Trust Policies using Rules, with Applications in Financial Services

Benjamin Grosof

Douglas Drane Assistant Professor of Information Technology

MIT Sloan School of Management

http://ebusiness.mit.edu/bgrosof

Chitravanu (Chitro) Neogy

Masters Of Technology student

MIT Sloan School of Management

http://www.chitro.com

Said Tabet

Macgregor Inc. and RuleML Initiative

stabet@comcast.net

including also joint work with Aykut Firat, Stuart Madnick, Joan Feigenbaum, Ninghui Li

Presentation at the WWW-2004 Developers Day Trust on the Web track

http://www.www2004.org . Held May 22, 2004, NYC.

Outline

- Introduction
  - Challenge of Semantics
  - Opportunities of the New Generation Web
- Policies and Compliance
  - Landscape Today
  - Advantages of Standardized Semantic Web Rules (Situated Courteous Logic Programs in RuleML)
  - Examples: Financial Trust Policies, e.g., Credit Verification
- Financial Information Integration
  - Conflicting Definitions in Business Reporting
  - Mapping Approach (Extended COntext INterchange)
- Research Challenges & Directions

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 reporting
  - Integrate policy-relevant information powerfully

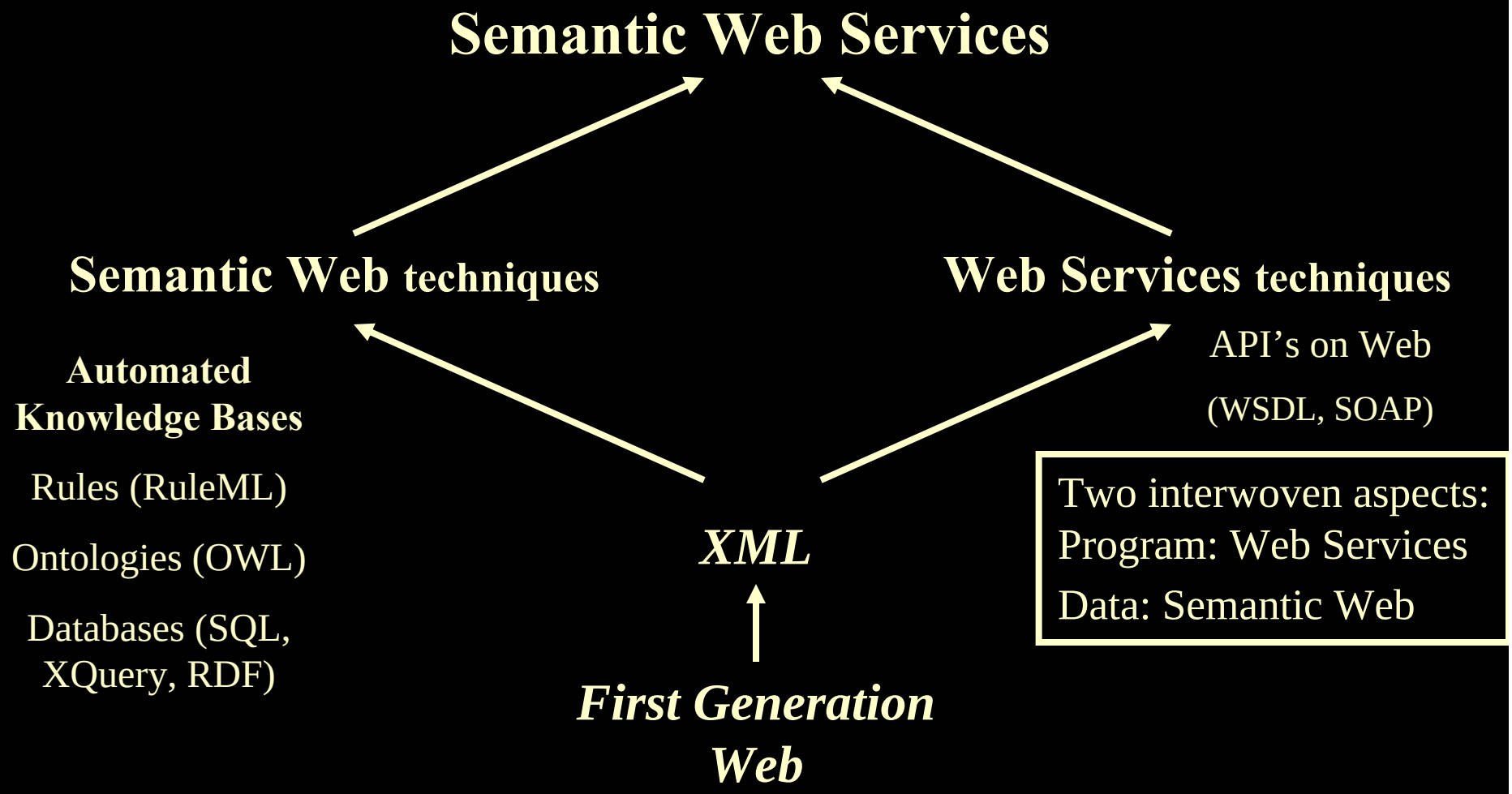
Opportunity from Semantic Web Services -- the New Generation Web Platform

- New technologies for Rules (RuleML standard, based on Situated Courteous Description 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)

(Ontology = structured vocabulary, e.g., with subclass-superclass, domain, range, datatypes. E.g., database schemas.)*

Next Generation Web



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

Outline

- Introduction
  - Challenge of Semantics
  - Opportunities of the New Generation Web
- Policies and Compliance
  - Landscape Today
  - Advantages of Standardized Semantic Web Rules (Situated Courteous Logic Programs in RuleML)
  - Examples: Financial Trust Policies, e.g., Credit Verification
- Financial Information Integration
  - Conflicting Definitions in Business Reporting
  - Mapping Approach (Extended COntext INterchange)
- Research Challenges & Directions

Policies and Compliance in US Financial Industry Today

- Ubiquitous high-stakes Regulatory Compliance requirements
  - Sarbanes Oxley, SEC, 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.

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

Advantages of Standardized SW Rules

- Easier Integration: with rest of business policies and applications, business partners, mergers & acquisitions
- Familiarity, training
- Easier to understand and modify by humans
- 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

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

- 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

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

Example III – Check 21

- "Hot" legislation to enable electronic check transfer between banks via substitutes.
- Rules for substitute check validation
  - Federal Reserve Board: needs match on check back/front image, MICR, BOFD
  - ANSI X9: match MICR, size
  - The Banks: must accept substitute checks
  - Country of Origin=US and EPC=2 or 5

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
  - Challenge of Semantics
  - Opportunities of the New Generation Web
- Policies and Compliance
  - Landscape Today
  - Advantages of Standardized Semantic Web Rules (Situated Courteous Logic Programs in RuleML)
  - Examples: Financial Trust Policies, e.g., Credit Verification
- Financial Information Integration
  - Conflicting Definitions in Business Reporting
  - Mapping Approach (Extended COntext INterchange)
- Research Challenges & Directions

Equational Ontological Conflicts in Financial Reporting

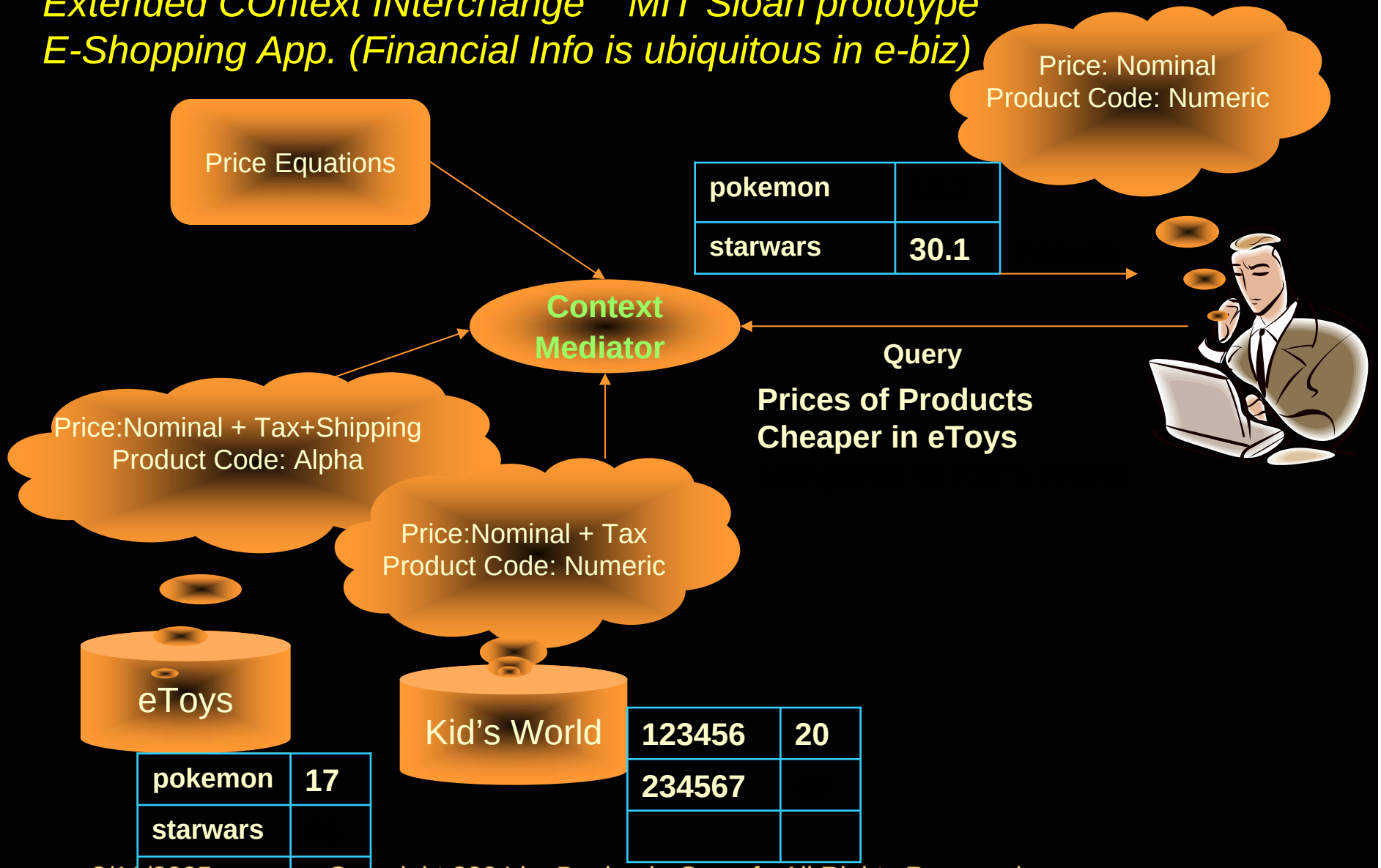
# of customers = # of end_customers + # of distributors	# of customers = # of end_customers + # of prospective customers
Gross Profit = Net Sales – Cost of Goods	Gross Profit = Net Sales – Cost of Goods – Depreciation
P/E Ratio = Price / Earnings(last 4 Qtr)	P/E Ratio = Price/ [Earnings(last 3 Qtr) +Earnings(next quarter)]
Price = Nominal Price + Shipping	Price = Nominal Price + Shipping + Tax

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

Solution Approach: ECOIN

Extended COntext INterchange MIT Sloan prototype

E-Shopping App. (Financial Info is ubiquitous in e-biz)



Approach: ECOIN

- Context-based loosely-coupled integration

  Extends the Context Interchange (COIN) framework developed at MIT

- Symbolic Equation Solving using Constraint Logic Programming

  Integrates symbolic equation solving techniques with abductive logic programming

Outline

- Introduction
  - Challenge of Semantics
  - Opportunities of the New Generation Web
- Policies and Compliance
  - Landscape Today
  - Advantages of Standardized Semantic Web Rules (Situated Courteous Logic Programs in RuleML)
  - Examples: Financial Trust Policies, e.g., Credit Verification
- Financial Information Integration
  - Conflicting Definitions in Business Reporting
  - Mapping Approach (Extended COntext INterchange)
- Research Challenges & Directions

Research Challenges: Core

- Integrating rules with ontologies
  - Rules refer to ontologies (e.g., in RuleML)
  - Rules to specify ontologies (e.g., Description Logic Programs)
  - Rules to map between ontologies (e.g., ECOIN)
  - Combined rules + ontologies knowledge bases (e.g., RuleML + OWL)

- Describing business processes & web services via rules + ontologies
  - Rules query web services (e.g., in RuleML Situated feature)
  - Rules trigger actions that are web services (e.g., ditto)
  - Capture object-oriented process ontologies
    - Default inheritance via rules (e.g., Courteous Inheritance)
    - Wrapper/transform to legacy C++, Java, UML
    - Develop open source knowledge bases (e.g., MIT Open Process Handbook Initiative)
  - Event triggering of rules (e.g., capture ECA rules in RuleML)

Research Challenges: Business Policies

- Apply advanced rule and ontology representation to business policies in compliance, trust, contracts, etc.
  - Application scenarios for compliance checking/support services intra- and inter- enterprise
  - Policy language & engines on top of rule language & engines
  - In/with existing/emerging standards: XBRL, XACML, P3P, ebXML, EDI, Legal XML, …
  - Strategy and roles in the market ecology: regulators, communal repositories, service providers, etc.
  - Embedding into the bigger pictures of financial services, e-commerce, semantic web services, business process automation

Context: Our Overall Research Agenda

- Invent Core Technologies and concepts of the New Generation Web
  - Semantic Web; Rules and RuleML emerging standard
    - supporting knowledge representation theory of Situated Courteous Description Logic Programs
  - Semantic Web Services; Business Process Automation for B2B and EAI
    - Requirements analysis
    - Use of Rules, together with ontologies – or to represent ontologies
- Pilot Business Application Scenarios
  - End-to-end e-contracting, e.g., in manufacturing supply chain
    - SweetDeal approach using rules (plus ontologies)
  - Financial information and reporting:
    - ECOIN approach mapping ontologies
  - Other: security authorization, travel, …
- Analyze Prospective Early Adopter Areas
  - Strategy: Adoption Roadmap; Market Evolution
  - Entrepreneurial Opportunities

More Strategic Opportunities in Compliance

- XBRL (eXtensible Business Reporting Language):
  - SWS rules + ontologies can reduce degree of industry consensus required to enable interoperability
    - Difficult to get agreement on single definition of “earnings”; easier to agree on “long-term capital gains realized from sale of real estate assets”.
    - Translate between different use contexts’ ontologies
- SEC and other regulatory agencies:
  - They can accelerate compliance
    - via providing automated SWS specifications of regulations and reporting forms (+ the instructions)
      - e.g., RuleML regulatory rulebases accessible via Web Services interfaces

Outline

- Introduction
  - Challenge of Semantics
  - Opportunities of the New Generation Web
- Policies and Compliance
  - Landscape Today
  - Advantages of Standardized Semantic Web Rules (Situated Courteous Logic Programs in RuleML)
  - Examples: Financial Trust Policies, e.g., Credit Verification
- Financial Information Integration
  - Conflicting Definitions in Business Reporting
  - Mapping Approach (Extended COntext INterchange)
- Research Challenges & Directions

OPTIONAL SLIDES FOLLOW

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)

Trust Management Policies: Role for Semantic Web Rules

- Policies usually well represented as rules
  - E.g., Role-based Access Control
    - This is the most important kind of trust policy in practical deployment today.
- Advantages of standardized SW rules:
  - Familiarity, training
  - Quality and Transparency of implementation in enforcement
  - Reduced Vendor Lock-in
  - Expressive power
  - Integration with rest of business policies

Trust in larger context of Business Policies and Contracts

- Trust/authorization is often closely tied to other business policies, e.g., pricing, bidding, customer selection, lead-time, service level. E.g.,
  - Risk of new business partner B, when supplier S makes bid.
    - ? Will B fulfill its commitments if B places an order?
    - ? Will S lose by reserving capacity while awaiting B's decision?
    - ? Will B leak information to competitors about B's pricing & capacity?

- *From another viewpoint:* Trust is what contracts are all about:
  - Contracts encode agreements that define conditions of trust.

Discussion

- Gray areas: trust/security/privacy policies vs. other business policies
  - Risk vs. benefit

Rule-based Policies for Trust and Security Authorization

- Use rule-based executable specification of security authorization policies, a.k.a. trust management: including delegation, certificates.
  - Straightforwardly generalizes Role-Based Access Control (RBAC).
- Often, authorization/trust policy is really a part of overall contract or business policy, at application-level. Unlike authentication.
- Advantages of rule-based approach, esp. from declarative semantics:
  - easier integration with general business policy.
  - easier to understand and modify by humans.
  - provable guarantees of behavior of implementation.
  - principled handling of negation and conflict.

Delegation Logic: Goal and Basic Approach*

- Goal: Develop a language that
  - can represent, with significant expressive power, *policies* and *credentials* for authorization in Internet scenarios
  - can provide mechanisms for delegation
  - has a clear declarative semantics

- Approach: Delegation Logic (DL): multi-agent logic programs with delegation to complex delegatees
  - D1LP: extends negation-free OLP $\Rightarrow$ with delegation
  - D2LP: extends Courteous LP $\Rightarrow$ with delegation
  - Tractable "Delegation compiler" similar to courteous compiler.

* [Li, Grosof, & Feigenbaum, ACM Transactions on Information Systems Security 2003]

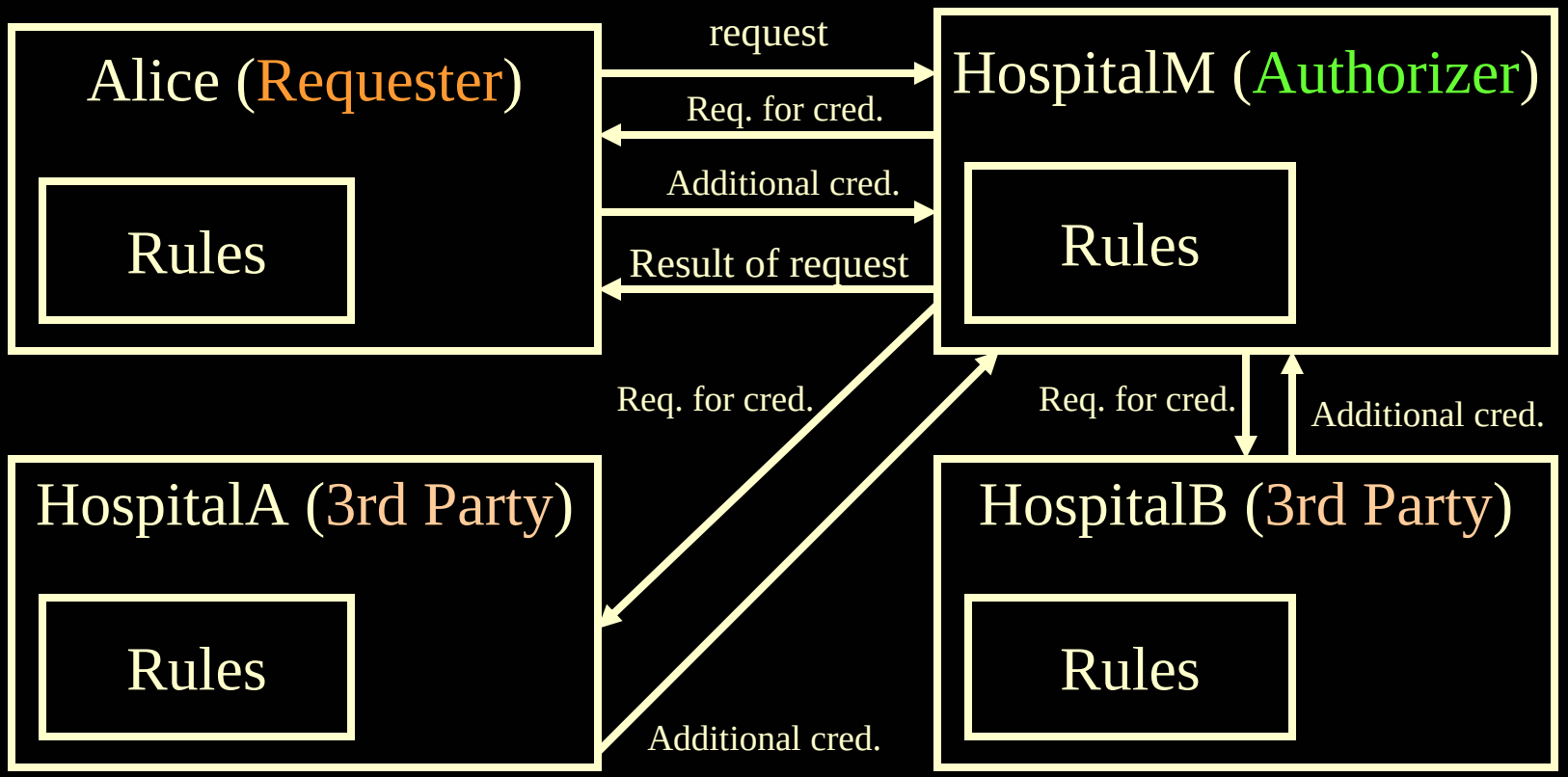
Delegation Logic (D1LP) Example: accessing medical records

- 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.*

Example Scenario of Delegation Logic

- Each agent is a principal; in a given scenario one is a requester.
- Each agent initially has a ruleset, that represents policies and/or credentials.
- Agent 1 as requester sends a request to Agent 2 as authorizer.
- The authorizer evaluates the request by executing the authorizer's policies:
  - Performs situated inference of the policy rules.
  - During evaluation, the authorizer also queries other agents (3rd parties, or the requester) for additional relevant credentials (rules).
    - Other agents, when queried, respond by sending credentials to the authorizer.
- After evaluation, the authorizer informs the requester about the decision.

Example Scenario Information Flow



What is a Delegation Relationship?

- What relationships can be viewed as a delegation from *Alice* to *Bob*?

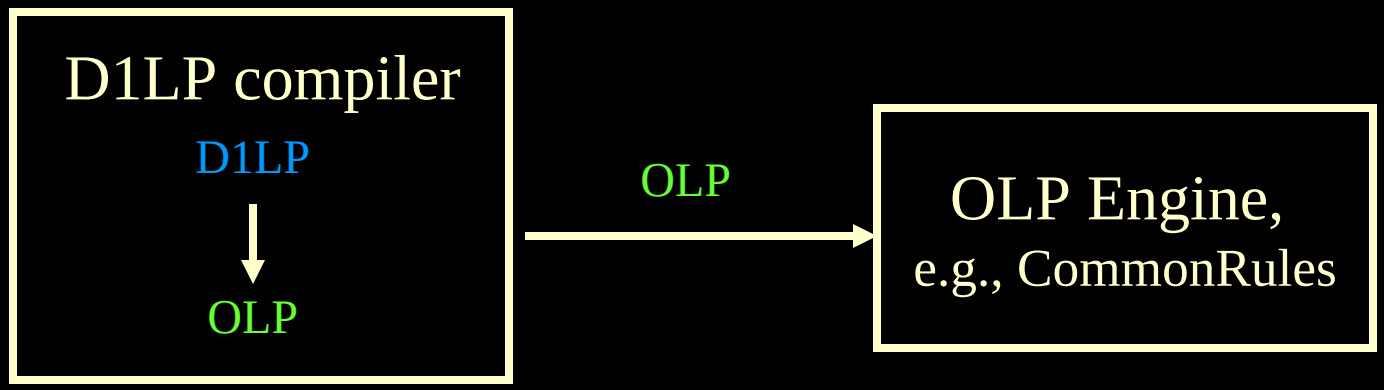
1. Trusting
   - *Alice* trusts *Bob* on something
   - Implication: if *Bob* says something, then *Alice* agrees
2. Entrusting
   - *Alice* allows *Bob* to act on *Alice*'s behalf
   - Implication: a request from *Bob* should be viewed as from *Alice*
3. Granting
   - *Alice* grants certain rights to *Bob*
   - Implication: if *Alice* has a certain right, then *Bob* should also have it

D1LP: Semantics - overview

- An authorizer has *policies* and receives *credentials*. Taken together, these form a rule-set *P*, a.k.a. a logic program.
- The declarative semantics of D1LP decides a unique set of statements that are true according to *P*, *i.e.*, the conclusions of *P*.
- The conclusion set is unique and **finitely computable**.
- Define semantics via a transform to ordinary logic programs:
  - Given a D1LP *P* , define a language *LO_P* that expresses definite logic programs. (definite = without negation-as-failure).
  - Given an model-theoretic interpretation *I* of *LO_P*, transform *P* into a ground definite logic program *O^I* in *LO_P*.

D1LP Compiler (Architecture)

- Java Implementation (part of CommonRules research prototype)



- Prolog Implementation:
  - The compiler is written in Prolog
  - The compiler dynamically asserts OLP rules into Prolog engine
  - Uses Prolog engine to do inference

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
  - Many believe it is an inhibitor to progress
  - *Discussion: What do you think?*

eXtensible Access Control 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