Semantic Web Services, Rules, and E-Contracting

Benjamin Grosof
MIT Sloan School of Management
Information Technologies group
http://ebusiness.mit.edu/bgrosof

(ITM = Information Technology & Management, a joint PhD program between Business School and Division of Engineering and Applied Sciences)
http://www.hbs.edu/doctoral/programs/itm/index.html

10/3/2003 Copyright 2002 by Benjamin Grosof. All Rights Reserved
Outline of Talk

• I. Overview: Semantic Web Services (SWS) and Rules
  – Concepts, Today’s Scene, Visions
  – Applications, incl. B2B
• II. E-Contracting via Rule-based SWS
  – SweetDeal Approach, Application Scenario
• III. Overall Requirements and Uses for Rule-based SWS
• IV. Research Directions
  – Theory, Applications, Technologies, Strategy
Next Generation Web

Semantic Web Services

Semantic Web techniques

Existing Web

Web Services techniques
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
Semantic Web: concept, approach, pieces

- Shared semantics when interchange data \therefore 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 \(\sim\) subset of First-Order Logic (FOL)
- **Rules** = if-then logical implications, facts \(\sim\) subsumes SQL DB’s
  - RuleML: “Rule Markup Language” emerging standard
    - Based on Logic Programs (LP) KR \(\sim\) extension of Horn FOL
Some Semantic Web Advantages for Biz

- Builds upon XML’s much greater capabilities (vs. HTML*) for **structured detailed descriptions** that can be processed **automatically**.
  - Eases application development effort for **assimilation of data in inter-enterprise interchange**

- **Knowledge-Based E-Markets -- where Agents Communicate**
  (Agent = knowledge-based application)
  - .::. potential to **revolutionize interactivity in Web marketplaces**: B2B, …

- **Reuse same knowledge for multiple purposes/tasks/app’s**
  - Exploit declarative KR; Schemas

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

10/3/2003   Copyright 2002 by Benjamin Grosof. All Rights Reserved
W3C Semantic Web “Stack”: Standardization Steps

Emerging Standards
pioneered in DARPA Agent Markup
Language (DAML) program:

- RuleML
- OWL

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

10/3/2003 Copyright 2002 by Benjamin Grosof. All Rights Reserved
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
  – Integrated knowledge
• (Semantic Web) Services: e.g., infrastructural
  – Knowledge/info/DB integration
  – Inferencing and translation
**SWS Language effort, on top of Current WS Standards Stack**

**“Wire” Protocols**
- W3C WS Choreography Group
- BPEL4WS (Microsoft, IBM, BEA)
- WSCL (HP)BPML (Most but Microsoft)
- WSCI (Sun, BEA, Yahoo, …)
- XLANG (Microsoft), WSFL (IBM), …

**Service Description**
- SOAP Blocks
- SOAP/XMLP
- XML
- HTTP/SMTMP
- TCP/IP
- SWS Language
  - Process
  - WSDL Extensions
  - WSDL
  - XML

**SWS Initiative (SWSI) -- automate Tasks of:**
- Discovery
- Invocation
- Interoperation
- Deal Negotiation
- Composition
- Monitoring
- Verification

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

• I. Overview: Semantic Web Services (SWS) and Rules
  – Concepts, Today’s Scene, Visions
  – Applications, incl. B2B
• II. E-Contracting via Rule-based SWS
  – SweetDeal Approach, Application Scenario
• III. Overall Requirements and Uses for Rule-based SWS
• IV. Research Directions
  – Theory, Applications, Technologies, Strategy
**B2B Tasks: Communication for Business Processes with Partners**

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

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

Coming soon to a world near you:...

- billions/trillions of agents (= k-b applications)
- ...with smarts: knowledge gathering, reasoning, economic optimization
- ...doing our bidding
  - but with some autonomy
- A 1st step: ability to communicate with sufficiently precise shared meaning... via the SEMANTIC WEB
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]:
Outline of Talk

• I. Overview: Semantic Web Services (SWS) and Rules
  – Concepts, Today’s Scene, Visions
  – Applications, incl. B2B
• II. E-Contracting via Rule-based SWS
  – SweetDeal Approach, Application Scenario
• III. Overall Requirements and Uses for Rule-based SWS
• IV. Research Directions
  – Theory, Applications, Technologies, Strategy
Looks Simple To Start... then Gets Interestingly Precise

A Vision/Approach of what Web & Agents enable

**SALES RECEIPT**

<table>
<thead>
<tr>
<th>Receipt ID # K46239...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed, Benjamin</td>
</tr>
</tbody>
</table>

Web info/knowledge “behind the curtain”

- ComfieCo.com 5way Chair Blue
- Operating Rules of MIT Sloan
- $140. VISA Europe

10/3/2003 Copyright 2002 by Benjamin Grosof. All Rights Reserved
Contracting 1-2-3

1. Find Contracting Opportunity
2. Negotiate Contract
3. Execute Contract Terms

DISCOVER   NEGOTIATE   EXECUTE

- Applies to any contracting, electronic or not.
- May iterate or interleave these steps.
- Boundaries not necessarily sharp.
What’s Doable Today in rule-based agent contracting, based on our approach to rule representation (“SweetDeal”)

• Communicate: with deep shared semantics
  – XML, inter-operable with same sanctioned inferences
  – ↔ heterogeneous rule systems / rule-based agents

• Execute contract provisions:
  – infer; ebiz actions; authorize; ...

• Modify easily: contingent provisions
  – default rules; modularity; exceptions, overriding

• Reason about the contract/proposal
  – hypotheticals, test, evaluate; tractably
  – (also need “solo” decision making/support by each agent)
Approach:

**Rule-based Contracts for E-commerce**

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

- SWEET = Semantic WEB Enabling Technology
  - software components, theory, approach
  - pilot application scenarios, incl. contracting (SweetDeal)
- Uses/contributes *emerging standards* for XML and knowledge representation:
  - RuleML (co-founder)
  - WebOnt ontologies (W3C)
- Uses *repositories* of business processes and contracts
  - MIT Process Handbook (Sloan IT)
  - legal/regulatory sources: law firms, ABA, CommonAccord, … *Suggestions welcome!!*
Contract Rules across Applications / Enterprises

Application 1, e.g., seller e-storefront

Business Logic

Rules

e.g., OPS5

“E-Business”

Contract Rules Interchange

Application 2, e.g., buyer shopbot agent

Business Logic

Rules

e.g., Prolog

“E-Commerce”

“E-Business”

Contracting parties integrate e-businesses via shared rules.

10/3/2003

Copyright 2002 by Benjamin Grosfe. All Rights Reserved
Examples of Contract Provisions
Well-Represented by Rules
in Automated Deal Making

• Product descriptions
  – Product catalogs: properties, conditional on other properties.
• Pricing dependent upon: delivery-date, quantity, group memberships, umbrella contract provisions
• Terms & conditions: refund/cancellation timelines/deposits, lateness/quality penalties, ordering lead time, shipping, creditworthiness, biz-partner qualification, service provisions
• Trust
  – Creditworthiness, authorization, required signatures
• Buyer Requirements (RFQ, RFP) wrt the above
• Seller Capabilities (Sourcing, Qualification) wrt the above
Contract Rules during Negotiation

Contracting parties NEGOTIATE via shared rules.

Buyer, e.g., manufacturer

Business Logic

Rules

e.g., OPS5

Seller, e.g., supplier of parts

Business Logic

Rules

e.g., Prolog

Interchange

Contract Rules Interchange

As part of XML documents

10/3/2003 Copyright 2002 by Benjamin Grosof. All Rights Reserved
Exchange of Rules Content during Negotiation: example

Buyer, e.g., manufacturer

Request For Quote

Quote

Purchase Order

Ack. Deal

Seller, e.g., supplier of parts
Exchange of *Rules Content* during Negotiation: example

- Buyer, e.g., manufacturer
- Seller, e.g., supplier of parts

- Req. For Proposal
- Proposal
- Counter-Proposal
- Final Offer
- Purchase Order
- Ack. Deal
Negotiation Example XML Document:
Proposal from supplierCo to manufCo

- `<negotiation_message>`
- `<message_header>`
  - `<proposal/>`
  - `<from> supplierCo </from>`
  - `<to> ManufCo </to>`
- `<rules_content>`
  - …[see next slide]
- `<rules_content>`
  - …
- `</negotiation_message>`

Example of similar message document format:
- FIPA Agent Communication Markup Language (draft industry standard).
Courteous LP Example: E-Contract Proposal from supplierCo to manufCo

• ...  
  <usualPrice>  price(per_unit, ?PO, $60) ←  
  • purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧  
  • quantity_ordered(?PO, ?Q) ∧ (?Q ≥ 5) ∧ (?Q ≤ 1000) ∧  
  • shipping_date(?PO, ?D) ∧ (?D ≥ 24Apr00) ∧ (?D ≤ 12May00).  
  • <volumeDiscount>  price(per_unit, ?PO, $51) ←  
  • purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧  
  • quantity_ordered(?PO, ?Q) ∧ (?Q ≥ 100) ∧ (?Q ≤ 1000) ∧  
  • shipping_date(?PO, ?D) ∧ (?D ≥ 28Apr00) ∧ (?D ≤ 12May00).  
  overrides(volumeDiscount, usualPrice).  
• ⊥ ← price(per_unit, ?PO, ?X) ∧ price(per_unit, ?PO, ?Y) GIVEN (?X ≠ ?Y).  
• ...
Negotiation Ex. Doc. Rules:

Counter-Proposal from manufCo to supplierCo

- ...<usualPrice> price(per_unit, ?PO, $60) ← ...
- <volumeDiscount> price(per_unit, ?PO, $51) ←
- purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧
- quantity_ordered( ?PO, ?Q) ∧ (?Q ≥ 5) ∧ (?Q ≤ 1000) ∧
- shipping_date(?PO, ?D) ∧ (?D ≥ 28Apr00) ∧ (?D ≤ 12May00) . overrides(volumeDiscount, usualPrice).
- ⊥ ← price(per_unit, ?PO, ?X) ∧ price(per_unit, ?PO, ?Y) GIVEN (?X ≠ ?Y).
- <aSpecialDeal> price(per_unit, ?PO, $48) ←
- purchaseOrder(?PO, supplierCo, manufCo) ∧
- quantity_ordered( ?PO, ?Q) ∧ (?Q ≥ 400) ∧ (?Q ≤ 1000) ∧
- shipping_date(?PO, ?D) ∧ (?D ≥ 02May00) ∧ (?D ≤ 12May00) . overrides(aSpecialDeal, volumeDiscount) .
- overrides(aSpecialDeal, usualPrice) .
Negotiation Example --

XML Encoding of Rules in RuleML

• `<rulebase>`
• `<imp>`
• `<_rlab>usualPrice</_rlab>`
• `<_head>`
• `<cslit>`
  • `<_opr><rel>price</rel></_opr>`
  • `<ind>per_unit</ind>`
• `<var>PO</var>`
• `<ind>$60</ind>`
  • `<./cslit>`
• `<./_head>`
• `<_body>`  … (see next page)  `<./_body>`
• `<./imp>`
• …
• `<./rulebase>`
Negotiation Example --

**XML Encoding of Rules in RuleML, Continued**

- `<_body>`
- `<andb>`
- `<fclit>`
  - `<_opr><rel>purchaseOrder</rel></_opr>`
  - `<var>PO</var>`
  - `<ind>supplierCo</ind>`
  - `<var>AnyBuyer</var>`
- `</fclit>`
- `<fclit>`
- `...`
- `</fclit>`
- `...`
- `</andb>`
- `</_body>`
Commercial Implementation & Piloting

- **IBM CommonRules**: AlphaWorks Java library
  - implements rule-based capabilities:
    - XML inter-operability; prioritized conflict handling

- **Rule Markup Language**: nascent industry standards effort
  - XML Knowledge Representation (KR) → make the Web be “Semantic”
  - KR: **Situated Courteous Logic Programs in XML**

- EECOMS industry consortium including Boeing, Baan, TRW, Vitria, IBM, universities, small companies
  - $29Million 1998-2000; 50% funded by NIST ATP
  - application piloted
    - contracting & negotiation; authorization & trust
Outline of Talk

• I. Overview: Semantic Web Services (SWS) and Rules
  – Concepts, Today’s Scene, Visions
  – Applications, incl. B2B

• II. E-Contracting via Rule-based SWS
  – SweetDeal Approach, Application Scenario

• III. Overall Requirements and Uses for Rule-based SWS

• IV. Research Directions
  – Theory, Applications, Technologies, Strategy
Flavors of Rules Commercially Most Important today in E-Business

• E.g., in OO app’s, DB’s, workflows.

• Relational databases, SQL: Views, queries, facts are all rules.
  • SQL99 even has recursive rules.

• Production rules (OPS5 heritage): e.g.,
  – Blaze, ILOG, Haley: rule-based Java/C++ objects.

• Event-Condition-Action rules (loose family), cf.:
  – business process automation / workflow tools.
  – active databases; publish-subscribe.

• Prolog. “logic programs” as a full programming language.

• (Lesser: other knowledge-based systems.)
Vision: Uses of Rules in E-Business

• Rules as an important aspect of coming world of Internet e-business: rule-based business policies & business processes, for B2B & B2C.
  – represent seller’s offerings of products & services, capabilities, bids; map offerings from multiple suppliers to common catalog.
  – represent buyer’s requests, interests, bids; → matchmaking.
  – represent sales help, customer help, procurement, authorization/trust, brokering, workflow.
  – high level of conceptual abstraction; easier for non-programmers to understand, specify, dynamically modify & merge.
  – executable but can treat as data, separate from code
    • potentially ubiquitous; already wide: e.g., SQL views, queries.
• Rules in communicating applications, e.g., embedded intelligent agents.
Criteria for Contract Rule Representation

1. **High-level:** Agents reach common understanding; contract is easily modifiable, communicatable, executable.
2. Inter-operate: heterogeneous commercially important rule systems.
3. Expressive power, convenience, natural-ness.
   
   ... but: computational **tractability**.
4. **Modularity** and locality in revision.
5. Declarative semantics.
   - essential feature in commercially important rule systems.
7. Prioritized conflict handling.
8. Ease of parsing.
9. Integration into Web-world software engineering.

**OLP**
**Courteous**
**XML**
**Situated**

10/3/2003 Copyright 2002 by Benjamin Grososf. All Rights Reserved
Also Currently Being Developed in the world today

- Delegations between agents
- XML Ontologies (Vocabularies)
  - knowledge representation: infer with definitional knowledge
  - specific domain/industry vocabularies
- DARPA Agent Markup Language: ontologies, rules
- Industry Standards:
  - Web, incl. Web services
  - Agents, Business Processes, Workflow
  - E-Commerce: ebXML, ...
  - Industry-Specific
  - Legal XML
- Law: Electronic Signatures, ...
- Reusable Contract doc’s on Web: CommonAccord, our work, ...
Rule-based Semantic Web Services

• Rules/LP in appropriate combination with DL as KR, for RSWS
  – DL good for categorizing: a service overall, its inputs, its outputs

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

• Rules to specify deals about services: cf. e-contracting.
Rule-based Semantic Web Services

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

• **Infrastructural**: rule system functionality as services:
  – e.g., inferencing, translation
Analysis:
High-Level Requirements for SWS

• Support Biz-Process Communication
  – E.g., B2B SCM, CRM
  – E.g., e-contracts, financial info, trust management.

• Support SWS Tasks above current WS layers:
  – Discovery/search, invocation, deal negotiation, selection, composition, execution, monitoring, verification
New Analysis:
Key Technical Requirements for SWS

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

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

• I. Overview: Semantic Web Services (SWS) and Rules
  – Concepts, Today’s Scene, Visions
  – Applications, incl. B2B
• II. E-Contracting via Rule-based SWS
  – SweetDeal Approach, Application Scenario
• III. Overall Requirements and Uses for Rule-based SWS
• IV. Research Directions
  – Theory, Applications, Technologies, Strategy
3 Areas of New Fundamental KR Theory that enable Key Technical Requirements for SWS

1. Description Logic Programs:
   KR to combine LP (RuleML) rules on top of DL (OWL) ontologies, with:
   - Power in inferencing (including for consistency)
   - Scaleability of inferencing

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

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

10/3/2003 Copyright 2002 by Benjamin Grosof. All Rights Reserved
Research Directions

• Requirements Analysis
• Fundamental KR theory, techniques, tools:
  – Courteous LP, Situated LP, Description LP
  – More: nonmon OO ontologies, multi-agent nonmon, equational ontologies, context mappings, …
• Web Services / Business Processes Knowledge Bases:
• Standards: Rules (RuleML/DAML), SWS (SWSI)
• Applications: e-contracting, finance, trust mgm., travel
• Fundamental theory for e-contracting
• Strategy wrt SWS uses, adoption, markets
Contributions to Early Standards Efforts: RuleML, SWSI

- **RuleML Initiative**
  - Co-Lead, Co-Founder
  - RuleML based largely on IBM CommonRules
  - Designed most key RuleML features
  - RuleML already has basic support for Description LP, Situated LP, Courteous LP

- **Active in SWSI, esp. on Rules**
  - Member of SWS Language committee
  - Co-chair Industrial Partners forum: several dozen companies
  - Technical challenge: representing service pre- / post-conditions, coherently on top of evolving messiness of WS process models (e.g., BPEL4WS)
SW Early Adoption Candidates: High-Level View

• “Death. Taxes. Integration.”

• Application/Info Integration:
  – Intra-enterprise
    • EAI, M&A; XML infrastructure trend
  – Inter-enterprise
    • E-Commerce: procurement, SCM
  – Combo
    • Business partners, extranet trend
SWS Adoption Roadmap: Strategy Considerations

- Expect see beginning in a lot of B2B interoperability or heterogeneous-info-integration intensive (e.g., finance, travel)
  - Actually, probably 1st intra-enterprise, e.g., EAI
- Reduce costs of communication in procurement, operations, customer service, supply chain ordering and logistics
  - increase speed, creates value, increases dynamism
  - macro effects create
    - stability sometimes (e.g., supply chain reactions due to lag; other negative feedbacks)
    - volatility sometimes (e.g., perhaps financial market swings)
  - increase flexibility, decrease lock-in
- Agility in business processes, supply chains
SW Early Adopters: Areas by Industry or Task

• Early SW techniques already in use:
  – e-contracting, supply chain incl. procurement
    • manufacturing, e.g. computer/electronics (RosettaNet), automotive (Covisint),
    • EECOMS pilot (Boeing, IBM, TRW, Baan)
  • office supplies (OBI)
  • retailing: shopbots and salesbots: comparisons, recommendations
  • extensive standards activity: Oasis ebXML, XML eContracts, UN UBL, EDI
SW Early Adopters: Areas by Industry or Task

• **Continued:** Early SW techniques already in use:
  – cyber goods:
    • financial services (rules; onto translation)
    • travel "agency", i.e.: tickets, packages (AI smarts for scheduling)
  – military intelligence (e.g., funded DAML)
Outline of Talk

I. Overview: Semantic Web Services (SWS) and Rules
   – Concepts, Today’s Scene, Visions
   – Applications, incl. B2B

II. E-Contracting via Rule-based SWS
    – SweetDeal Approach, Application Scenario

III. Overall Requirements and Uses for Rule-based SWS

IV. Research Directions
    – Theory, Applications, Technologies, Strategy
Acknowledgements

• SweetDeal e-contracting: student: Terrence Poon
• Situated Courteous Logic Programs: collaborator on implementation: Hoi Chan
• RuleML design: collaborators: Harold Boley, Said Tabet

• Support for the work was provided by DARPA Agent Markup Language program and Center for eBusiness @ MIT Vision Fund
**OWL: SW ontologies KR standard**

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

10/3/2003  Copyright 2002 by Benjamin Grosof. All Rights Reserved
Prioritized argumentation in an opposition-locale.

Conclusions from opposition-locales previous to this opposition-locale \{p_1, \ldots, p_k\}

(Each \(p_i\) is a ground classical literal. \(k \geq 2\).)

Run Rules for \(p_1, \ldots, p_k\)

Set of Candidates for \(p_1, \ldots, p_k\):
Team for \(p_1, \ldots, p_k\)

Prioritized Refutation

Set of Unrefuted Candidates for \(p_1, \ldots, p_k\):
Team for \(p_1, \ldots, p_k\)

Skepticism

Conclude Winning Side if any: at most one of \{p_1, \ldots, p_k\}
Courteous LP’s: Keys to Tractability

- Overall: mutex’s & conflict locales → keep tractability.
- LP’s: disallow contraposition (= \{¬a ← ., a ← b ∧ c.\} ⇒ (¬b ∨ ¬c)}) which requires disjunctive conclusions. “Directional”. Classical allows ⇒ NP-hard.
- Highly expressive prioritized rule representations (e.g., Prioritized Default Logic, Prioritized Circumscription) allow minimal conflict sets of arbitrary size ⇒ NP-hard overhead for conflict handling.
- Courteous conflict handling involves essentially only pairwise conflicts, i.e., minimal conflict sets of size 2. (Current work: possibly generalize to size k.)
  - Novelty: generalize to pairwise mutex’s beyond ⊥ ← p ∧ ¬p, e.g., partial-functional, thus avoid need for contraposition and larger conflict sets.
- Courteous conflict handling is local within an opposition locale: a set of rules whose heads oppose each other through mutex’s. Refutation and Skepticism are applied within each locale.
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
  - Currently morphing into a W3C activity
- “Agreement” here = agreement between invoker and provider of the service, described at knowledge level
- Overall: lots of proprietary jockeying and de-facto mode testing/pressuring of the open-consortial standards bodies (e.g., of W3C) “riding the tiger”
SWS Tasks at higher layers of WS stack

Automation of:

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

• Web service **invocation**  
  *Buy me “Harry Potter and the Philosopher’s Stone” at www.amazon.com*

• Web service **deals**, i.e., contracts, and their **negotiation**  
  *Propose a price with shipping details for used Dell laptops to Sue Smith.*

• Web service **selection**, **composition** and **interoperation**  
  *Make the travel arrangements for my WWW11 conference.*
SWS Tasks at higher layers of WS stack, continued

• Web service execution monitoring and problem resolution
  Has my book been shipped yet? … [NO!] Obtain recourse.

• Web service simulation and verification
  Suppose we had to cancel the order after 2 days?

• Web service executably specified at “knowledge level”
  The service is performed by running the contract ruleset through a rule engine.

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