Rules Knowledge Representation for Privacy Policies: RuleML, Semantic Web Services, and their Research Frontiers

Benjamin Grosof
Douglas Drane Assistant Professor of Information Technology
MIT Sloan School of Management
http://ebusiness.mit.edu/bgrosof
including also joint work with Joan Feigenbaum, Aykut Firat, Ninghui Li, Stuart Madnick, Chitravanu Neogy, and Said Tabet

Invited Presentation (delivered by Joan Feigenbaum) at the PORTIA Workshop on Sensitive Data in Medical, Financial, and Content-Distribution Systems,
held Stanford University, Stanford, CA, July 8-9, 2004
http://crypto.stanford.edu/portia/workshops/2004_7.html

Quickie Bio of Presenter

- 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 web services is main research area:
  - Rules as core technology
  - Business Applications, Implications, Strategy:
    - e-contracting/supply-chain; finance; trust; …
  - Overall knowledge representation, e-commerce, intelligent agents
- Co-Founder, Rule Markup Language Initiative – the leading emerging standards body in semantic web rules (http://www.ruleml.org)
- Core participant in Semantic Web Services Initiative – which coordinates world-wide SWS research and early standards (http://www.swsi.org)
  - Area Editor for Contracts & Negotiation, Language Committee
  - Co-Chair, Industrial Partners program (SWSIP)

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies

- Background: Knowledge Representation Meets the Web
  - Challenge of Semantics
  - What is Knowledge Representation
  - Opportunities of the New Generation Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

Trust Policy Management

- Privacy policy management is a special case of a somewhat more generic task of trust policy management, where …
- Trust policies include for: security, access control, privacy (incl. confidentiality), & partner selection in contracting.
  - Abstract as: Authorization of access or transaction
- Policy management tasks include:
  - Represent: specify, communicate
  - Evaluate: execute/decide, monitor compliance, enforce, reason about, test, verify
  - *Underlying: capabilities for reasoning/inferencing*
- Distributed information and decision-making raise new challenges in the Internet/Web era
  - Heterogeneous sources and contexts of information
  - Heterogeneous applications do the processing

Rules for Authorization Policies

- Rules well represent authorization policies
  - Rules well represent many kinds of policies, more generally
  - Rules well represent privacy policies, more specifically
- "Rules" here means cf. declarative Logic Programs (LP) knowledge representation, for which the emerging industry standard is RuleML (i.e., Semantic Web / XML rules)
- E.g., *if __complex condition C1__ then permit access to __resource R__ ;*
- E.g., *if __complex condition C2__ then deny access to __resource R__ ;*
- E.g., *if __complex condition C3__ then __intermediate condition C4__ ;*

- Examples:
  - RBAC (Role Based Access Control)
    - The most important and widely deployed kind of trust policy mechanism
  - XACML (eXtensible Access Control Markup Language)
    - The most important emerging standard for XML-info access control
  - P3P (Platform for Privacy Preferences)
    - The most important standard for Web browser client privacy policies

Overall Suggested Research Directions for Privacy, and Open Questions

- Design privacy policy languages and standards
  - Reformulate and improve early-version/emerging standards for Web privacy including XACML, P3P, and Web Services policies, and good old RBAC
    - Develop semantic foundations, algorithms
- Use and extend declarative Logic Programs knowledge representation cf. RuleML (i.e., Situated Courteous Description Logic Programs) to represent and evaluate privacy policies.
  - Try out modern rule KR – esp. Situated Courteous Logic Programs and associated tools – for privacy scenarios
  - Open source tools available, e.g., updated SweetRules (from B. Grosof's group + collaborators) soon on SemWebCentral.org
  - Extend the underlying rule KR expressively as necessary, e.g., with privacy-/policy- specific constructs
    - E.g., do we want an Ignorance operator in the language? If so, is it adequate to use one that is expressively reducible to negation-as-failure?

Overall Suggested Research Directions for Privacy, and Open Questions, cont.'d

- Embrace movement towards Semantic Web, Semantic Web Services.
  - In particular, newly use for privacy the emerging SW knowledge representation technologies/standards for rules and also ontologies (structured vocabularies with subclass hierarchy, domain, range).
  - SW community quite interested in trust overall, but not yet focused on privacy.
  - Address privacy within Web Services / Semantic Web Services
- Focus on financial, medical, police/military domains as prospects for early adoption by industry/government.
- Explore privacy in the context of powerful information integration, where inferencing (e.g., from a distributed set of rulebases + ordinary databases) can result in "leakage" of private information.
- Overall: Combine KR with crypto and social policy mechanisms.
  - Rules good to represent regulations

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies

- Background: Knowledge Representation Meets the Web
  - Challenge of Semantics
  - What is Knowledge Representation
  - Opportunities of the New Generation Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

Challenge: Capturing Semantics

- Deep challenge is to capture the semantics of data and processes, so that can:
  - Represent, monitor, and enforce policies – e.g., trust and contracts
  - Map between definitions of entities, e.g., in financial or medical domains
  - Integrate policy-relevant information powerfully
- Best tool available today:
  - Knowledge Representation (the field of it)

Background: What is Knowledge Representation (KR)?

- The *field* of KR studies and designs particular knowledge representation languages/systems (KR's).
- A KR includes:
  - A formal language for expressing premises.
  - A formal language for expressing conclusions.
  - A set of entailment principles that together, for any given set of premises, formally defines an associated set of sanctioned conclusions.
    - In "declarative" KR, these principles are independent of inferencing procedure/control-strategy, and thus constitute a semantics, e.g., a model theory.

Background: What is Knowledge Representation (KR)? – cont.'d

- Usage scenarios drive choice/design of KR.
  - Domain of application; domain of knowledge available.
  - Need sufficient & convenient expressiveness. ⇒ Seek extensions of KR.
- Computational scalability/tractability is a critical consideration.
  - ⇒ Seek restrictions on KR.
- ⇒ Concepts; Theory on language, semantics, complexity; algorithms/techniques; application scenarios; standards incl. for syntax; prototyping of tools, scenarios, applications.

Background: Example KR's

1. Relational databases: relational algebra.
   - This is a restricted form of declarative Logic Programs ("Datalog Horn").
2. Mathematical classical logic: first-order logic (FOL), higher-order logic.
   - Used in verification of programs, for example.
3. Rules in various flavors.
   - Central abstraction: declarative Logic Programs, which extend Horn FOL.
     - (Core) SQL database is an LP rulebase.
4. Many others: Bayesian probabilistic networks, inductive learning, Description Logic, fuzzy logic, temporal modal logic, etc.

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.)*

Knowledge Representation: What's the Game?

- Expressiveness: useful, natural, complex enough
  - Consider usage scenarios
- Semantics: principles of sanctioned inference, independent of reasoning algorithms
- 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
- Syntax: encoding data format, -- here, in XML

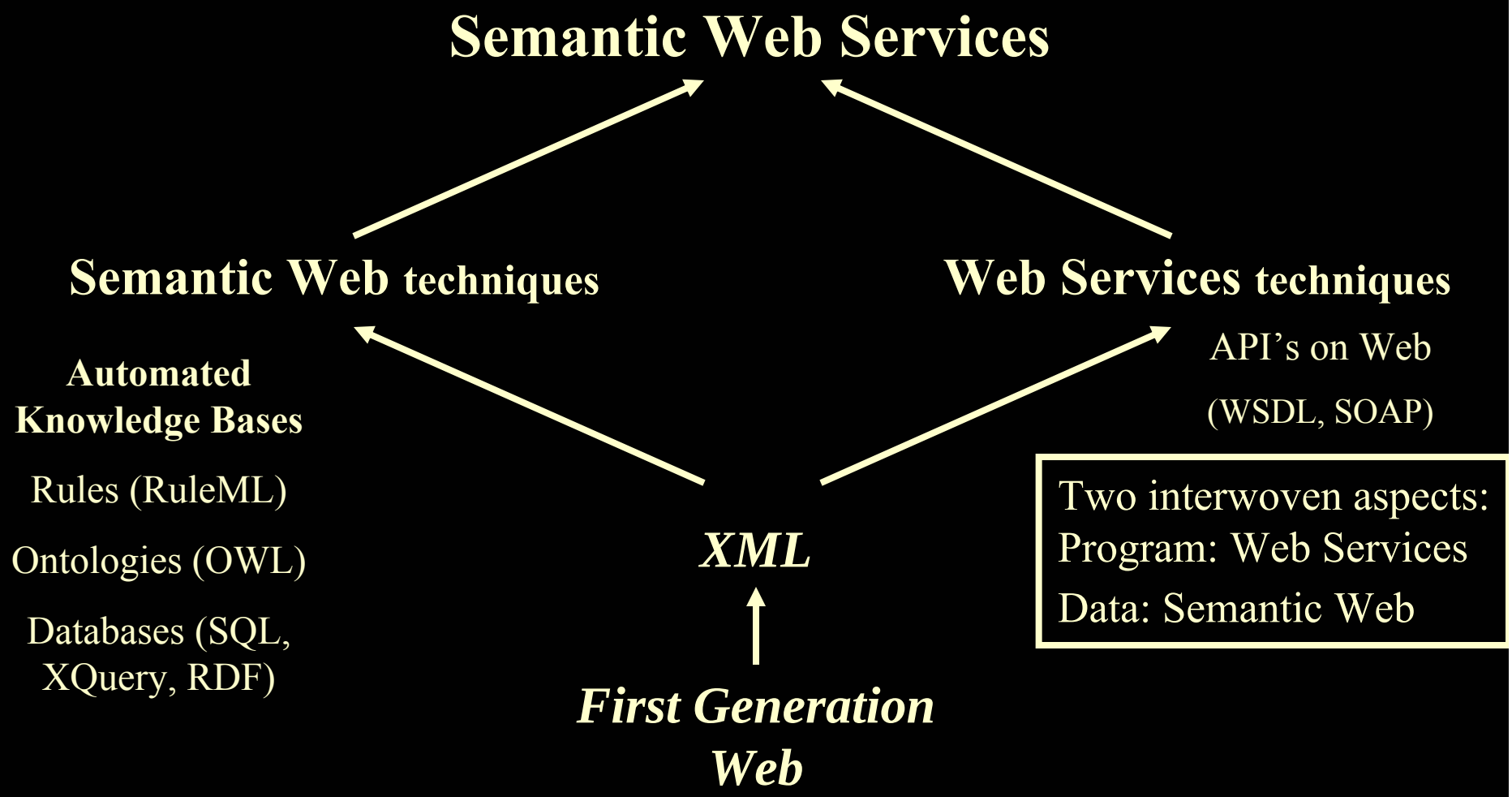
Opportunity from Semantic Web Services -- the New Generation Web Platform

- Semantic Web = Knowledge Representation on the Web.
- New technologies for Rules (RuleML* standard, based on Situated Courteous Logic Programs knowledge representation)
  - + New technologies for Ontologies** (OWL standard)
  - + Databases (SQL, XQuery, RDF)
  - + Web Services (WSDL, SOAP, J2EE, .Net)

- Status today:
  - Technologies: emerging, strong research theory underneath
  - Standards activities: intense (W3C, Oasis, …)
  - Commercialization: early-phase (majors in alpha, startups)

*(* RuleML = Rule Markup Language (a.k.a. Rule Modeling Language) emerging standard.*
*** Ontology = structured vocabulary, e.g., with subclass-superclass, domain, range, datatypes. E.g., database schemas. OWL = W3C's Ontology Web Langauge.)*

Next Generation Web



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

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies

- Background: Knowledge Representation Meets the Web
  - Challenge of Semantics
  - What is Knowledge Representation
  - Opportunities of the New Generation Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

SWS 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

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

- Trust Policies usually well represented as rules
  - Evaluation 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.
    - Is easily recast as LP/RuleML rules [e.g., see Ninghui Li *et al.* papers]
  - W3C P3P privacy standard, Oasis XACML XML access control emerging standard, …
  - Broad approach: layer policy-particular constructs/language on top of generic rule KR. E.g., add {permit, deny, must} constructs. E.g., see Rei language by Lalana Kagal *et al.* (her PhD dissertation nearing completion; B. Grosof is on her thesis committee)

Policies for Compliance and Trust Mgmt.: Role for Semantic Web Rules, cont.'d

- Many Business Policies beyond trust arena, too, are well represented as rules
  - "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

Advantages of Standardized SW Rules

- Principled and tested techniques and implementations
- Reuse of previous theory, techniques, implementations, training
- Standardization network effect (virtuous circle) of social/business investment
- Easier Integration: with rest of business policies and applications, business partners, mergers & acquisitions
- Lower cognitive complexity in: training, familiarity, rule authoring cost, requirements analysis, marketing
  - Easier to understand and modify by humans

Advantages of Standardized SW Rules, cont.'d

- Quality and Transparency of implementation in enforcement
  - Provable guarantees of behavior of implementation
  - Much lessened risk of backdoors
- Reduced Vendor Lock-in
- Expressive power
  - Principled handling of conflict, negation, priorities
  - More: logical functions, recursion, drawing upon distributed knowledge bases, discipline wrt side-effectful actions triggered by rules
- Theoretical guarantees incl. wrt scalability/tractability, consistency

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

- Reduced system dev./maint./training costs
- Better/faster/cheaper policy admin.
- Interoperability, flexibility and re-use benefits
- Greater visibility into enterprise policy implementation => better compliance
- Centralized ownership and improved governance by Senior Management
- Rich, expressive trust management language allows better conflict handling in policy-driven decisions

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies

- Background: Knowledge Representation Meets the Web
  - Challenge of Semantics
  - What is Knowledge Representation
  - Opportunities of the New Generation Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

eXtensible Access Control Markup Language (XACML)

- Oasis XACML is leading technical standard for access control policies in XML
  - Access to XML info
  - Policies in XML
- Uses a rule-based approach
  - Including for prioritized combination of policies
- Status: Emerging
- Needs a formal semantics -- and a more principled and standardized approach to rules KR, generally.
  - Research opportunity!

Platform for Privacy Preferences (P3P)

- W3C P3P is leading technical standard for privacy policies representation and enforcement
- Client privacy policies specified in a simple rule language (APPEL, part of P3P)
- Has not achieved great usage yet
  - Microsoft dominance of browsers a strategic issue
- Needs a formal semantics -- and a more principled and standardized approach to rules KR, generally.
  - Research opportunity!

Web Services Trust Policy Management

- Web Services (WS) area is evolving quickly
- Emerging hot area: WS policy management, including for security/trust -- which includes privacy
  - Defined as next-phase agenda in standards efforts, major vendor white papers/proposals (e.g., Microsoft, IBM)
  - Semantic Web Services research in this is growing, e.g., DAML-Security effort
- Research opportunity!

Verticals that appear good candidates for Early Adoption of SW Rules for Privacy

- Financial
  - Cf. discussion earlier in this talk
  - Historically, an early adopter of information technology overall esp. for integration
  - Large sector of global economy
  - Privacy/trust policies very important, distributed & heterogeneous
- Medical
  - Privacy/trust policies very important, distributed & heterogeneous
  - Expecting help on privacy from information technology
  - Large sector of global economy
- Police/Military
  - Privacy/trust policies very important, distributed & heterogeneous
  - Looking for help on privacy from information technology
  - Major funder of SWS basic research to date, e.g., DARPA Agent Markup Language program 2000-2005
- In many other realms, there's a large gap between revealed vs. avowed preferences for value of privacy/confidentiality.

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.

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies
  - Challenge of Semantics
  - Opportunities of the New Generation Web

- Background: Knowledge Representation Meets the Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

Overall Suggested Research Directions for Privacy, and Open Questions

- Design privacy policy languages and standards
  - Reformulate and improve early-version/emerging standards for Web privacy including XACML, P3P, and Web Services policies, and good old RBAC
    - Develop semantic foundations, algorithms
- Use and extend declarative Logic Programs knowledge representation cf. RuleML (i.e., Situated Courteous Description Logic Programs) to represent and evaluate privacy policies.
  - Try out modern rule KR – esp. Situated Courteous Logic Programs and associated tools – for privacy scenarios
  - Open source tools available, e.g., updated SweetRules (from B. Grosof's group + collaborators) soon on SemWebCentral.org
  - Extend the underlying rule KR expressively as necessary, e.g., with privacy-/policy- specific constructs
    - E.g., do we want an Ignorance operator in the language? If so, is it adequate to use one that is expressively reducible to negation-as-failure?

Overall Suggested Research Directions for Privacy, and Open Questions, cont.'d

- Embrace movement towards Semantic Web, Semantic Web Services.
  - In particular, newly use for privacy the emerging SW knowledge representation technologies/standards for rules and also ontologies (structured vocabularies with subclass hierarchy, domain, range).
  - SW community quite interested in trust overall, but not yet focused on privacy.
  - Address privacy within Web Services / Semantic Web Services
- Focus on financial, medical, police/military domains as prospects for early adoption by industry/government.
- Explore privacy in the context of powerful information integration, where inferencing (e.g., from a distributed set of rulebases + ordinary databases) can result in "leakage" of private information.
- Overall: Combine KR with crypto and social policy mechanisms.
  - Rules good to represent regulations

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies

- Background: Knowledge Representation Meets the Web
  - Challenge of Semantics
  - What is Knowledge Representation
  - Opportunities of the New Generation Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

For More Info

- Please contact Benjamin Grosof
- http://ebusiness.mit.edu/bgrosof
- bgrosof@mit.edu

OPTIONAL SLIDES FOLLOW about Misc.

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.

3 Areas of New Fundamental KR Theory

that enable Key Technical Requirements for SWS

(advances by B. Grosof et al. in last decade, underlying RuleML)

- 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

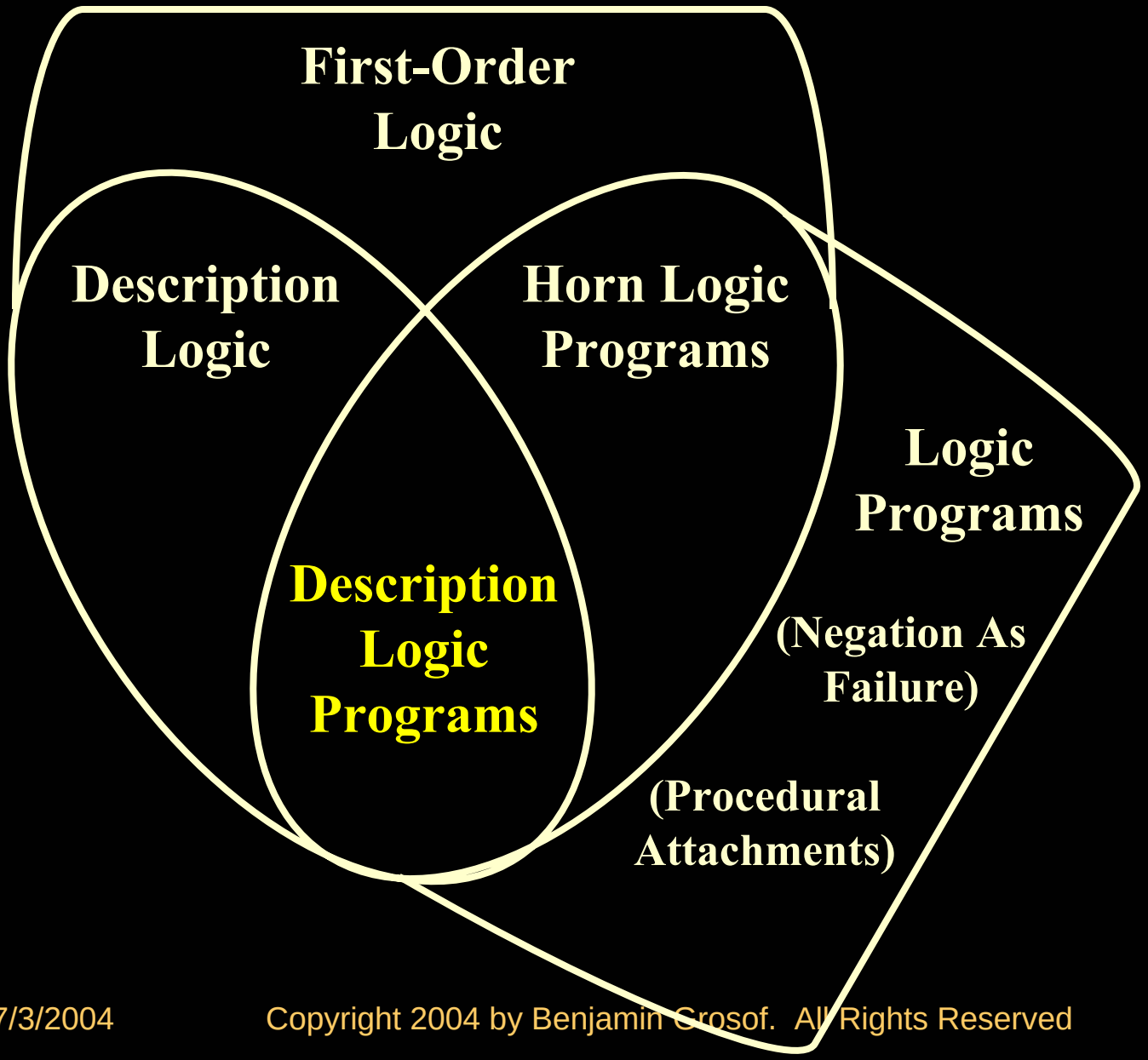
  - Rules use services, e.g., to query, message, act with side-effects
  - Rules constitute services executably, e.g., workflow-y business processes

- 3. Courteous Logic Programs:

  KR to combine rules from many sources, with:

  - Prioritized conflict handling to enable consistency, modularity; scaleably
  - Interoperable syntax and semantics

Venn Diagram: Expressive Overlaps among KR's



Example I – Credit Card Verification System

- Typical for eCommerce websites accepting credit cards – Visa, MC, Discover, Amex
- Rules for transaction authorization
  - Bank performs account limit, expiration, address and card code verification
  - A fraud alert service may flag a card
  - Service provider may blacklist customer
- Overrides, e.g., alert service over bank rules

Example II – Brokerage Access Control

- Need protection of customer accounts of retail (own) and many client correspondents from unauthorized access by traders (reps)
- Many Complex Rules for access control
  - Retail reps can look at any retail account but not correspondent accounts
  - A correspondent user may look at accounts for their organization but…
  - Only from those branches over which rep's branch has fiduciary responsibility
  - For certain branches, customer accounts are explicitly owned by certain reps and cannot be divulged even to his partner!
- More rules, with several overrides

CommonRules Implementation for Credit Card Verification Example

Sample Rule Listing

```
<bankResp>
  if checkTran(?Requester)
  then
    transactionValid(self,?Requester);
<cardRules2>
      if      checkCardDet(?Requester, ?accountLimit, ?exp_flag, ?cardholderAddr,
      ?cardholderCVC) and
      checkTranDet(?Requester, ?tranAddr, ?tranCVC) and
      notEquals(?tranCVC, ?cardholderCVC)
      then
      CNEG transactionValid(self,?Requester);
…
overrides(cardRules2, bankResp);
checkTran(Joe);
checkCardDet(Joe, 50, "false", 13, 702);
checkTranDet(Joe, 13, 702);
cardGood(Fraudscreen.net,Joe,good);
customerRating(Amazon.com, Joe, good);
```

CommonRules translates straightforwardly ↔ RuleML.

We show its human-oriented syntax as a presentation syntax for RuleML.

Runtime Results for Credit Card Verification

Sample Output

SCLPEngine: Adorned Derived Conclusions:

CNEG transactionValid_c_3(self, Mary);
transactionValid_c_2(self, Joe);
transactionValid_c_2(self, Mary);
transactionValid_r_2(self, Mary);
transactionValid_u(self, Joe);
CNEG transactionValid_u(self, Mary);

transactionValid(self, Joe);
CNEG transactionValid(self, Mary);

Adorned **conclusions represent intermediate phases of prioritized conflict handling in Courteous Logic Programs**

CNEG **= limited classical negation (which is permitted in Courteous LP)**

CNEG p means p is (believed to be) false

Self **= the agent making the authorization decision, i.e., the viewpoint of this local rulebase.**

(This is as usual in trust management.)

Outline

- Introduction
  - Privacy policies as special case of trust management
  - Rules well represent authorization policies
  - Challenge of Semantics
  - Opportunities of the New Generation Web

- Background: Knowledge Representation Meets the Web
  - Semantic Web, Web Services, Semantic Web Services
  - RuleML and Situated Courteous Logic Programs (SCLP)
  - E-Contracting and Trust Policies, e.g., in Supply Chain and Finance

- Privacy Policies – the Landscape Today
  - RBAC
  - XACML, P3P
  - Regulatory and Compliance Initiatives: Sarbanes-Oxley, HIPAA, etc.

- Advantages of Standardized Semantic Web Rules (SCLP RuleML)
  - Examples: Financial Trust Policies, e.g., Brokerage Account Access

- Research Opportunities, Challenges & Directions
  - Use and extend SCLP RuleML
  - RBAC, XACML, P3P, Web Services
  - Financial, medical, police/military

OPTIONAL SLIDES FOLLOW
about 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 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

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 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

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.

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

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

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

Research Aspects/Questions about the New Generation 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; …

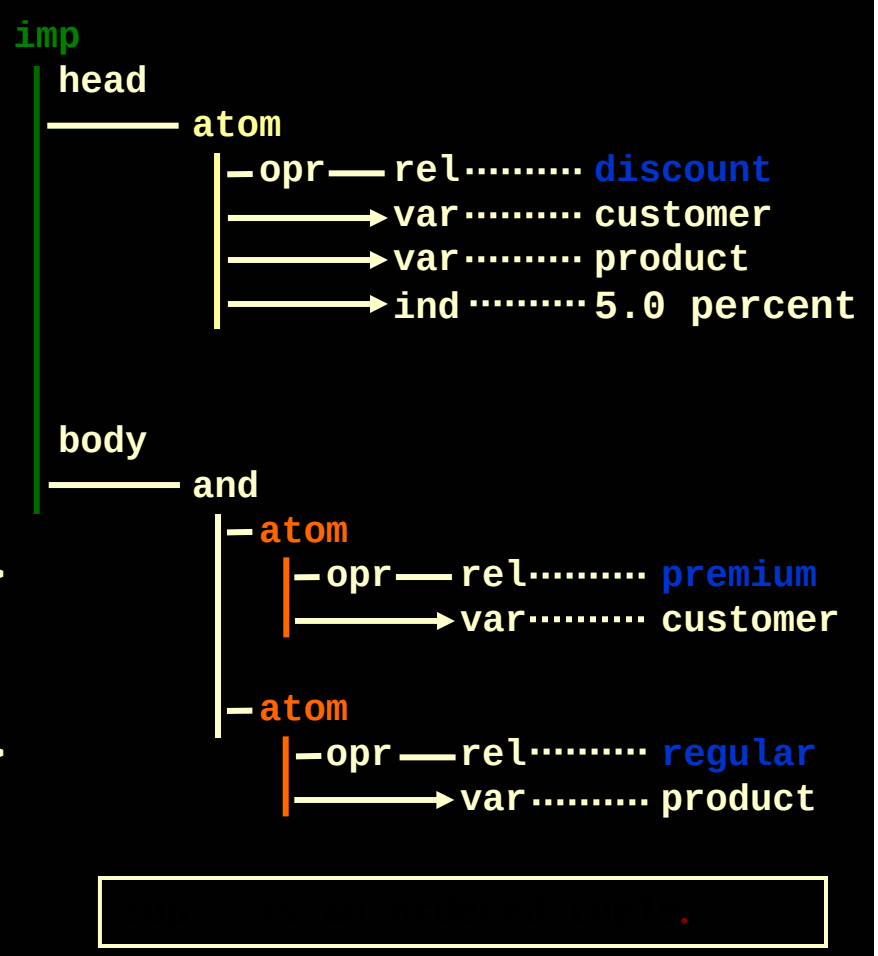
RuleML Example: Markup and Tree

"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

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>
```

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!**

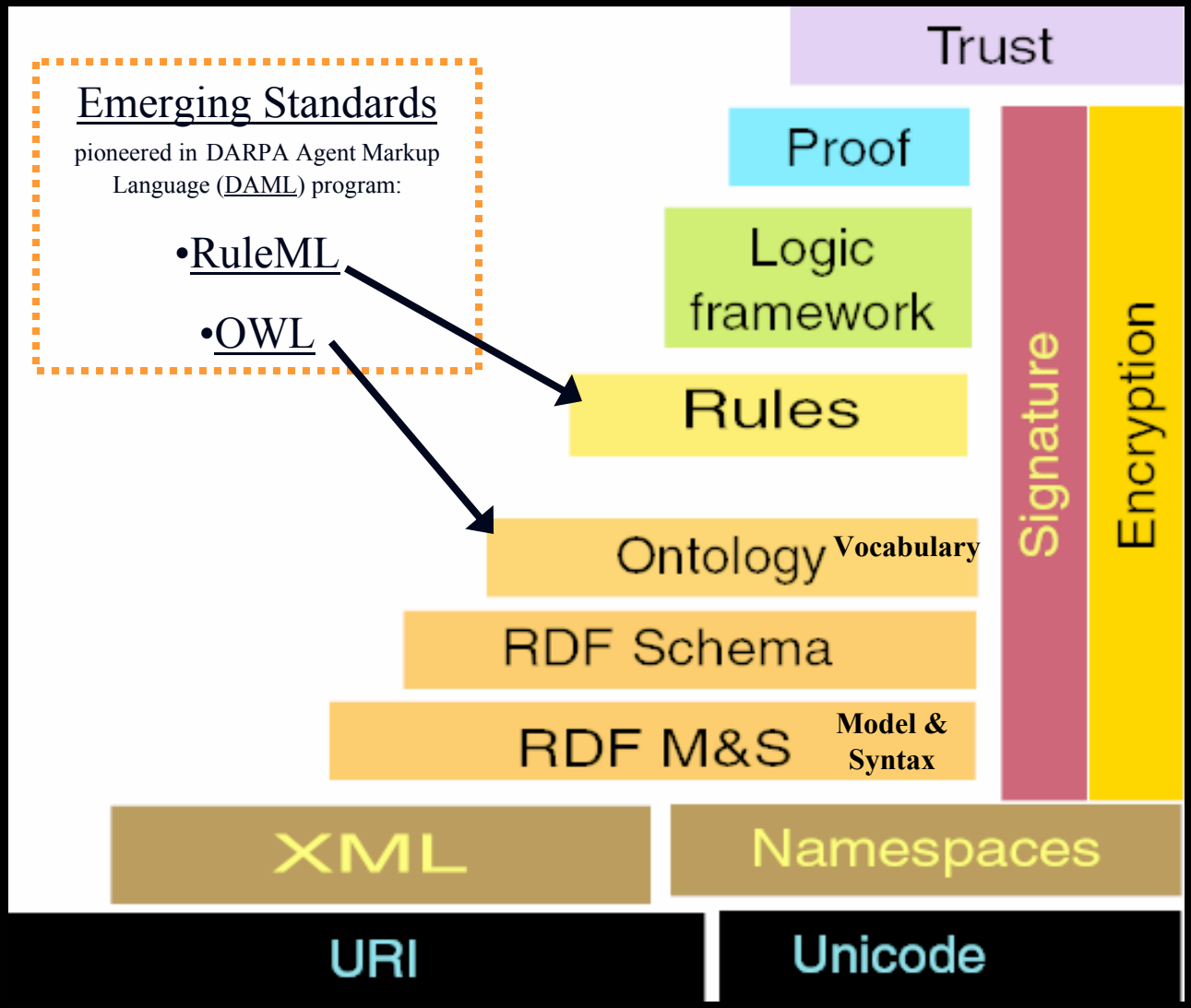
- Helpful Approach: **precedence** between the rules. Often only *partial* order of precedence is justified. E.g., C > A.

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.

MORE OPTIONAL SLIDES FOLLOW about Semantic Web

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.

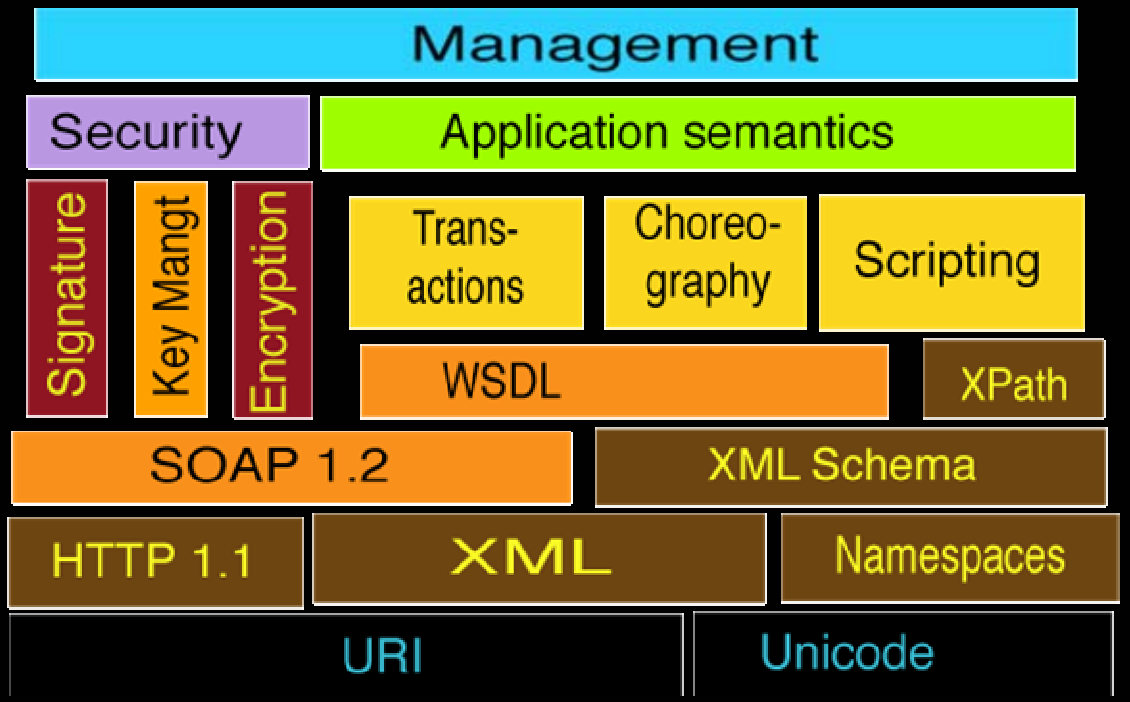
SW: Standards Players

- 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.

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

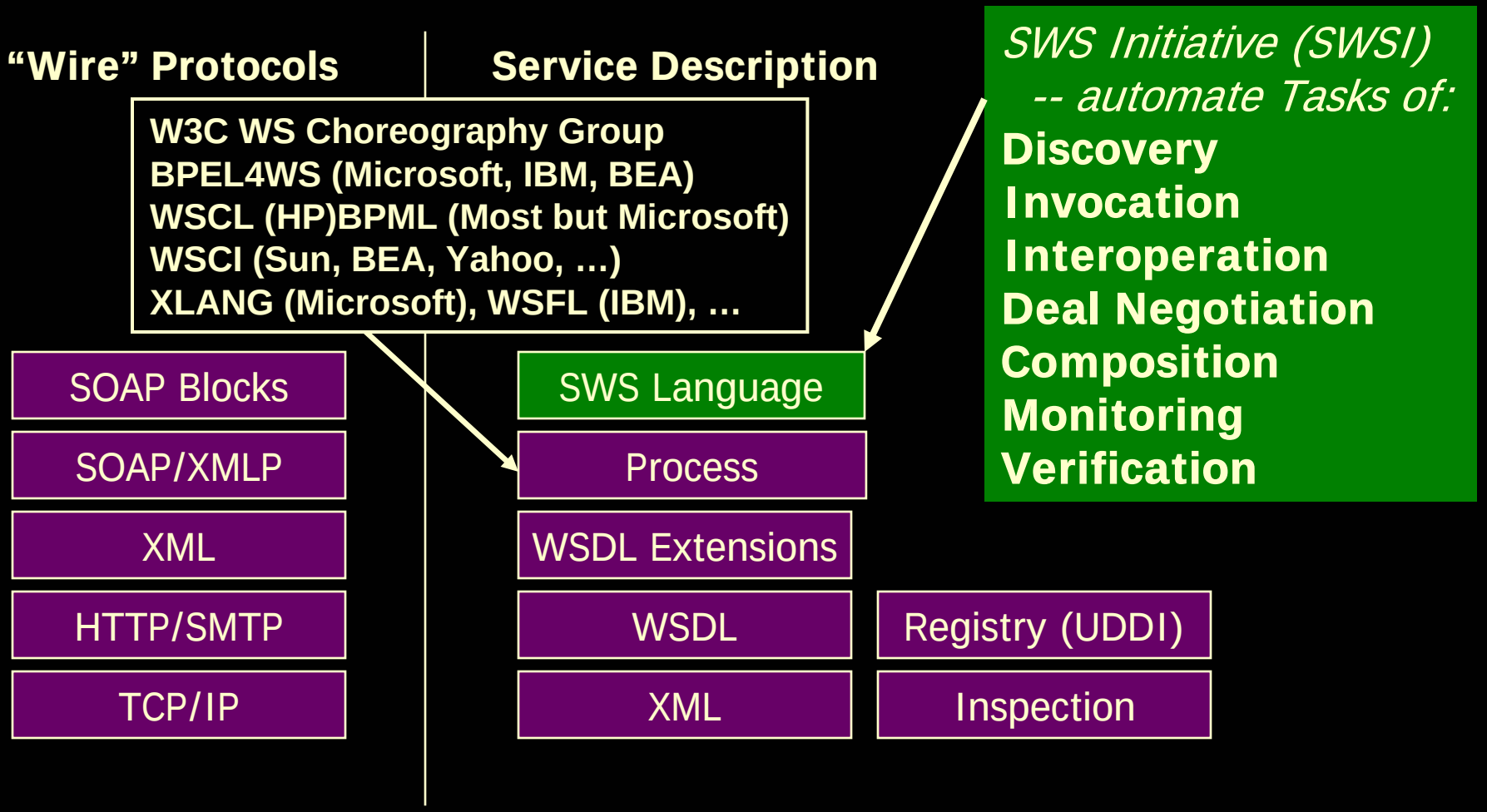
WS Stack: some Acronym Expansion

- 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

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