E-Services Knowledge Management on the New Generation Web: End-to-End Contracting and Business Process Automation

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

Slides presented at Harvard University Econ/CS Seminar,
May 4, 2004
Host: Prof. David Parkes
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 of Talk

• Intro: Research on Semantic Web Services (SWS), its Business Uses
  – Rules, contracting, trust, policies
  – Integration, knowledge representation, standards

• Problem: Reusable Knowledge to Describe Services
  – Technique: knowledge representation to standardize on
  – Content investment: how to leverage legacy business process K

• New Technical Approach to represent OO Frameworks using SW
  – Courteous Inheritance: default rules increases reuse in ontologies

• New Strategy: go where the knowledge already is, then work outwards
  – Begin with MIT Process Handbook – open-source version in development
    • Example: process knowledge about selling
  – Future: Transformational wrappers around various legacy OO frameworks

• Roadmapping Market Evolution
  – Early adopters, creators, catalysts
  – Strategic players, forces

5/5/2004 Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.
Next Generation Web

Semantic Web Services

Semantic Web techniques
- Automated Knowledge Bases

Web Services techniques
- API’s on Web

XML

First Generation Web

Two interwoven aspects:
- Program: Web Services
- Data: Semantic Web
Some relevant websites

- http://www.oasis-open.org Oasis, e.g., its web services standards and ebXML
- http://www.xbrl.org XBRL eXtensible Business Reporting Language
Big Questions
about the New Generation Web

• What are the critical features/aspects of the new technology?

• What business problems does it help solve?

• What are the likely innovation evolution paths, and associated entrepreneurial opportunities?
Some Answers to: “Why does SWS Matter to Business?”


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-\textit{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
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.

5/5/2004 Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.
Our Overall SWS 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
Where are the Holdups?  
... and Challenges for Research

- KR & standards to integrate Rules with Ontologies
- KR, & later standards, to represent Services descriptions using Rules and Ontologies
- KR & strategy to leverage legacy content, e.g., OO service/process ontologies
  - A rich research area. We are doing much current work on that.

- Procedural process models aspect of SWS, as underlying foundation
  - Messy, many competing conceptual approaches
  - Realm of slow progress; much energy in WS standards efforts:
    - Oasis WSBPEL, W3C WS Choreography

- Hookup to negotiation strategies and valuation reasoning -> game theory:
  - In SweetDeal “Solo” reasoning factored as separate
  - Relevant: Work by Peyman Faratin of MIT

- Integrate risk management aspects, contract theory  -> game theory, other areas of economics
  - A rich research area. We are doing current work on this.
OPTIONAL BACKUP SLIDES
FOLLOW

• More about research directions
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
  – Scalability of inferencing

• 2. Hook rules (with ontologies) up to web services
  – Ex. web services: enterprise applications, databases
  – Rules use services, e.g., to query, message, act with side-effects
  – Rules constitute services executably, e.g., workflow-y business processes
  – Rules describe services non-executably, e.g., for discovery, deal negotiation
  – On top of web service process models, coherently despite evolving messiness
3 Areas of New Fundamental KR Theory that enable Key Technical Requirements for SWS

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

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

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

• More about research directions
OPTIONAL BACKUP SLIDES

- About what are Semantic Web, Web Services, and Semantic Web Services
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
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
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
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]

5/5/2004 Copyright 2002-2004 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
• About what are Semantic Web, Web Services, and Semantic Web Services
Outline of Talk

• Intro: Research on Semantic Web Services (SWS), its Business Uses
  – Rules, contracting, trust, policies
  – Integration, knowledge representation, standards
• Problem: Reusable Knowledge to Describe Services
  – Technique: knowledge representation to standardize on
  – Content investment: how to leverage legacy business process K
• New Technical Approach to represent OO Frameworks using SW
  – Courteous Inheritance: default rules increases reuse in ontologies
• New Strategy: go where the knowledge already is, then work outwards
  – Begin with MIT Process Handbook – open-source version in development
    • Example: process knowledge about selling
  – Future: Transformational wrappers around various legacy OO frameworks
• Roadmapping Market Evolution
  – Early adopters, creators, catalysts
  – Strategic players, forces
Problem: Reusable Knowledge to Describe Services

- Has two aspects:

  1. **Technical/technique problem**: what form of knowledge? I.e., what knowledge representation to standardize on?

  2. **Content investment problem**: how to leverage to accomplish the reuse of legacy business process knowledge?
Outline of Talk

• Intro: Research on Semantic Web Services (SWS), its Business Uses
  – Rules, contracting, trust, policies
  – Integration, knowledge representation, standards
• Problem: Reusable Knowledge to Describe Services
  – Technique: knowledge representation to standardize on
  – Content investment: how to leverage legacy business process K
• New Technical Approach to represent OO Frameworks using SW
  – Courteous Inheritance: default rules increases reuse in ontologies
• New Strategy: go where the knowledge already is, then work outwards
  – Begin with MIT Process Handbook – open-source version in development
    • Example: process knowledge about selling
  – Future: Transformational wrappers around various legacy OO frameworks
• Roadmapping Market Evolution
  – Early adopters, creators, catalysts
  – Strategic players, forces
Opportunity for MIT Process Handbook in SWS

• Need for Shared Web Services / Business Processes Knowledge Bases

• MIT Process Handbook as candidate nucleus for shared business process ontology for SWS
  – 5000+ business processes, + associated class/property concepts, as structured knowledge

• Related: use in particular for E-Contracting
  – Interoperable business objects, business processes
  – Also for policies (e.g., trust), 3rd-party services
Some Specializations of “Sell” in the MIT Process Handbook (PH)
OPTIONAL BACKUP SLIDES FOLLOW

• About SweetDeal’s use of Process Handbook ontology in rule-based e-contracts
Some Exceptions in the MIT Process Handbook

[Diagram showing exceptions and sub-exceptions related to system, social dilemmas, unclassified, missed opportunity, deadlock, any agent, agent, auctioneer, auction agent, matchmaker, host, communication, host problem, host slow, etc.]

5/5/2004

Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.
Some exception handlers in the MIT Process Handbook
payment(?R,base,?Payment) <-
http://xmlcontracting.org/sd.daml#result(co123,?R) AND
price(co123,?P) AND quantity(co123,?Q) AND
multiply(?P,?Q,?Payment) ;

<drm:imp>
  <drm:_head> <drm:atom>
    <drm:_opr><drm:rel>payment</drm:_opr><drm:rel> <drm:tup>
      <drm:var>R</drm:var> <drm:ind>base</drm:ind> <drm:var>Payment</drm:var>
    </drm:tup></drm:atom>  </drm:_head>
  <drm:_body>
    <drm:andb>
      <drm:atom> <drm:_opr>
        <drm:rel href="http://xmlcontracting.org/sd.daml#result"/>
        <drm:tup>
          <drm:ind>co123</drm:ind> <drm:var>Cust</drm:var>
        </drm:tup>  </drm:_opr>
      </drm:atom>  ...
    </drm:andb>  </drm:_body>  </drm:imp>

5/5/2004  Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.
Example Contract Proposal, Continued:

lateDeliveryPenalty exception handler module

lateDeliveryPenalty_module {
   // lateDeliveryPenalty is an instance of PenalizeForContingency
   // (and thus of AvoidException, ExceptionHandler, and Process)
   http://xmlcontracting.org/pr.daml#PenalizeForContingency(lateDeliveryPenalty) ;
   // lateDeliveryPenalty is intended to avoid exceptions of class
   // LateDelivery.
   http://xmlcontracting.org/sd.daml#avoidsException(lateDeliveryPenalty,
      http://xmlcontracting.org/pr.daml#LateDelivery);
   // penalty = - overdueDays * 200 ; (negative payment by buyer)
   <lateDeliveryPenalty_def> payment(?R, contingentPenalty, ?Penalty) <-
      http://xmlcontracting.org/sd.daml#specFor(?CO,?PI) AND
      http://xmlcontracting.org/pr.daml#hasException(?PI,?EI) AND
      http://xmlcontracting.org/pr.daml#isHandledBy(?EI,lateDeliveryPenalty) AND
      http://xmlcontracting.org/sd.daml#result(?CO,?R) AND
      http://xmlcontracting.org/sd.daml#exceptionOccurred(?R,?EI) AND
      shippingDate(?CO,?CODate) AND shippingDate(?R,?RDate) AND
      subtract(?RDate,?CODate,?OverdueDays) AND
      multiply(?OverdueDays, 200, ?Res1) AND multiply(?Res1, -1, ?Penalty) ;
   }
   <lateDeliveryPenaltyHandlesIt(e1)> // specify lateDeliveryPenalty as a handler for e1
      http://xmlcontracting.org/pr.daml#isHandledBy(e1,lateDeliveryPenalty);
Example Contract Proposal, Continued

- Buyer adds **rule modules** to the contract proposal to specify:
  - 1. detection of an exception
    - **LateDelivery** as a potential exception of the contract’s process
    - **detectLateDelivery** as exception handler: recognize occurrence
  - 2. avoidance of an exception (and perhaps also resolution of the exception)
    - **lateDeliveryPenalty** as exception handler: penalize per day

- Rule module = a nameable ruleset → a subset of overall rulebase
  - can be included directly and/or imported via link; nestable
    - similar to legal contracts’ “incorporation by reference”
  - an extension to RuleML; in spirit of “Webizing”
Example, Continued: Counter-Proposal

- Seller modifies the draft contract (it’s a *negotiation*!)
- **Simply adds** another rule module to specify:
  - *lateDeliveryRiskPayment* as exception handler
    - lump-sum in advance, based on average lateness
      - instead of proportional to actual lateness
    - higher-priority for that module than for the previous proposal, e.g., higher than *lateDeliveryPenalty*’s rule module
- Courteous LP’s prioritized conflict handling feature is used
- *NO change* to previous proposal’s rules needed!
  - similar to legal contracts’ accumulation of provisions
Example Counter-Proposal’s ruleset’s prioritized conflict handling

// priority specified via syntactically reserved “overrides” predicate

overrides(lateDeliveryRiskPaymentHandlesIt(e1),
    lateDeliveryPenaltyHandlesIt(e1) ) ;

// There is at most one avoid handler for a given exception instance.
// Consistency is enforced wrt this “mutex” integrity constraint.

MUTEX
http://xmlcontracting.org/pr.daml#isHandledBy(?EI, ?EHandler1) AND
http://xmlcontracting.org/pr.daml#isHandledBy(?EI, ?EHandler2)
GIVEN
http://xmlcontracting.org/sd.daml#AvoidException(?Ehandler1) AND
http://xmlcontracting.org/sd.daml#AvoidException(?Ehandler2) ;
Courteous feature: compileable, tractable

Courteous compiler

Courteous representation
(Sit.) Courteous LP.
mutex* priorities >

≡ equivalent semantically

ordinary ("vanilla")
(Sit.)OLP representation

Tractable compilation:
O(n^3), often linear

Tractable inference: e.g., worst-case when no ctor’s ("Datalog") & bounded \( v = |\text{var's per rule}| \)
is equivalent to OLP with \( v \rightarrow (v+2) \)

Preserves ontology.
Plus extra predicates for
- phases of prioritized argumentation (refutation, skepticism)
- classical negations

* classical negation too
Overview II: More New Contributions

1. Combine Situated Courteous Logic Programs (SCLP) case of RuleML with DAML+OIL (close predecessor of OWL); i.e., SCLP + Description Logic (DL)
   - rules "on top of" ontologies
   - show how and why to do as representational style (KR, syntax)
     - DAML+OIL class or property used as predicate in RuleML
     - heavily exploit feature of RuleML that predicate can be a URI
   - in progress: deeper semantics of the combination
     - more generally, 1st combo of nonmon RuleML / SCLP with DL
     - 1st combo of nonmon rules + DL (also Antoniou, independently)

2. Combine further with process descriptions

1st substantial practical e-business application domain scenario for 1., 2.

Point of convergence between Semantic Web and Web Services

1st: approach to automate MIT Process Handbook using: a) XML; b) powerful KR (but encoded only small fraction of its content so far!)
   - underline incapacity of OWL/DAML+OIL to represent default inheritance
END OF
OPTIONAL BACKUP SLIDES

• About SweetDeal’s use of Process Handbook ontology in rule-based e-contracts
OPTIONAL BACKUP SLIDES
FOLLOW

• About Rules for Semantic Web Services, esp. in e-contracting
**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.
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
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*
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, ...

5/5/2004 Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.
END OF
OPTIONAL BACKUP SLIDES

- About Rules for Semantic Web Services, esp. in e-contracting
Outline of Talk

• Intro: Research on Semantic Web Services (SWS), its Business Uses
  – Rules, contracting, trust, policies
  – Integration, knowledge representation, standards
• Problem: Reusable Knowledge to Describe Services
  – Technique: knowledge representation to standardize on
  – Content investment: how to leverage legacy business process K
• New Technical Approach to represent OO Frameworks using SW
  – **Courteous Inheritance**: default rules increases reuse in ontologies
• New Strategy: go where the knowledge already is, then work outwards
  – Begin with **MIT Process Handbook** – open-source version in development
    • Example: process knowledge about selling
  – Future: **Transformational wrappers** around various legacy OO frameworks
• Roadmapping Market Evolution
  – Early adopters, creators, catalysts
  – Strategic players, forces

5/5/2004  Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.

- Use SW KR and standards to represent Object-Oriented framework knowledge: class hierarchy, types, generalization-specialization, domain & range, properties/methods’ association with classes
- Surprise: use SW rule language not the main SW ontology language! I.e., use RuleML not OWL.
- Exploit RuleML’s nonmonotonic ability to represent prioritized default reasoning as kind of knowledge representation (KR)
New Technical Approach, continued

- Courteous Inheritance KR is built simply on top of the (Situated) Courteous Logic Programs KR of RuleML
  - A few dozen background axioms. Linear-size reformulation. Inferencing is tractable computationally.
- Particularly: represent PH's structured part
  - a scheme specific to PH’s flavor of OO
- PH becomes a SWS process ontology repository
  - to be combined, fed, used with/by other SWS
- Kill two birds with one stone:
  - form of K that facilitates leveraging of legacy process K content including PH, OO
New Technical Approach, continued more

• Example(s): selling, PO, price, shipping, delivery, payment, lateness.

• For details, see submitted paper “Beyond Monotonic Inheritance: Towards Semantic Web Process Ontologies” on webpage.
  – Example: selling process
Brief Tour of selling example in the paper.
Outline of Talk

• Intro: Research on Semantic Web Services (SWS), its Business Uses
  – Rules, contracting, trust, policies
  – Integration, knowledge representation, standards

• Problem: Reusable Knowledge to Describe Services
  – Technique: knowledge representation to standardize on
  – Content investment: how to leverage legacy business process K

• New Technical Approach to represent OO Frameworks using SW
  – Courteous Inheritance: default rules increases reuse in ontologies

• New Strategy: go where the knowledge already is, then work outwards
  – Begin with MIT Process Handbook – open-source version in development
    • Example: process knowledge about selling
  – Future: Transformational wrappers around various legacy OO frameworks

• Roadmapping Market Evolution
  – Early adopters, creators, catalysts
  – Strategic players, forces
Larger Approach: Transformation Wrappers for OO Frameworks

• New Strategy: go where the knowledge already is, then work outwards

• Future: Transformational wrappers around various legacy OO frameworks
  – C++
  – Java, C#
  – UML

• Can use XSLT, SW tools, and/or XQuery engines to implement the transformations, guided by SWS ontology standardization practices
Outline of Talk

• Intro: Research on Semantic Web Services (SWS), its Business Uses
  – Rules, contracting, trust, policies
  – Integration, knowledge representation, standards
• Problem: Reusable Knowledge to Describe Services
  – Technique: knowledge representation to standardize on
  – Content investment: how to leverage legacy business process K
• New Technical Approach to represent OO Frameworks using SW
  – Courteous Inheritance: default rules increases reuse in ontologies
• New Strategy: go where the knowledge already is, then work outwards
  – Begin with MIT Process Handbook – open-source version in development
    • Example: process knowledge about selling
  – Future: Transformational wrappers around various legacy OO frameworks
• Roadmapping Market Evolution
  – Early adopters, creators, catalysts
  – Strategic players, forces
Some relevant example companies

- Users: *Amazon, Fidelity, Boeing; UPS, GM, *Orbitz, eBay
- Vendors: IBM, Microsoft, Oracle, HP, BEA, SAP; Sun, *Compiere
- Standards-oriented organizations: *SWSI, *BPMI, *OPHI, UN CEFACT
Some More relevant websites

- [http://www.orbitz.com](http://www.orbitz.com) Orbitz, e.g., their vacation travel packages
- [http://www.compiere.org](http://www.compiere.org) Compiere open source ERP
Market Evolution: Discussion

Questions

- Existing and prospective early adopters
- Importance of open source content: seems to be an assumption/axiom for many people
- Prospective sources of open source content
Strategy Questions for Discussion

• ? Who/players: adopters, creators, catalysts?
• ? What forces/drivers for acceleration of adoption or investment, vs. inertia?
• ? Which additional interesting questions?
Yet More Discussion Questions: Early Adoption Application Prospects for SWS

- What business applications do you think are likely or interesting?
  - By vertical industry domain, e.g., health care or security
  - By task, e.g., authorization
  - By kind of shared information, e.g., patient records
  - By aspect of business relationships, e.g., provider network
- What do you think are entrepreneurial opportunity areas?
WRAP-UP: Outline of Talk

- Intro: Research on Semantic Web Services (SWS), its Business Uses
  - Rules, contracting, trust, policies
  - Integration, knowledge representation, standards
- Problem: Reusable Knowledge to Describe Services
  - Technique: knowledge representation to standardize on
  - Content investment: how to leverage legacy business process K
- New Technical Approach to represent OO Frameworks using SW
  - Courteous Inheritance: default rules increases reuse in ontologies
- New Strategy: go where the knowledge already is, then work outwards
  - Begin with MIT Process Handbook – open-source version in development
    - Example: process knowledge about selling
  - Future: Transformational wrappers around various legacy OO frameworks
- Roadmapping Market Evolution
  - Early adopters, creators, catalysts
  - Strategic players, forces
OPTIONAL BACKUP SLIDES FOLLOW

• About early adopter prospects in SWS
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
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)
END OF
OPTIONAL BACKUP SLIDES

• About early adopter prospects in SWS
• About Presenter’s SWS Research Agenda
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)
More about our SWS Technical Research Agenda

• Requirements Analysis (Biz → Tech)
  – New Application scenarios: e.g., SweetDeal e-contracting
  – Integrating rules, ontologies from many sources
  – Interoperability, power, consistency, scaleability

• New Fundamental Theory (Theory → Tech)
  – Description Logic Programs: bridging rules and ontologies
  – Situated Logic Programs: hooking rules to services
  – Courteous Logic Programs: prioritized conflict handling

• More:
  – Contributions to Early Standards Efforts: RuleML, SWSI
  – Piloting Early Adopter Areas: E-Contracts/SCM, Finance, Travel
  – Strategy Considerations and Implications
Analysis: High-Level Requirements for SWS

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

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

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

2. Hook rules (with ontologies) up to web services
   - Ex. web services: enterprise applications, databases
   - Rules use services, e.g., to query, message, act with side-effects
   - Rules constitute services executably, e.g., workflow-y business processes
   - Rules describe services non-executably, e.g., for discovery, deal negotiation
   - On top of web service process models, coherently despite evolving messiness
3 Areas of New Fundamental KR Theory that enable Key Technical Requirements for SWS

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

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

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

• About Semantic Web, Web Services
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
**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/SMTP
- 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)]

5/5/2004  Copyright 2002-2004 by Benjamin Grosof. All Rights Reserved.