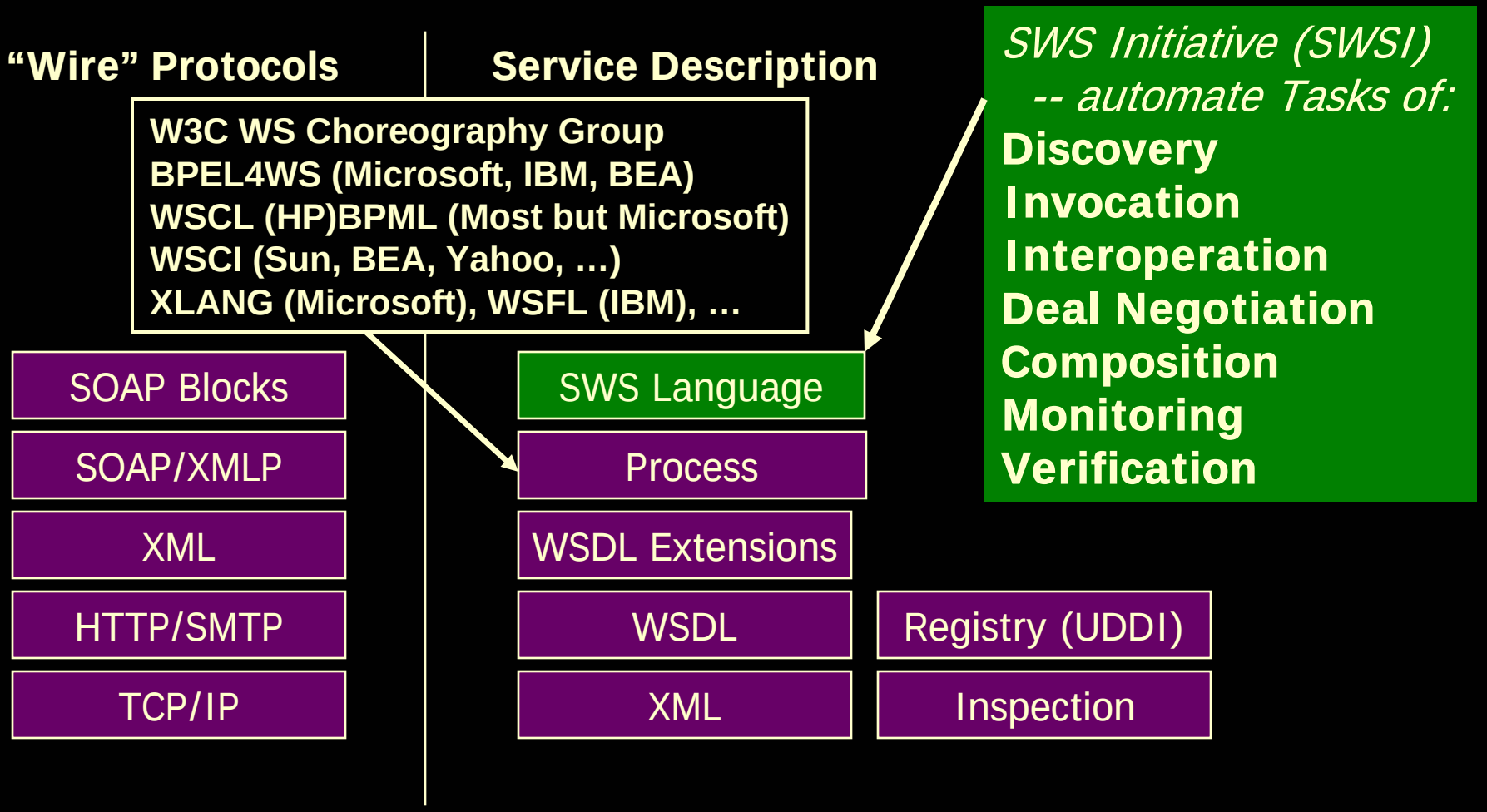
Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Semantic Web Services

- Convergence of Semantic Web and Web Services
- Consensus definition and conceptualization still forming
- Semantic (Web Services):
  - Knowledge-based service descriptions, deals
    - Discovery/search, invocation, negotiation, selection, composition, execution, monitoring, verification
    - Advantage: **reuse** of knowledge across app's, these tasks
  - Integrated knowledge
- (Semantic Web) Services: e.g., infrastructural
  - Knowledge/info/DB integration
  - Inferencing and translation

SWS Language effort, on top of Current WS Standards Stack



[Slide authors: Benjamin Grosof (MIT Sloan), Sheila McIlraith (Stanford) , David Martin (SRI International), James Snell (IBM)]

SWS: Research Players

- DAML Services (DAML-S)
  - service descriptions using ontologies and now rules
- Web Services Modeling Framework (WSMF)
  - EU, Oracle
  - early phase; list of many companies
- @ MIT: Sloan IT:
  - SweetDeal: e-contracting, policies
  - Extended COIN: financial info integration

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

B2B Tasks: Communication for Business Processes with Partners

- B2B business processes involving significant Communication with customers/suppliers/other-partners is overall a natural locus for future first impact of SWS.
- Customer Relationship Management (CRM)
  - sales leads and status
  - customer service info and support
- Supply Chain Management (SCM):
  - source selection
  - inventories and forecasts
  - problem resolution
  - transportation and shipping, distribution and logistics
- orders; payments, bill presentation

Some B2B Tasks (continued)

- bids, quotes, pricing, CONTRACTING; AUCTIONS; procurement
- authorization (vs. authentication) for credit or trust
- database-y: e.g.,
  - catalogs & their merging
  - policies
- inquiries and answers; live feedback
- notifications
- trails of biz processes and interactions
- ratings, 3rd party reviews, recommendations
- knowledge management with partners/mkt/society

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*

Some Answers to: "Why does SWS 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 structured knowledge.
  - Theme: *reuse* of knowledge across multiple tasks/app's/org's

Role of Standards

- Obs.: Standards are crucial, and central, to integration in an open era.
- → high percentage of effort invested in standards development in new generation web (XML, WS, SW, SWS)
- In SWS, this begins with basic research!

- Lots of strategy surrounding standards.
- Emerging standards efforts include much research.

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Research Agenda overall

- Develop core technologies and standards
  - *Knowledge representation* theory is critical foundation.
- Develop business applications, strategy
- Analyze requirements & opportunities wrt biz ↔ tech
- *Includes: concepts, theory, algorithms, design, prototyping, application scenarios, strategy, standards; evangelism*

- Benjamin Grosof's group:
  - Core rules, integration w/ ontologies, standards for that
  - End-to-end e-contracting; also finance, trust, biz policies
  - Business applications, implications, strategy more generally

New Research Application Scenarios for Rule-based Semantic Web Services

- SweetDeal [Grosof & Poon WWW-2003] configurable reusable e-contracts:
  - Represents modular modification of proposals, service provisions
    - LP rules as KR. E.g., prices, late delivery exception handling.
    - On top of DL ontologies about business processes from MIT Process Handbook
  - Evolved from EECOMS pilot on agent-based manufacturing SCM ($51M NIST ATP 1996-2000 IBM, Boeing, TRW, Vitria, others)
- Financial knowledge integration (ECOIN) [Firat, Madnick, & Grosof 2002]
  - Maps between contexts using LP rules, equational ontologies, SQL DB's.
- Business Policies:
  - Trust management (Delegation Logic) [Li, Grosof, & Feigenbaum 2003]: Extend LP KR to multi-agent delegation. Ex.: security authorization.

Knowledge Representation: What's the Game?

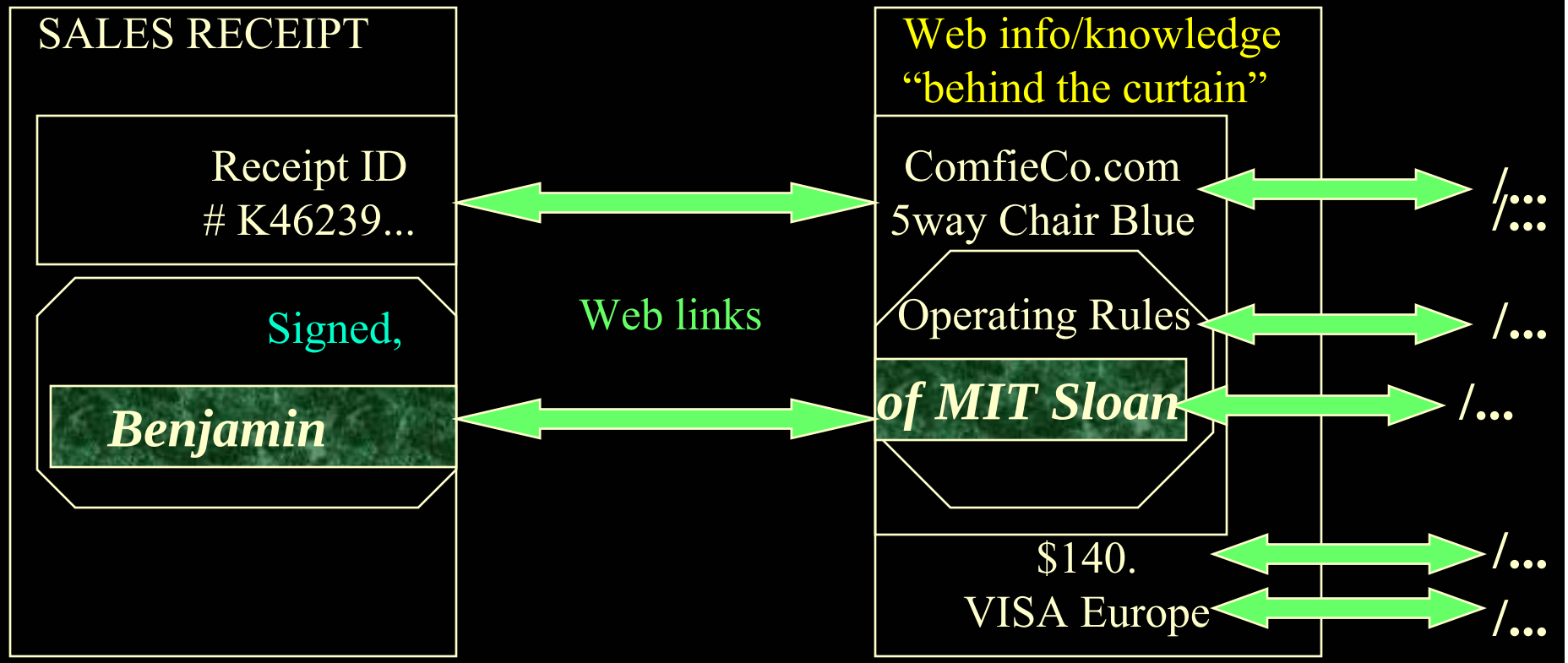
- Expressiveness: useful, natural, complex enough
- Reasoning algorithms
- Syntax: encoding data format, -- here, in XML
- Semantics: principles of sanctioned inference, independent of reasoning algorithms
- Computational Tractability (esp. worst-case): scale up in a manner qualitatively similar to relational databases: computation cycles go up as a polynomial function of input size

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Looks Simple To Start... then Gets Interestingly Precise

A Vision/Approach of what Web & Agents enable



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

SWS Tasks at higher layers of WS stack

Automation of:

- Web service discovery

  Find me a shipping service that will transport frozen vegetables from San Francisco to Tuktoyuktuk.

- Web service invocation

  Buy me "Harry Potter and the Philosopher's Stone" at www.amazon.com

- Web service deals, i.e., contracts, and their negotiation

  Propose a price with shipping details for used Dell laptops to Sue Smith.

- Web service selection, composition and interoperation

  Make the travel arrangements for my WWW11 conference.

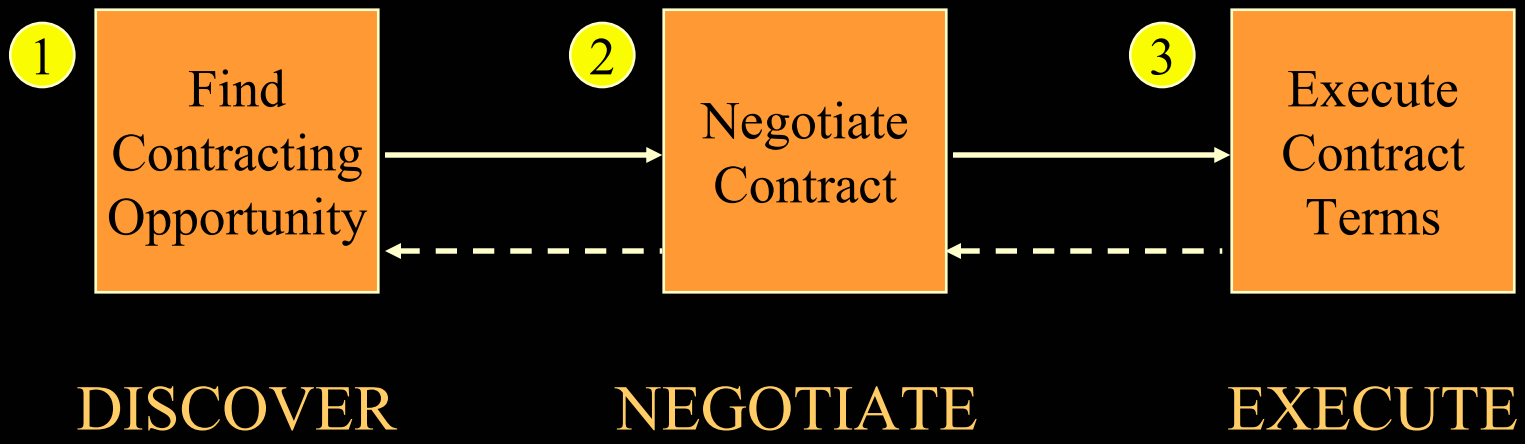
[Modification of slide also by Sheila McIlraith (Stanford) and David Martin (SRI International)]

SWS Tasks at higher layers of WS stack, continued

- Web service execution monitoring and problem resolution
  Has my book been shipped yet? … [NO!] Obtain recourse.
- Web service simulation and verification
  Suppose we had to cancel the order after 2 days?
- Web service executably specified at "knowledge level"
  The service is performed by running the contract ruleset through a rule engine.

[Modification of slide also by Sheila McIlraith (Stanford) and David Martin (SRI International)]

Contracting 1-2-3



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

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
  - ⇔ 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

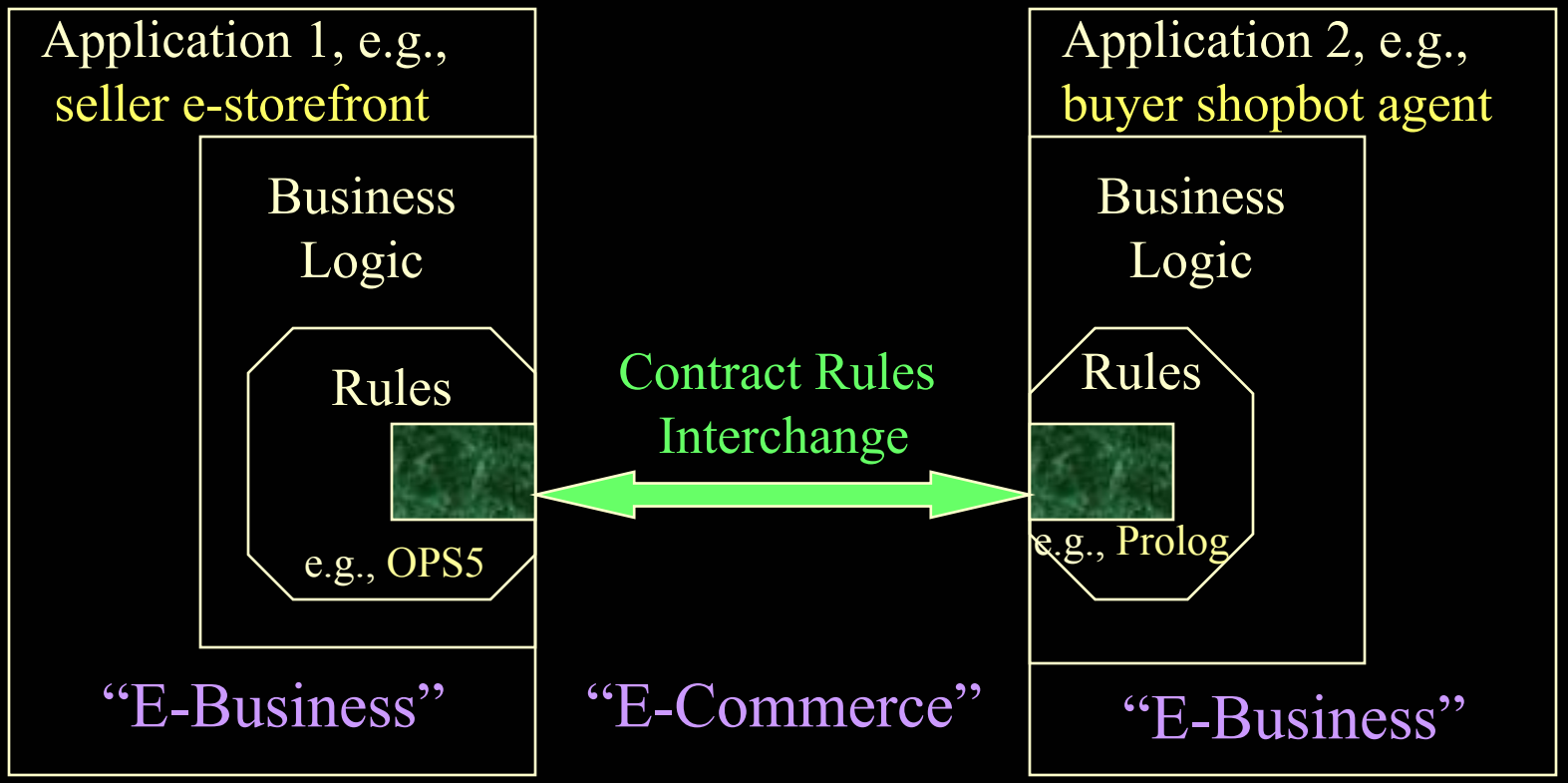
- Rules as way to specify (part of) business processes, policies, products: as (part of) contract terms.
- Complete or partial contract.
  - As default rules. Update, e.g., in negotiation.
- Rules provide high level of conceptual abstraction.
  - easier for non-programmers to understand, specify, dynamically modify & merge. E.g.,
  - by multiple authors, cross-enterprise, cross-application.
- Executable. Integrate with other rule-based business processes.

SweetDeal Approach

[Grosof , Labrou, & Chan EC-99; Wellman, Reeves, & Grosof Computational Intelligence 2002; Grosof & Poon Intl. J. of Electronic Commerce 2004]

- 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 semantic web rules
  - OWL 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

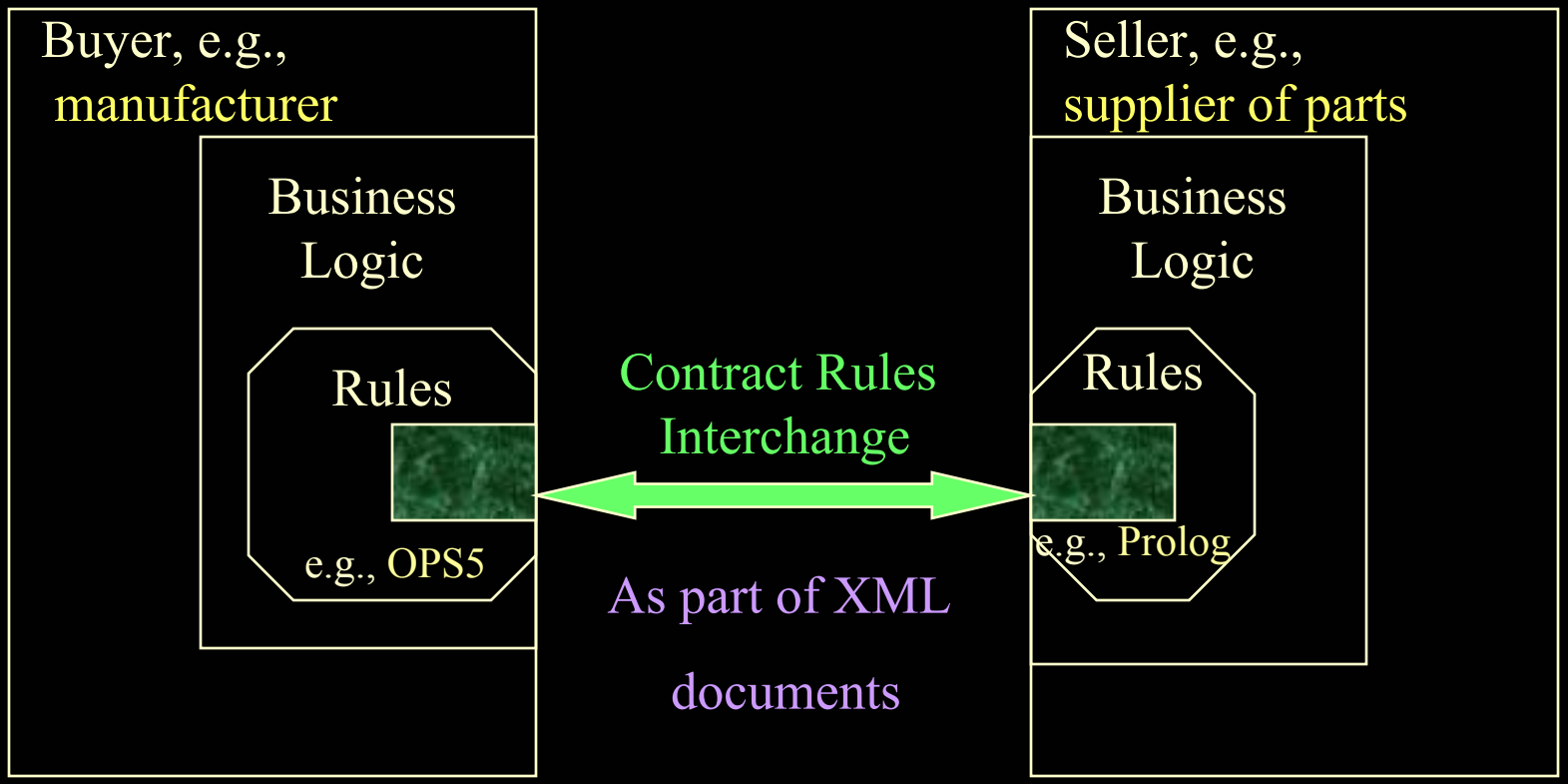


Contracting parties integrate e-businesses via shared rules.

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

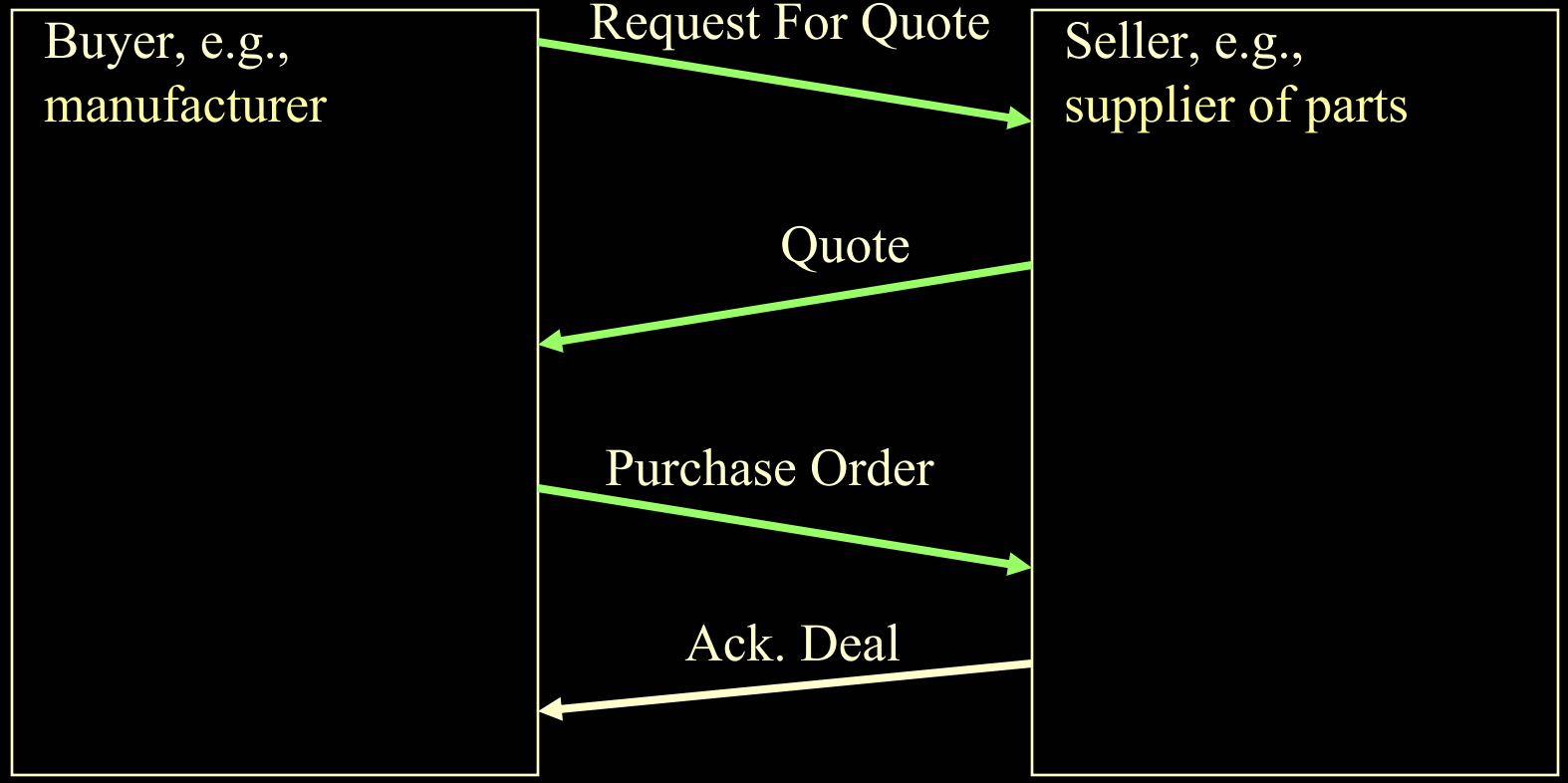
- Product descriptions
  - Product catalogs: properties, conditional on other properties.
- Pricing dependent upon: delivery-date, quantity, group memberships, umbrella contract provisions
- Terms & conditions: refund/cancellation timelines/deposits, lateness/quality penalties, ordering lead time, shipping, creditworthiness, biz-partner qualification, 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

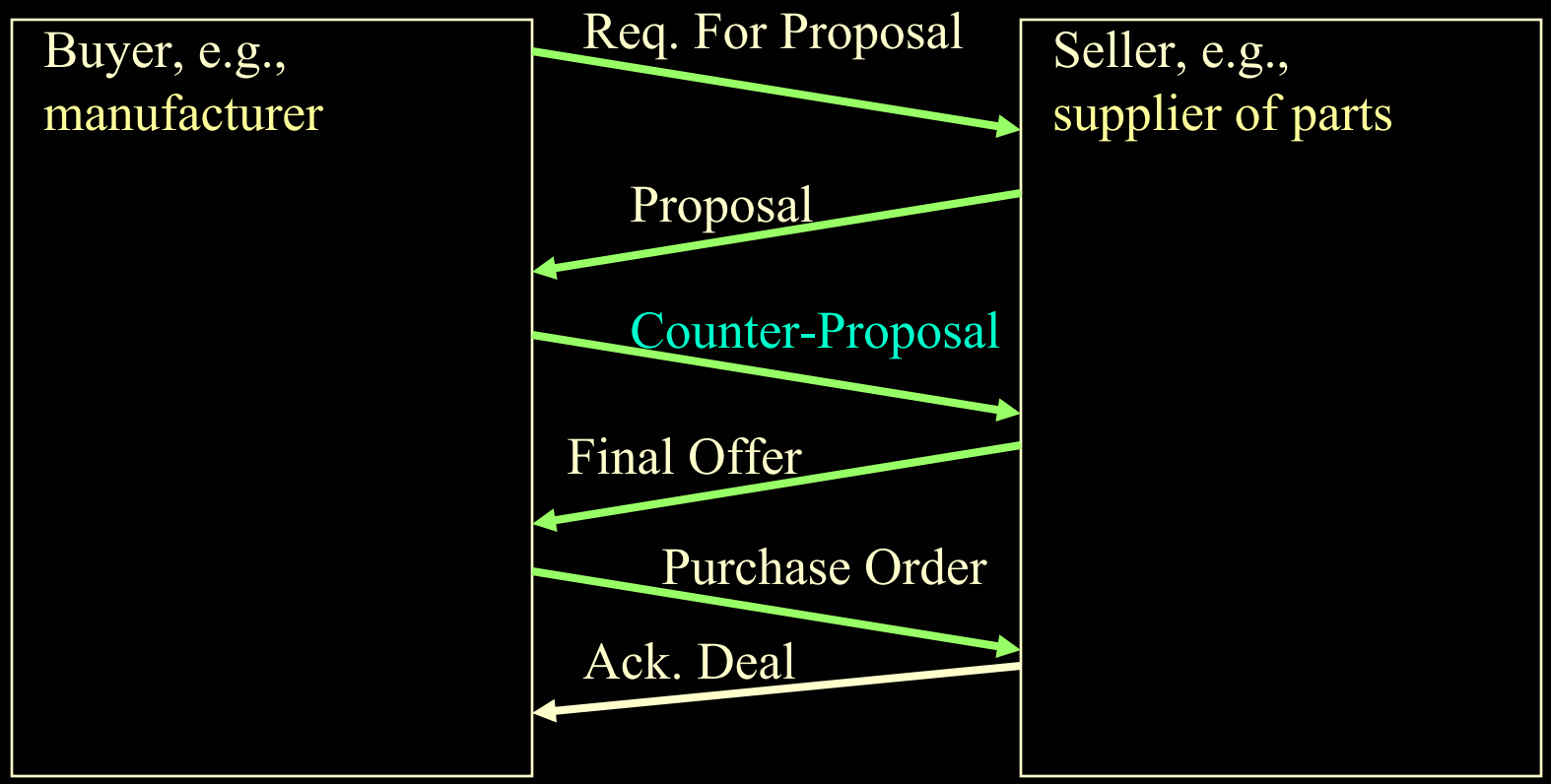


Contracting parties NEGOTIATE via shared rules.

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).

Courteous LP 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!

Negotiation Example --

XML Encoding of Rules in RuleML

```
<rulebase>
 <imp>
  <_rlab>usualPrice</_rlab>
  <_head>
   <cslit>
    <_opr><rel>price</rel></_opr>
    <ind>per_unit</ind>
    <var>PO</var>
    <ind>$60</ind>
   </cslit>
  </_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

EECOMS Example of Conflicting 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)
- ← preferredCustomerOf(?Buyer,?Seller) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- <leadTimeRule2> orderModificationNotice(?Order,30days)
- ← minorPart(?Buyer,?Seller,?Order) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- <leadTimeRule3> orderModificationNotice(?Order,2days)
- ← preferredCustomerOf(?Buyer,?Seller) ∧
- orderModificationType(?Order,reduce) ∧
- orderItemIsInBacklog(?Order) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- overrides(leadTimeRule3 , leadTimeRule1) .
- ⊥ ← orderModificationNotice(?Order,?X) ∧
- orderModificationNotice(?Order,?Y); GIVEN ?X ≠?Y.

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, ...*

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Flavors of Rules Commercially Most Important today in E-Business

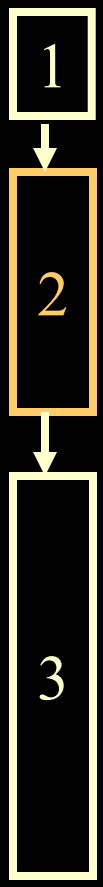
- E.g., in OO app's, DB's, workflows.

- 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.

Criteria for Contract Rule Representation



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

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*

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

Rule-based Semantic Web Services

- Rules/LP in appropriate combination with DL as KR, for RSWS
  - DL good for categorizing: a service overall, its inputs, its outputs
- Rules to describe service process models
  - rules good for representing:
    - preconditions and postconditions, their contingent relationships
    - contingent behavior/features of the service more generally,
      - e.g., exceptions/problems
  - familiarity and naturalness of rules to software/knowledge engineers
- Rules to specify deals about services: cf. e-contracting.

Rule-based Semantic Web Services

- Rules often good to executably specify 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
- Infrastructural: rule system functionality as services:
  - e.g., inferencing, translation

Analysis: High-Level Requirements for SWS

- Support Biz-Process Communication
  - E.g., B2B SCM, CRM
  - E.g., e-contracts, financial info, trust management.
- Support SWS Tasks above current WS layers:
  - Discovery/search, invocation, deal negotiation, selection, composition, execution, monitoring, verification

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

3 Areas of New Fundamental KR Theory
that enable Key Technical Requirements for SWS

- 1. Description Logic Programs:
  KR to combine LP (RuleML) rules on top of DL (OWL) ontologies, with:
  - Power in inferencing (including for consistency)
  - Scaleability of inferencing
- 2. Situated Logic Programs:
  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
- 3. Courteous Logic Programs:
  KR to combine rules from many sources, with:
  - Prioritized conflict handling to enable consistency, modularity; scaleably
  - Interoperable syntax and semantics

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

OWL: SW ontologies KR standard

- Recent Standard of W3C Web Ontologies Working Group, closely based on DAML+OIL precursor from research community. Uses RDF as syntax, extends RDF Schema.
- Based on Description Logic, a logical KR that has subset of expressiveness of first-order classical logic.
- Enables one to represent class hierarchies plus some more expressiveness, e.g., about cardinalities of properties and overlaps of classes.
- Still needs more theoretical and practical work to interoperate and bridge with conventional database schemas (e.g., Entity-Relationship (E-R) models and UML and SQL) and software engineering inheritance (e.g., class hierarchies in object-oriented (OO) languages such as Java and C++).
- Description Logic's commercial adoption, deployment, and application is much much less (yet) than Rules', and hugely less than OO/E-R/UML/SQL.

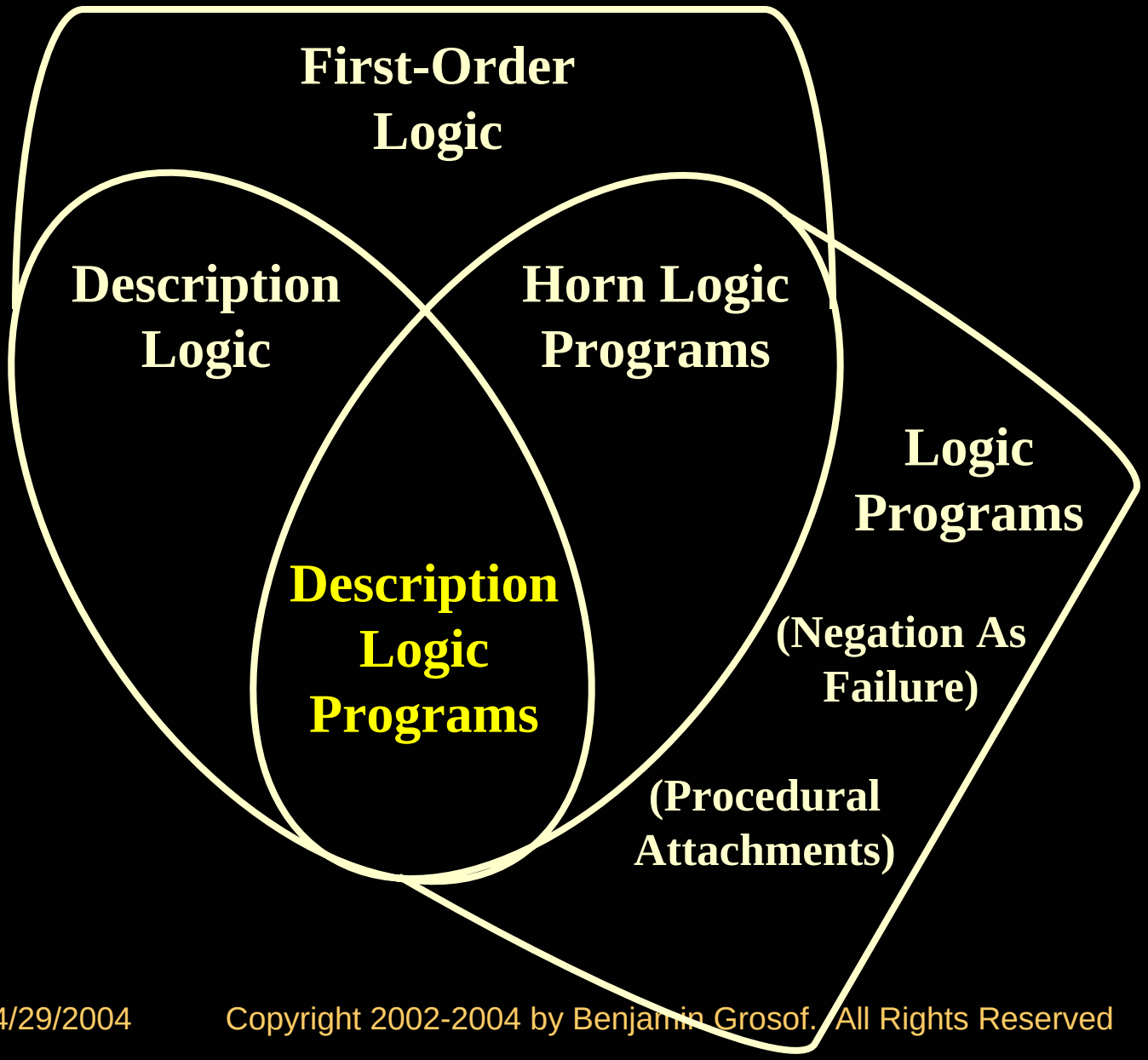
Simple Examples of the Mapping from DL to LP

- Simple: (are in RDF-Schema subset):
  - dog *is a subclass of* animal:
    - DL: dog $\subseteq$ animal $\Leftrightarrow$ LP: animal(?x) $\leftarrow$ dog(?x)
  - *Domain of* hasBitten *is* animal:
    - DL: Top $\subseteq$ hasBitten.animal
    - $\Leftrightarrow$ LP: animal(?x) $\leftarrow$ hasBitten(?x,?y)

More Complex Example of the Mapping from DL to LP

- More complex: (beyond RDF-Schema subset):
  - DL: $(\text{pet} \cap ((\text{dog} \cap \exists \text{hasBitten.person}) \cup (\text{feline} \cap \text{large})))$
    $\subseteq ((\text{dangerous} \cap \text{animal}) \cap (\forall \text{keeper.careful}))$
  - $\Leftrightarrow$ LP: $\text{dangerous}(?x) \wedge \text{animal}(?x)$
    $\leftarrow \text{pet}(?x) \wedge$
    $(\quad (\text{dog}(?x) \wedge \text{hasBitten}(?x,?y) \wedge \text{person}(?y))$
    $\vee (\text{feline}(?x) \wedge \text{large}(?x)) \quad) ;$
  - $\text{careful}(?z)$
    $\leftarrow \text{pet}(?x) \wedge \text{keeper}(?x,?z) \wedge$
    $(\quad (\text{dog}(?x) \wedge \text{hasBitten}(?x,?y) \wedge \text{person}(?y))$
    $\vee (\text{feline}(?x) \wedge \text{large}(?x)) \quad)$

Venn Diagram: Expressive Overlaps among KR's



Technical Capabilities Enabled by DLP

- LP rules "on top of" DL ontologies.
  - E.g., LP imports DLP ontologies, with completeness & consistency
  - Consistency via completeness and use of Courteous LP
- Translation of LP rules to/from DL ontologies.
  - E.g., develop ontologies in LP (or rules in DL)
- Use of efficient LP rule/DBMS engines for DL fragment.
  - E.g., run larger-scale ontologies
  - $\Rightarrow$ Exploit: Scaleability of LP/DB engines >> DL engines , as |instances| $\uparrow$.

- Translation of LP conclusions to DL.
- Translation of DL conclusions to LP.

- Facilitate rule-based mapping between ontologies / "contexts"

Sequence Outline of Workshop

- Introduction: motivations; *acquainting; identify questions*
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- Drill-down on Rules: requirements; uses; maturity; rule-based SWS
- Analysis of Technical Requirements for SWS
- Overview of New Fundamental KR Theory for SWS
  - Drill-downs on: Description Logic Programs; RuleML - Skim
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- SWS Research Directions
- SWS Adoption Roadmap; Market Evolution
- *Discussion:* Application & Entrepreneurial Opportunities

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
  - Workshop Series established on Rules, annually at International Semantic Web Conference
  - Has now a "home" institutionally in DAML and Joint Committee
    - Discussions well underway to launch W3C, Oasis efforts
- 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

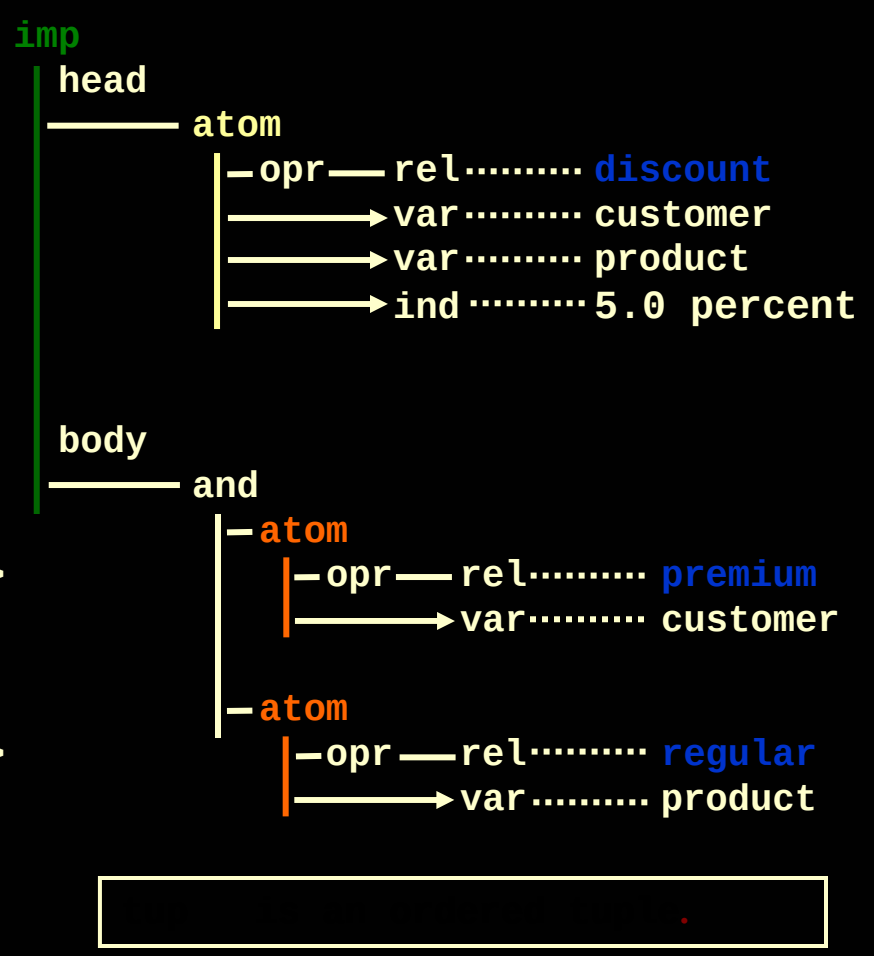
RuleML Example: Markup and Tree

"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);

Markup Syntax

```
<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>
```



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
    - $\vee,\forall,\exists$ in body; $\wedge,\forall$ 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

Tools: SweetRules, including SweetJess

- SweetRules V1 '01: RuleML inferencing and bi-directional translation with equivalent semantics via RuleML, between:
  - XSB Prolog: backward Ordinary Logic Programs (OLP)
  - Smodels: forward OLP
  - IBM CommonRules: forward Situated Courteous LP (SCLP)
  - Knowledge Interchange Format (KIF): First Order Logic interlingua
  - *+ Design in principle for:* SQL
    - well-understood in theory literature: as OLP
  - *+ Design in principle for:* production (OPS5), ECA
    - Based on Situated extension of LP, piloted in IBM Agent Building Environment '96 for info-workflow applications. Also piloted in EECOMS.
    - BUT: not much other literature/theory to support
    - HENCE motivation to "bring them to the party" … resulting in:
- …V2 '02: adds SweetJess as component:
  - Jess: production (OPS5) , close to ECA
    - popular, open-source, Java: it's useful in particular
    - expressive restriction: "all bound sensors"

SWEET = Semantic WEb Enabling Tools

Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of Research Agenda; incl. SWS application scenarios
- End-to-end E-Contracting as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New Fundamental KR Theory, incl. Description LP
- Financial Info & Reporting: ECOIN ontologies/contexts integration
- Trust Management Policies: rules & delegation in authorization
- SWS Research Directions
- SWS E-C Adoption Roadmap; Market Evolution

INSERT HERE:
a look at www.XBRL.org

XBRL = eXtensible Business Reporting Language

a major industry standards effort for business reporting in XML