Commercializing Semantic Web: Rules, Services, and Roadmapping

Prof. Benjamin Grosof

MIT Sloan School of Management

Information Technologies group

http://ebusiness.mit.edu/bgrosof

Invited Keynote Presentation (1-hour) at the

1st European Semantic Technology Conference (ESTC-2007)

Vienna, Austria, June 1, 2007

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, Stanford;   BA Applied Math Econ/Mgmt, Harvard
- Semantic technology as main research area:
  - Rules as core technology; on web; in combination with ontologies
  - Business Applications, Implications, Strategy, Market Evolution
- Overall: knowledge representation, e-commerce, agents

- ***News:*** Will join Vulcan Inc. in July 2007
  - leading a new research program; and working closely with the VC arm.

Outline

- Introduction
- Semantic Rules technology: commercial scene, recent developments
  - research breakthroughs → initial steps of commercial adoption
  - relationship to ontology
- Roles for rules in services; roadmapping
  - conceptual; lifecycle tasks; policies
  - application areas; examples; business value; drivers
- Roadmapping rule technology
  - research directions
  - disruption pattern in commercial business rules market
- Conclusions

Next Generation Web

Semantic Web Services

Semantic Web techniques

Web Services techniques

Automated Knowledge Bases

Rules (RuleML)

Ontologies (OWL)

Databases (SQL, XQuery, RDF)

API's on Web

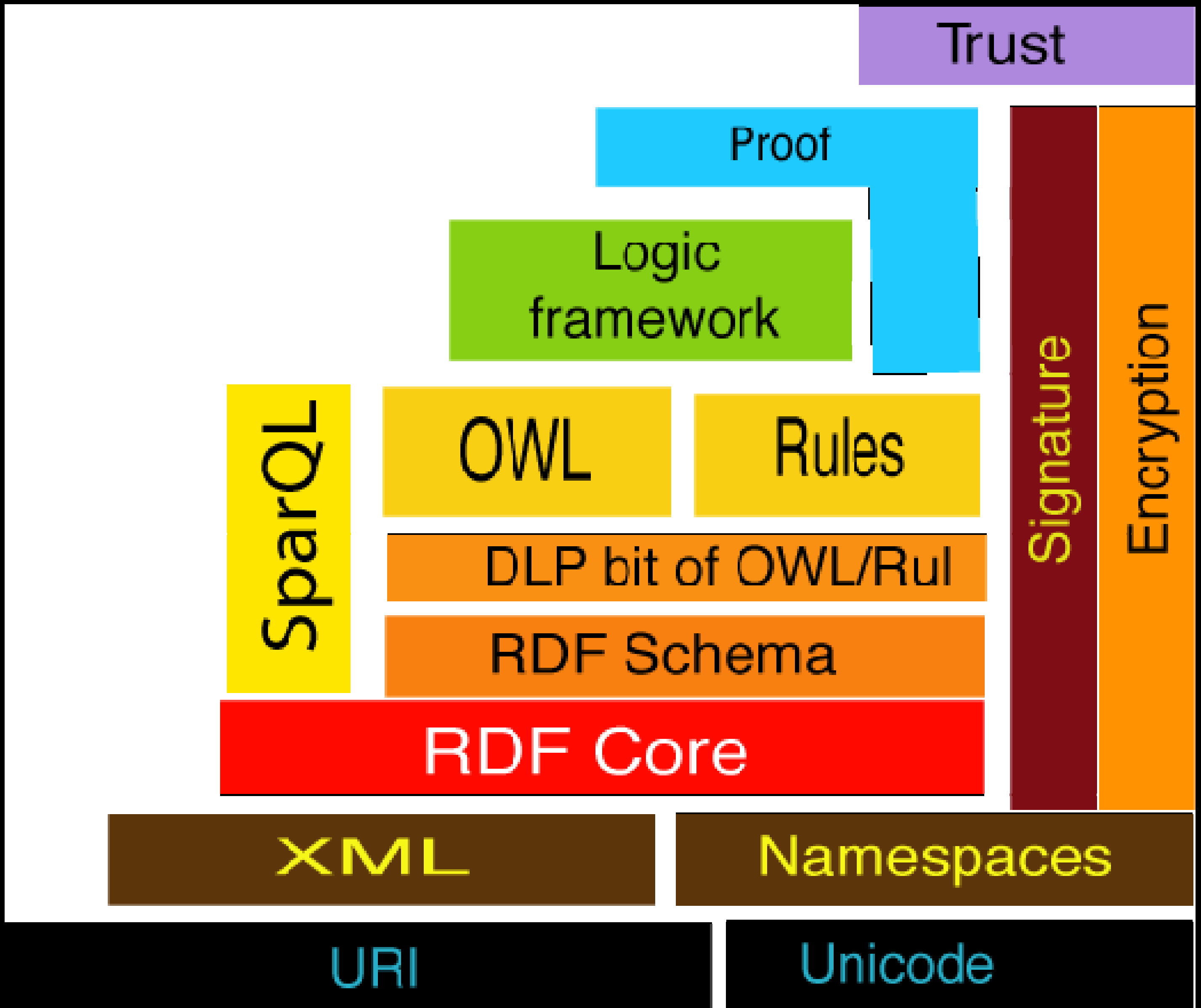
(WSDL, SOAP)

XML

First Generation Web

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

2005 W3C Semantic Web "Stack": Standardization Steps



Semantic Web Services

- Convergence of Semantic Web and Web Services
- Consensus definition and conceptualization still forming
- Semantic (Web Services)
  - Knowledge-based service descriptions, deals
    - Support many lifecycle tasks: discovery/search, invocation, negotiation, selection, composition, execution, monitoring, verification. Also: change management, provenance.
    - Advantage: **reuse** of knowledge across app's, these tasks
    - Semantic SOA is closely related
- (Semantic Web) Services: e.g., infrastructural
  - Knowledge/info/DB integration
  - Inferencing and translation

Our Semantic Technology Research Areas

- Rules as core technology; on web; in combination with ontologies
  - Co-Founder, RuleML (Rule Markup Language Initiative) standards design
  - Invented several techniques being widely adopted commercially, e.g.:
    - interoperable business rules in XML: declarative logic programs (LP) interchange
    - LP rules represent subset of Description Logic (DLP); translate OWL2LP
    - translate LP ↔ production rules (PR); LP2Jess; strong semantics for PR
  - W3C Rule Interchange Format (RIF, in draft) is based largely on the above

- Business Applications, Implications, Strategy, Market Evolution:
  - E-services lifecycle incl. e-contracting
  - Policies; trust; shopping & ads; financial; info integration with ontological context mediation; business process exception handling; …
  - Co-Editor, SWSF (Semantic Web Services Framework, 2005)
  - Adoption prospects for many industry verticals/tasks/domains

Outline

- Introduction
- Semantic Rules technology: commercial scene, recent developments
  - research breakthroughs → initial steps of commercial adoption
  - relationship to ontology
- Roles for rules in services; roadmapping
  - conceptual; lifecycle tasks; policies
  - application areas; examples; business value; drivers
- Roadmapping rule technology
  - research directions
  - disruption pattern in commercial business rules market
- Conclusions

Flavors/Families of Rules Commercially Most Important today in E-Business

- E.g., in OO app's, DB's, workflows.
- "CCI" = Currently Commercially (most) Important

1. Relational databases (RDBMS), SQL: Views, queries, facts are all rules.
   - XQuery, SPARQL emerging. SQL99 even has recursive rules.
2. Production rules (OPS5 and CLIPS heritage): e.g.,
   - Fair Isaac, ILOG, Haley, Jess, etc.
3. Event-Condition-Action rules (loose family similar to PR), cf.:
   - business process automation / workflow tools.
   - active databases; publish-subscribe
4. Prolog. *"logic programs" as a full programming language.*
   - *"Pure" Prolog – declarative LP subset, has no cuts or external procedure calls, does backward inferencing in declarative LP*
5. *(Lesser: other knowledge-based systems, and things hard to classify or further from declarative such as some "business rule" systems.)*

Production Rules (PR): History

- Grand-daddy: OPS5 research system at CMU in '70's
- <u>NOT declarative</u>
- CLIPS *system*: open PR system done by US govt ~ a decade ago
- CLIPS *syntax*: used with tweaks by many current PR systems
- Have incremental/dynamic/updating capabilities
- PR reengineered in '90's to be fine-grain embedded in C++/Java etc. programming language, with access to those external objects
- OMG PRR standards effort since late 2003
- *Full PR systems often also have scripting and a kind of backward-inferencing capability; for semantically interoperable web rules that has not been the focus (at least initially).*

Event-Condition-Action Rules (ECA): History

- Loose family; no consensus/standard detailed formulation/abstraction
- Fairly similar to Production Rules: forward, Conditions, Actions
  - NOT declarative
- Plus there's the "Event" notion (see next slide)
- More focus than PR on incremental inferencing and specialized optimizations around "Events"

- Many database systems have ECA capabilites, e.g., for transactional triggers or pub-sub.
- In '90's became used widely for loose coupled business process automation / workflow / integration / orchestration

Events in ECA Rules

- "Event" is a kind of premise info update, <u>and</u> a kind of control trigger for incremental inferencing
  - This conflates declarative and procedural aspects! *Challenge ;-)*
  - Often not precisely described, for given ECA system/language
- "Event" part of a rule body is a kind of condition, and control "port"
- Often there's "complex event processing" with specialized treatment
  - E.g., event sublanguage and special processors for generating/testing events
- *History* of event updates/info-states is often important

- Other flavors of rules can also do events and incremental inferencing, to varying degrees

Semantic Rules News I -- Research

Recent developments in semantic rules (last decade) :

- Fundamental theory and technique breakthroughs, e.g.:
  - Declarative logic programs (LP) basis for interoperability, then webized → RuleML standards design (2001-)
  - Description LP ontology integration, represent substantial OWL subset as rules (RDFS++, taxonomies, …); highly scalable
  - Production LP interoperability+semantics for production rules, declarative procedural attachments for actions and queries, correct default negation in PR
  - Courteous LP prioritized defaults, robust modular merging
  - … All while maintaining scalability similar to RDBMS (poly-time)
  - SweetRules V2 open source toolset platform (2004-)
    - Proof Of Concept for all the above
    - Interoperate and expressively extend: Jess production rules, XSB Prolog, IBM Common Rules, HP Jena, OWL-DLP, …
- Large US, EU research projects (DAML, WSMO) focus on rules (DARPA Agent Markup Language; Web Service Mediation Ontology)

Semantic Rules News II - Industry

- Semantic Web Services Framework design (2005) focuses on rules
- RuleML standards design gets large mindshare for its technical approach
- W3C forms Rule Interchange Format (RIF) Working Group, full standards effort, after holding a Workshop (Dec. 2005)
  - Based on RuleML design
  - Strong participation
  - *Breakthroughs (prev. slide) underpin agenda/optimism/energy in W3C RIF -- notably that production rule vendors join the SW*
- OMG forms standards efforts on production rules, rule management
  - Delegates semantics aspect to RuleML, W3C RIF efforts
- 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, RuleML-2006, RR-2007) colocated with ISWC / ESWC
- Gartner, Forrester, etc. regularly issue market analyst reports on rules
  - Strong sustained growth in market; attracts new entrants

Semantic Rules News III – Industry Adoption

- Commercial adoption of semantic rules technology *accelerates* (2005-)

  Vendors:

  – IBM CommonRules (1999) *AlphaWorks product*

  – HP Jena-2 (2004) *Open source*

  – SweetRules V2 (2004) *Open source platform, POC (multi-institutional)*

  – Ontoprise, OntoText, BBN Technologies, MITRE, VIS, Top Quadrant, several other startups (by 2006) *core proprietary*

  – Oracle main DBMS product (2007) *core proprietary*

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

- Get the KR right (knowledge representation)
  - More mature research understanding
  - Semantics independent of algorithm/implementation
  - Cleaner; avoid general programming/scripting language capabilities
  - Highly scaleable performance; better algorithms; choice from interoperability
  - Highly modular wrt updating; use prioritization
  - → Highly dynamic, scaleable rulebase authoring: distributed, integration, partnering
- Leverage Web, esp. XML
  - Interoperable syntax
  - Merge knowledge bases
- Embeddable
  - Into mainstream software development environments (Java, C++, C#); not its own programming language/system (cf. Prolog)
- Knowledge Sharing: intra- or inter- enterprise
- Broader set of Applications

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 Default Negation [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:

- Description LP: combine Description Logic ontologies [Grosof, Horrocks, *et al.*]
  - + Expressive extensions: [ter Horst]; [Motik DL-Safe]; WSML-Core; SWRL (Datalog Horn LP); …
- Courteous LP / Defeasible Logic: prioritized conflict handling [Grosof; Antoniou, Billington, *et al.*]
  - Robust, tractable, modular merging & updating
- Production Rules as LP: interoperate [Grosof, Gandhe, Ganjugunte, *et al.*]
  - Declarative LP as interoperable core between commercial families [Grosof *et al.*]
- Situated LP: hook rules up to services [Grosof]
- Default Inheritance in LP: combine OO default ontologies [Grosof & Bernstein; Yang & Kifer]
- Hypermonotonic Reasoning: combine with FOL [Grosof & Martin (in-progress)]
  - Also: Answer Set Semantics and Autoepistemic Logic [Gelfond, Eiter, Polleres, *et al.*]

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:

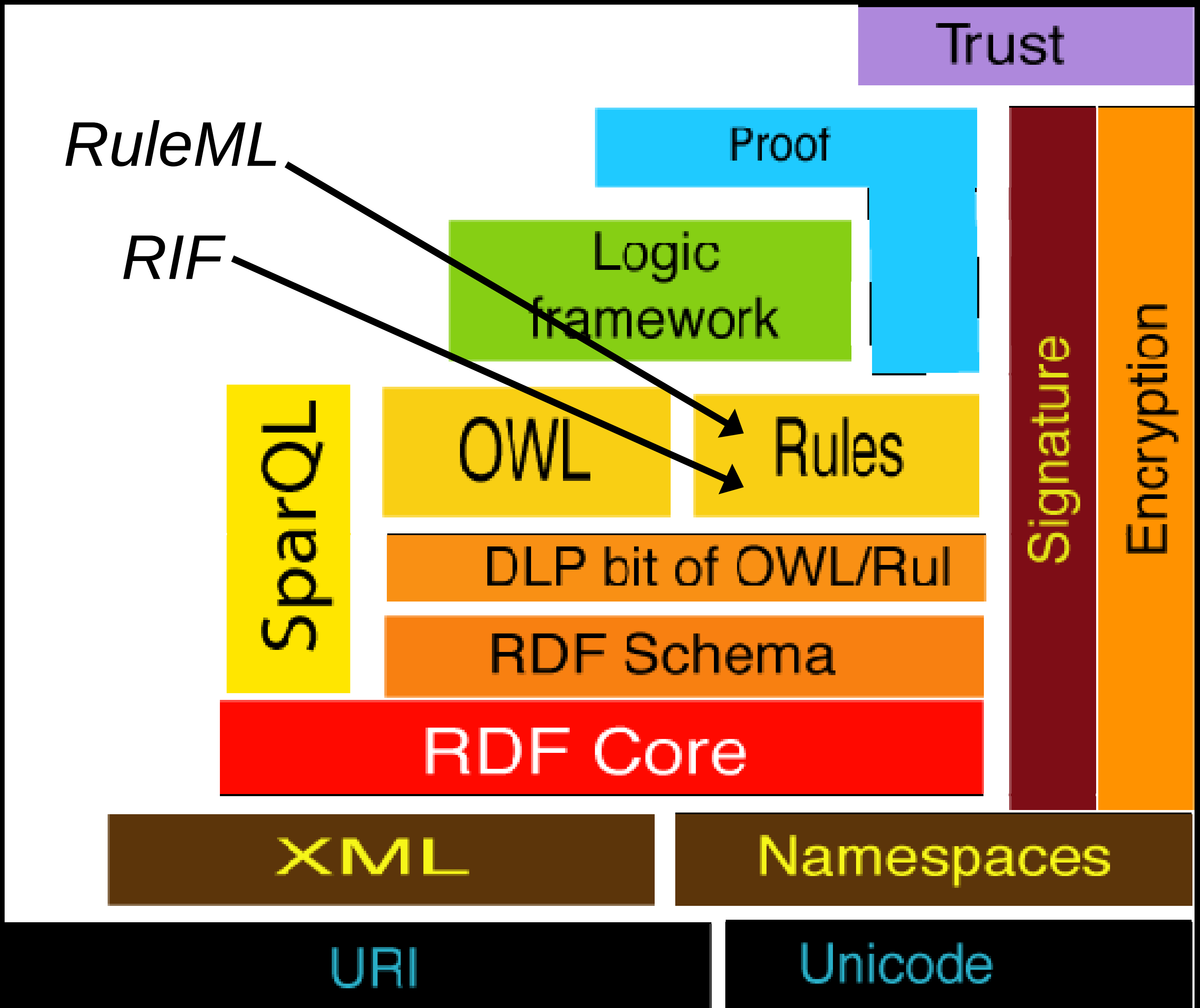
Suppose one adds the rule:

– @emailCouponAd(?cust, RamadaHotel, ?x)
  ← discount(?cust, RamadaHotel, ?x).

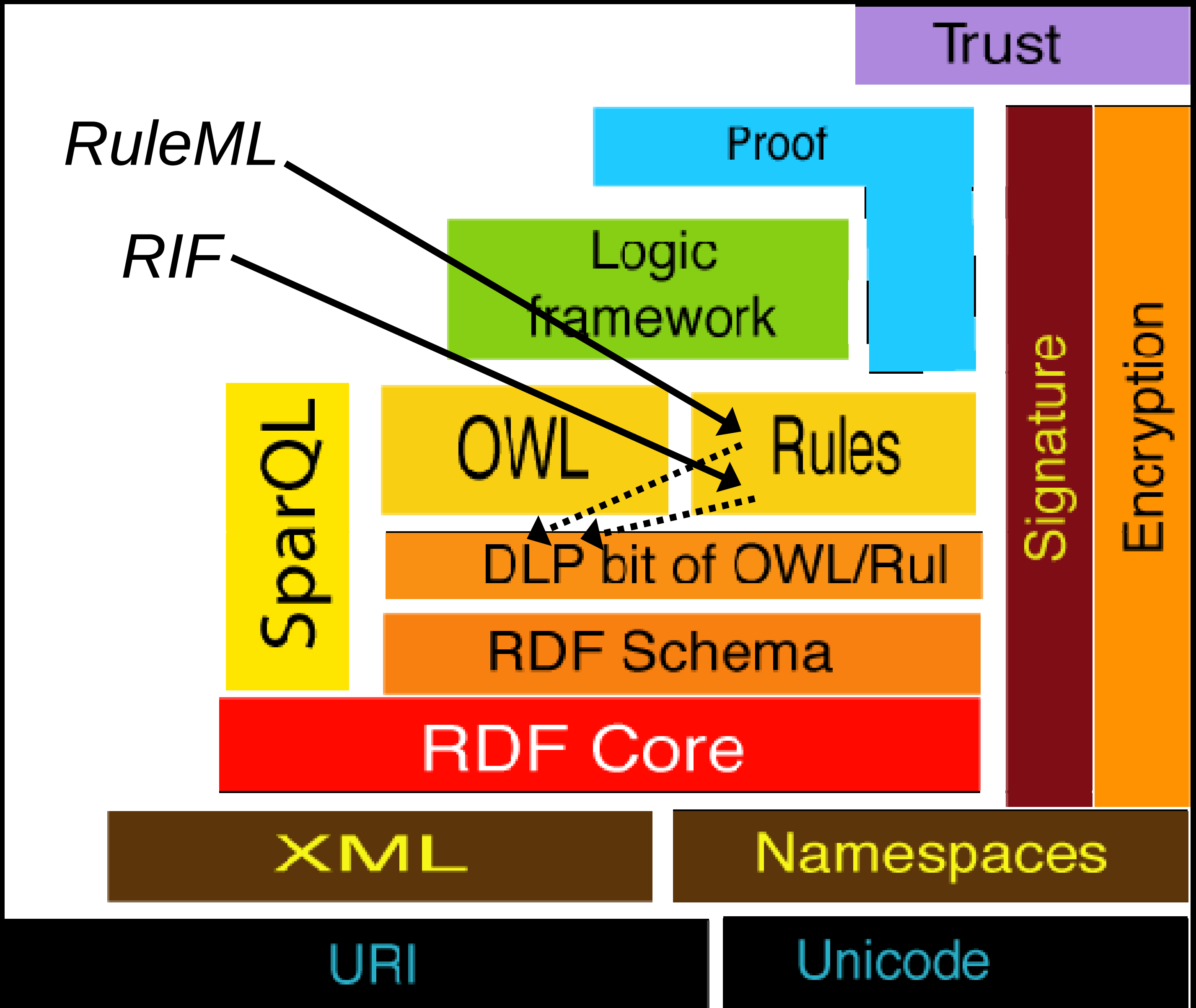
Then P entails the action (i.e., sanctions a call to an attached procedure):

@emailCouponAd(Faisal, RamadaHotel, 25percent).

2005 W3C Semantic Web "Stack": Standardization Steps



2005 W3C Semantic Web "Stack": Standardization Steps



SWRL as RuleML

- Essentially, SWRL is RuleML!!
  - What SWRL adds to OWL is just restricted RuleML rules*
  - *(Many people are not aware of this.)*
- More precisely:
  - SWRL = SWRL-rules* + OWL-DL.
  - SWRL-rules is restricted RuleML.
    - It's a member of the overall RuleML expressive family (lattice), syntactically and semantically.
      - subset of Datalog Horn with max predicate arity 2
      - … which is relatively basic and quite limited as compared to the full expressiveness of RuleML rules

  - * under the named-classes-only restriction, which is expressively inessential and typically is employed in practice, e.g., by virtually all SWRL tools today

"Ontology" More General than OWL

- "ontology" in general sense = definitional knowledge [sense from AI and philosophy]
  - Could be in any KR, e.g., FOL, LP, or probabilistic

- Important kinds of ontologies:
  - Taxonomies: vocabulary and basic class hierarchy
  - Description Logic; RDF-S
  - Object oriented with default inheritance, e.g., C++/Java/C# class-hierarchy frameworks with overriding or cancellation of inheritance
  - Database schemas; E-R
  - XML schemas
  - UML – some aspects
  - Axiomatizations in FOL, e.g., of processes, space, time

Outline

- Introduction

- Semantic Rules technology: commercial scene, recent developments
  - research breakthroughs → initial steps of commercial adoption
  - relationship to ontology

- Roles for rules in services; roadmapping
  - conceptual; lifecycle tasks; policies
  - application areas; examples; business value; drivers

- Roadmapping rule technology
  - research directions
  - disruption pattern in commercial business rules market

- Conclusions

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

EEECOMS 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)) ← (?X ≠?Y) .

Friday, October 15, 2004 MEMBERS LOG-IN | SEARC

XBRL®
eXtensible Business Reporting Language

PRESS ROOM EVENTS CONTACT

- 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

Welcome to XBRL International

16 - 17 NOVEMBER 2004 BRUSSELS
10th International 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 or a family of "XML" languages which is becoming a standard means communicating information between businesses and on the intern

XBRL is being developed by an international non-profit consortiur approximately 250 major companies, organisations and governm 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 benefits XBRL. It explains how individuals and companies can join the effo move forward and make use of the language.

PRESS ROOM EVENTS CONTACT US JUR

Equational Ontological Conflicts

Key Concepts

Gross Profit = Net Sales – Cost of Goods

Price = Nominal Price + Shipping

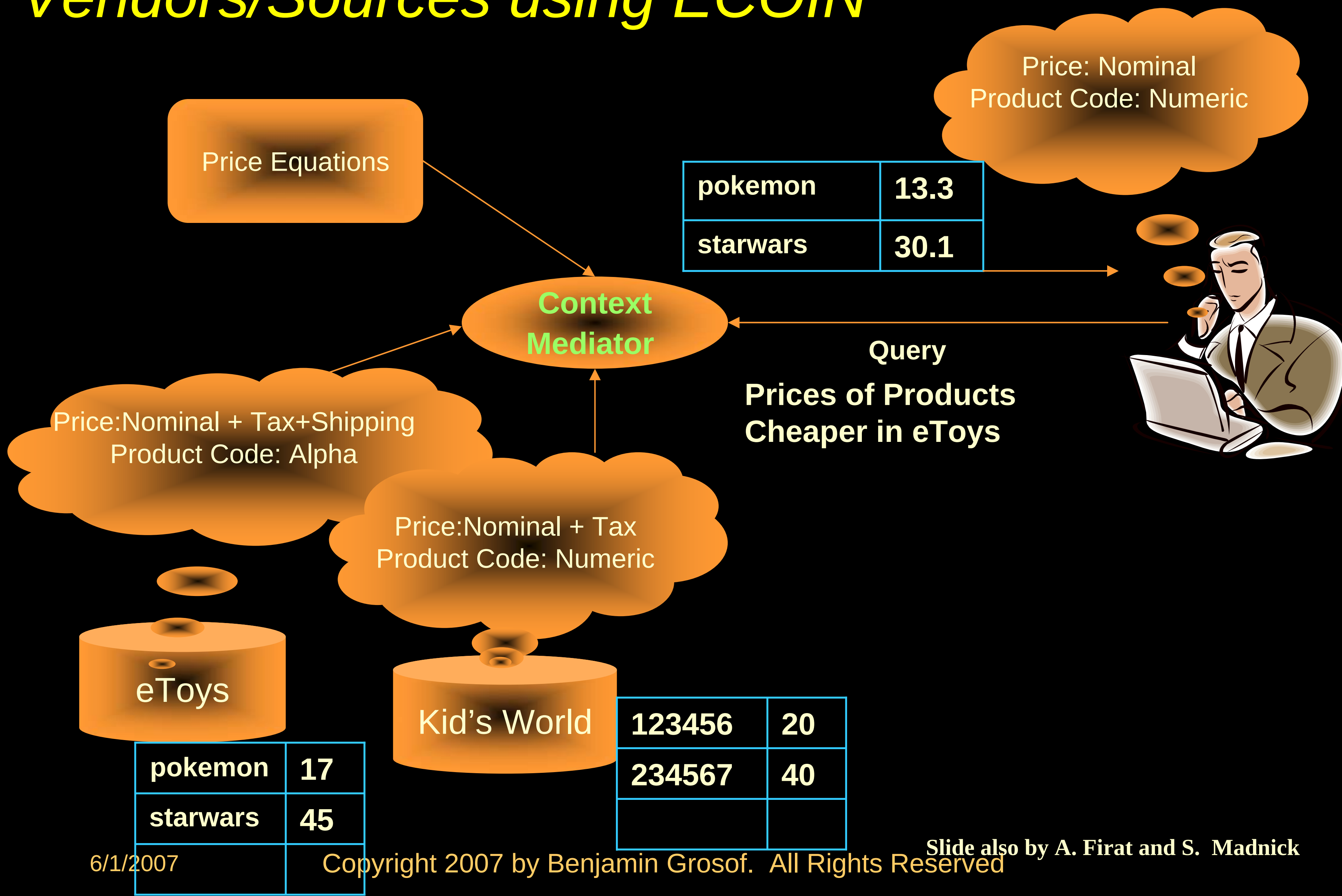
Gross Profit = Net Sales – Cost of Goods – Depreciation

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



Slide also by A. Firat and S. Madnick

Approach: ECOIN

- <u>E</u>xtended <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

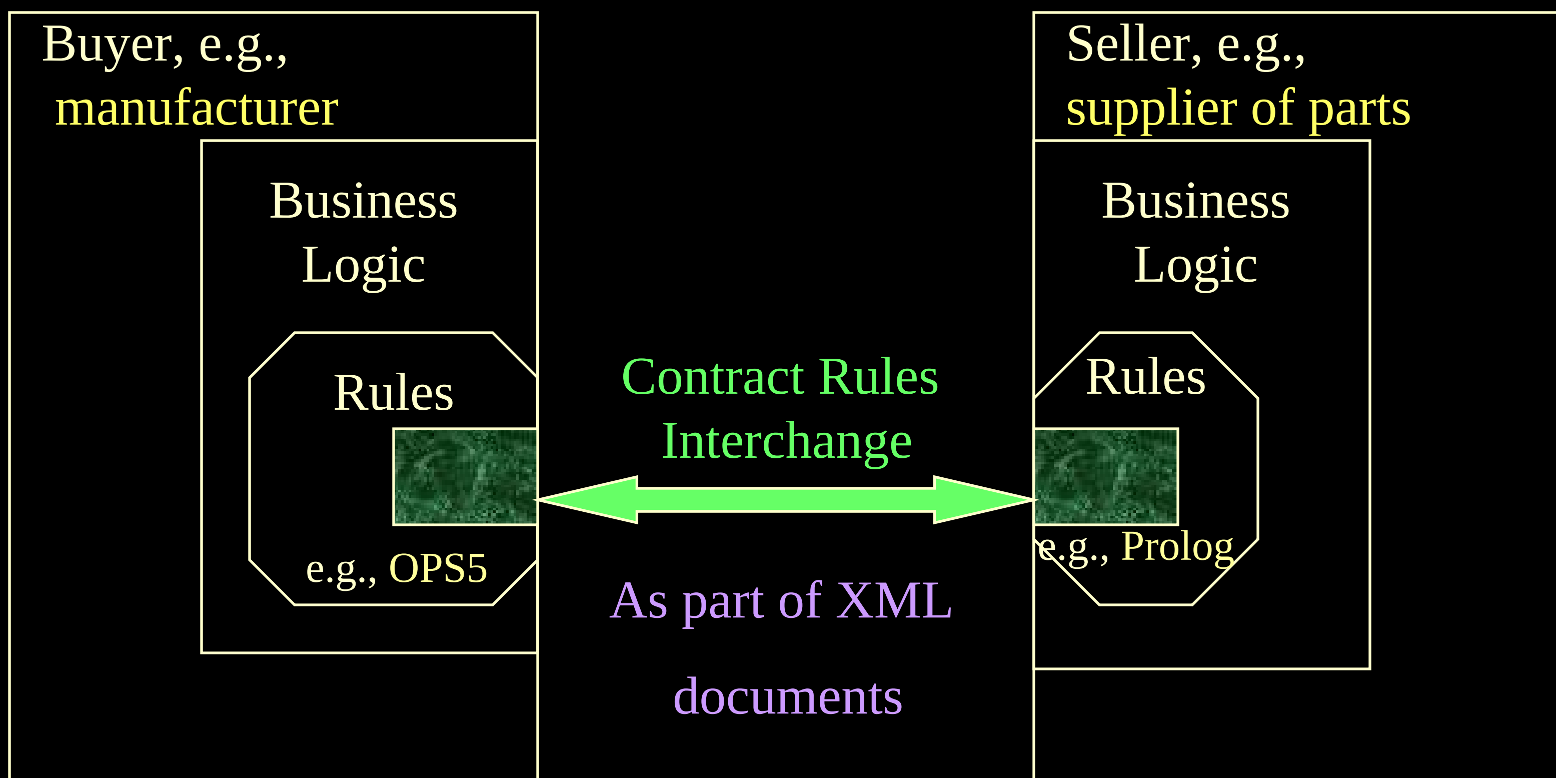
End-to-End E-Contracting Tasks

- 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

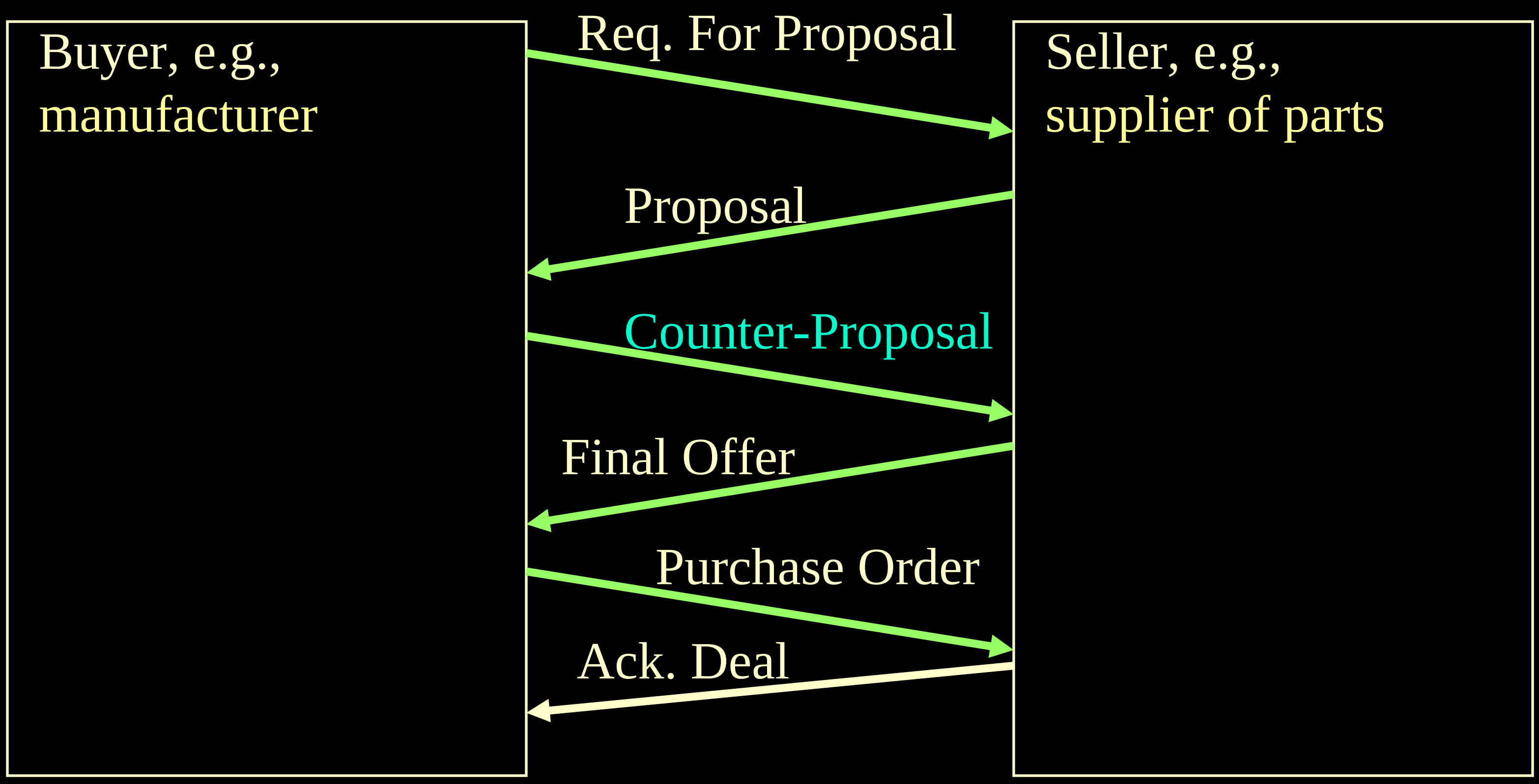


Contracting parties NEGOTIATE via shared rules.

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, service 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



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>
- <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
  - ⇔ heterogeneous rule/DB systems / rule-based applications (“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)*

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.

SWS and Rules   Summary

*** SWS Tasks Form 2 Distinct Clusters,*

each with associated Central Kind of Service-description Knowledge and Main KR

1. Security/Trust, Monitoring, Contracts, Advertising/Discovery, Ontology-mapping Mediation
   - Central Kind of Knowledge: Policies
   - Main KR: Nonmon LP (rules + ontologies)
2. Composition, Verification, Enactment
   - Central Kind of Knowledge: Process Models
   - Main KR: FOL (axioms + ontologies)
     - + Nonmon LP 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 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

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

Policies Via Rules

- Many, if not most, policy languages are rule-based.
- Rules (+ ontologies) are essentially sufficient for many policy languages / tasks / applications.
- Example: XACML (eXtensible Access Control Markup Language – an Oasis standard)

Concept of Authorization

- Authorization of Access to Information
  - E.g., to read or disclose (pull or push)
  - E.g., to write or change
- Authorization of Transactions
  - E.g., where resources are committed
  - E.g., purchase or sale or payment
- Can be viewed as:
  - Permission vs. prohibition (vs. neither)

Important Authorization Tasks

- Policies … and their
- Enforcement / execution …

} = **"trust management"**

- for …

- Security
- Confidentiality
- Privacy
- Access Control
- Trust

- Wrt …
- Information … and
- Transactions

Policies for Compliance and Trust Mgmt.: Role for Semantic Web Rules

- Trust Policies usually well represented as rules
  - Enforcement of policies via rule inferencing engine
  - E.g., Role-based Access Control
    - This is the most frequent kind of trust policy in practical deployment today.
  - W3C P3P privacy standard, Oasis XACML XML access control emerging standard, …
- Ditto for Many Business Policies beyond trust arena, too
  - "Gray" areas about whether a policy is about trust vs. not: compliance, regulation, risk management, contracts, governance, pricing, CRM, SCM, etc.
  - Often, authorization/trust policy is really a part of overall contract or business policy, at application-level. Unlike authentication.
  - Valuable to reuse policy infrastructure

Example of Rules: Accessing Medical Records

[N. Li, B. Grosof, J. Feigenbaum ACM TISSEC 2003]

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

Slide also by Ninghui Li and Joan Feigenbaum

Trust Policies and Compliance in US Financial Industry Today

- Ubiquitous high-stakes Regulatory Compliance requirements
  - Sarbanes Oxley, SEC (also in medical domain: HIPAA), etc.
- Internal company policies about access, confidentiality, transactions
  - For security, risk management, business processes, governance
- Complexities guiding who can do what on certain business data
- Often implemented using rule techniques

- Often misunderstood or poorly implemented leading to vulnerabilities
- Typically embedded redundantly in legacy silo applications, requiring high maintenance
- Policy/Rule engines lack interoperability

Example Financial Authorization Rules

Classification	Application	Rule
Merchant	**Purchase Approval**	If credit card has fraud reported on it, or is over limit, do not approve.
Mutual Funds	**Rep trading**	*Blue Sky*: State restrictions for rep's customers.
Mortgage Company	**Credit Application**	TRW upon receiving credit application must have a way of securely identifying the request.
Brokerage	**Margin trading**	Must compute current balances and margin rules before allowing trade.
Insurance	**File Claims**	Policy States and Policy type must match for claims to be processed.
Bank	**Online Banking**	User can look at own account.
All	**House holding**	For purposes of silo (e.g., statements or discounts), aggregate accounts of all family members.

Slide also by Chitravanu Neogy

INSERT HERE: (web-browse)

- SEE: Examples of E-Services Policies Represented as Rules in RuleML:

http://ebusiness.mit.edu/bgrosof/paps/policy-rules-for-sws-v4.html

Especially see there Section 2, particularly:

- The RuleML Presentation-Syntax Primer (section 2.1)
- Creditworthiness Example
- Credit Card Transaction Authorization Example

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 -- Change management
- Quality and Transparency of implementation in enforcement
  - Provable guarantees of behavior of implementation
- 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

Outline

- Introduction
- Semantic Rules technology: commercial scene, recent developments
  - research breakthroughs → initial steps of commercial adoption
  - relationship to ontology
- Roles for rules in services
  - conceptual; lifecycle tasks; policies
  - applications; examples; business value
- Roadmapping rule-based services
  - business value; drivers
  - early adoption areas
- Roadmapping rule technology
  - research directions
  - **DISRUPTION** pattern in commercial business rules market
- Conclusions

Outline

- Introduction

- Semantic Rules technology: commercial scene, recent developments
  - research breakthroughs → initial steps of commercial adoption
  - relationship to ontology

- Roles for rules in services; roadmapping
  - conceptual; lifecycle tasks; policies
  - application areas; examples; business value; drivers

- Roadmapping rule technology
  - research directions
  - disruption pattern in commercial business rules market

- Conclusions

Rules Roadmap/Conclusions I

- Relatively recent research breakthroughs in rule KR theory and techniques
- Several of these are now rapidly moving into commercial adoption, and are helping drive standardization in semantic rules
  - Declarative LP with: well integrated ontologies; actions; defaults; and more
  - Interoperable between Prolog, RDBMS/SQL, Production Rules / ECA Rules – both backward and forward inferencing
  - Highly scalable; complexity qualitatively similar to RDBMS
  - Prioritized conflict handling enables: modularity; robustness in face of inconsistency

Rules Roadmap/Conclusions II

- Driving applications in a number of areas:
  - policy, e.g.:
    - trust, contracting (shopping, ads, discovery, exceptions), services lifecycle
  - information integration and mediation
  - social networking; combining structured and unstructured for search/navigation
  - business process communications and integration
  - verticals: financial, biomedical, military intelligence, mobile/personal communications
  - event-driven architecture, and dynamic knowledge management

Rules Roadmap/Conclusions III

- Prospects for near-term technical progress are bright, if development investment is incented, in both:
  - back-end
    - expressiveness -- from recent KR advances
    - performance -- via compilation and distributed computation
  - front-end ease of authoring and testing/validation, particularly by business users -- from:
    - improved expressive convenience/power
    - controlled natural language
    - decision tables and structured templates/forms
    - graphical and conversational interfaces
    - cheaper processing & storage
    - collaboration/communication infrastructure

Rules Conclusions IV

- Current commercially important rule systems have changed their KR relatively little in ~25 years, have
- a fragmented market with customer silo'ing/lock-in,
- high prices, and
- high customer costs for KB authoring/testing lifecycle.

Rules Roadmap/Conclusions V

- Rule market roadmapping analysis hypothesis:

standardization

→ interoperability

→ undo silo'ing/lock-in

→ major upside for customers, but grave threat to vendor price margins

→ higher volume but classic market disruption pattern for the vendors

→ shake-out

→ best-of-breed differentiation and complementarity, e.g., back-end (engine) vs. front-end (authoring)

→ & opportunities for both vendor and customer entrants

→ & pressure for solution partnering.

Rules Roadmap/Conclusions VI

- Effectual standardization of rules must provide application builders the actually required KR expressiveness.
- Most current and potential applications need features such as default negation, actions, etc., that go well beyond RIF phase 1.
- RIF has been moving slowly.
- The design approach embodied in Production LP and RuleML points the way towards where the next phase/successor of RIF should go.

- RDBMS can

  →→→ SKMS

  ("Structured Knowledge Management System")

  with semantic rules + ontologies (+ databases)

  – It's a straight incremental extension expressively
    - *the* successor to the relational model?!?

Rules Roadmap/Conclusions VII

- Relevant further R&D agenda for rules includes:
  - authoring/testing UI
  - integration/polishing of the KR advances
  - incremental reasoning, event-driven, justification/provenance/explanation
  - deeper KR integration of FOL vs. LP with nonmon and actions – needs more theory
  - exploring highly distributed, dynamic, expressive KB's & reasoning – in part, needs more theory
  - Fulfilling much of the Web Services and SOA story considerably depends on equipping services with rule-based semantic descriptions functionality, e.g., for discovery, contracting, authorization, and monitoring.
    - Plenty more to do there

Outline

- Introduction

- Semantic Rules technology: commercial scene, recent developments
  – research breakthroughs → initial steps of commercial adoption
  – relationship to ontology

- Roles for rules in services; roadmapping
  – conceptual; lifecycle tasks; policies
  – application areas; examples; business value; drivers

- Roadmapping rule technology
  – research directions
  – disruption pattern in commercial business rules market

- Conclusions

Q&A

- Thanks for your attention ☺
- Questions Invited!
- In order to compensate for the echo-y room acoustics, it might help to:
  – speak loudly, and with extra crispness (e.g., when pronouncing consonants)

Outline

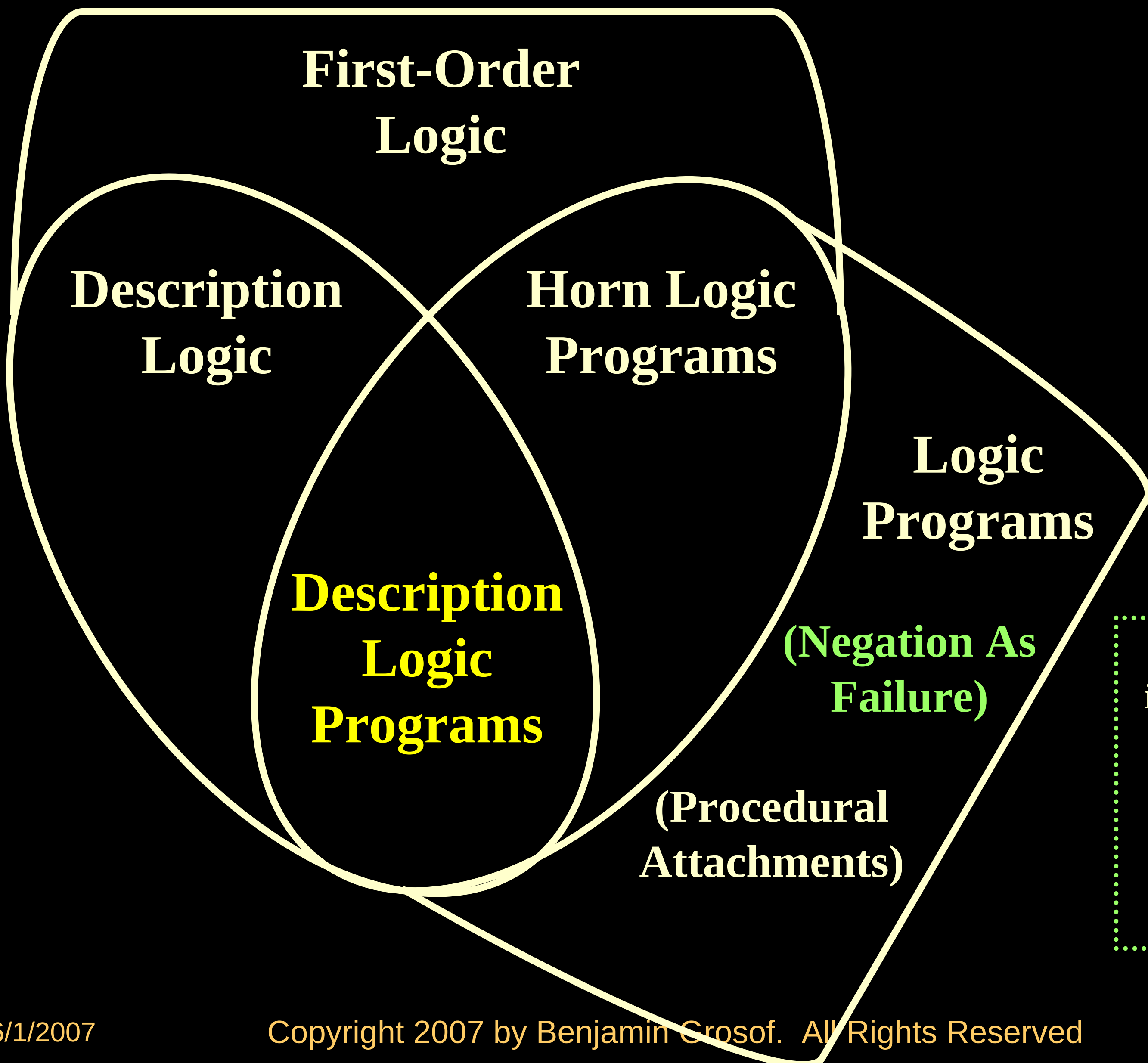
- Introduction
- Semantic Rules technology: commercial scene, recent developments
  - research breakthroughs → initial steps of commercial adoption
  - relationship to ontology
- Roles for rules in services; roadmapping
  - conceptual; lifecycle tasks; policies
  - application areas; examples; business value; drivers
- Roadmapping rule technology
  - research directions
  - disruption pattern in commercial business rules market
- Conclusions

OPTIONAL SLIDES BEGIN

Aspiration: Unifying FOL and Nonmon LP

- A challenge, a holy grail:
  - Wouldn't it be nice to have a single Knowledge Representation (KR) that unifies all of FOL and nonmon LP?
  - … or at least more of FOL and nonmon LP?
- Physics analogy: "A unified field theory for Semantic Web KR"

Venn Diagram: Expressive Overlaps among KR's



NB: Nonmon LP, including Courteous, relies on NAF as fundamental underlying KR expressive mechanism

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.
  – Semantic technology, that can be used on the web
  – Semantic = based on declarative knowledge representation

SW Overall Dependencies

- The W3C "stack" picture is a rough simplification.
- Rules do not require RDF
  – Can just use XML or even an ASCII "presentation syntax"
- Ontologies do not require RDF nor OWL
  – Ontology is a purpose for a KR which need not be Description Logic
    - Useful KR's also include FOL and LP Rules
    - Description Logic and OWL lack important features
- Customers and major vendors will be mostly still digesting XML data management in next ~3 years
  – … before moving on to heavy RDF usage

Our Research Aspects/Questions about the Semantic Web

- Core technologies: Requirements, concepts, theory, algorithms, standards?
  - Rules in combination with ontologies; probabilistic, decision-/game-theoretic
- Business applications and implications: concepts, requirements analysis, techniques, scenarios, prototypes; strategies, business models, market-level evolution?
  - End-to-end e-contracting, finance, trust; …

OPTIONAL SLIDES END