Directions in Semantic Web Services

Benjamin Grosof

MIT Sloan School of Management

Information Technologies group

http://ebusiness.mit.edu/bgrosof

Slides presented at CISR Lunch Seminar, May 5, 2003

Center for Information Systems Research,

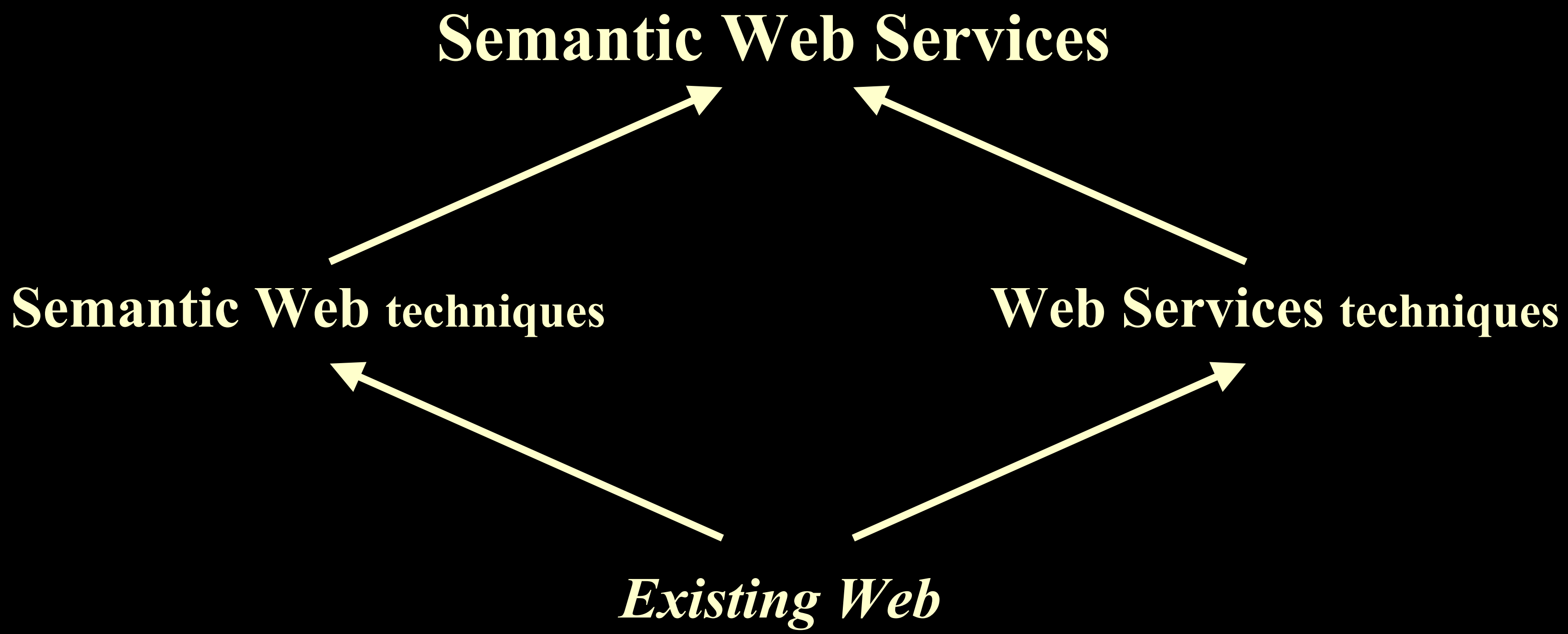
MIT Sloan School of Management

http://web.mit.edu/cisr/www

Outline of Talk

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

Next Generation Web



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 ∴ 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

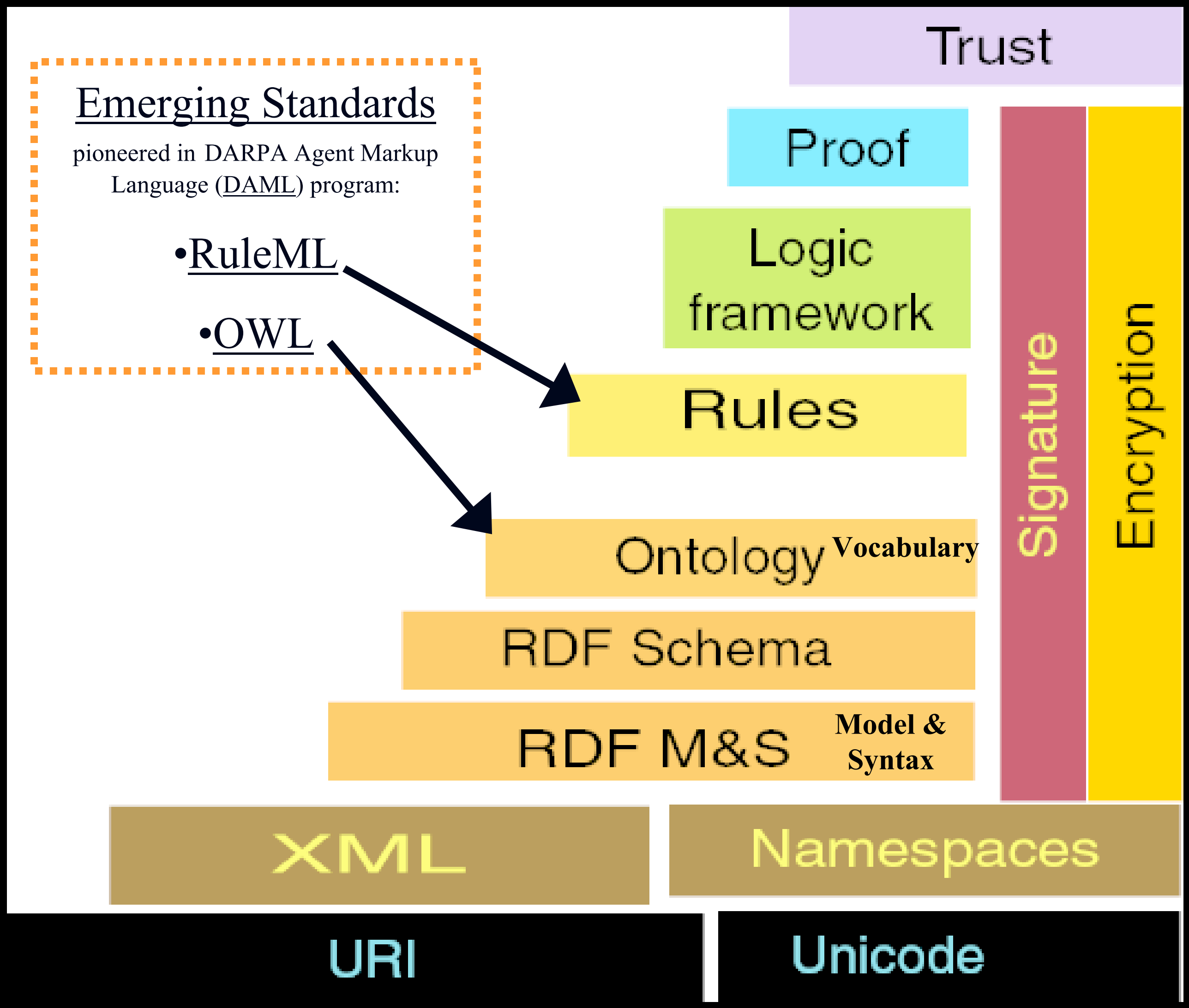
Some Semantic Web Advantages for Biz

- 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

W3C Semantic Web "Stack": Standardization Steps



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

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 | **Service Description**

W3C WS Choreography Group
BPEL4WS (Microsoft, IBM, BEA)
WSCL (HP)BPML (Most but Microsoft)
WSCI (Sun, BEA, Yahoo, …)
XLANG (Microsoft), WSFL (IBM), …

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

"Wire" Protocols	Service Description	
SOAP Blocks	SWS Language	
SOAP/XMLP	Process	
XML	WSDL Extensions	
HTTP/SMTP	WSDL	Registry (UDDI)
TCP/IP	XML	Inspection

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

Outline of Talk

- Introduction
  – Semantic Web Services (SWS)
- 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
- Conclusions

Outline of Talk

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

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

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.

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

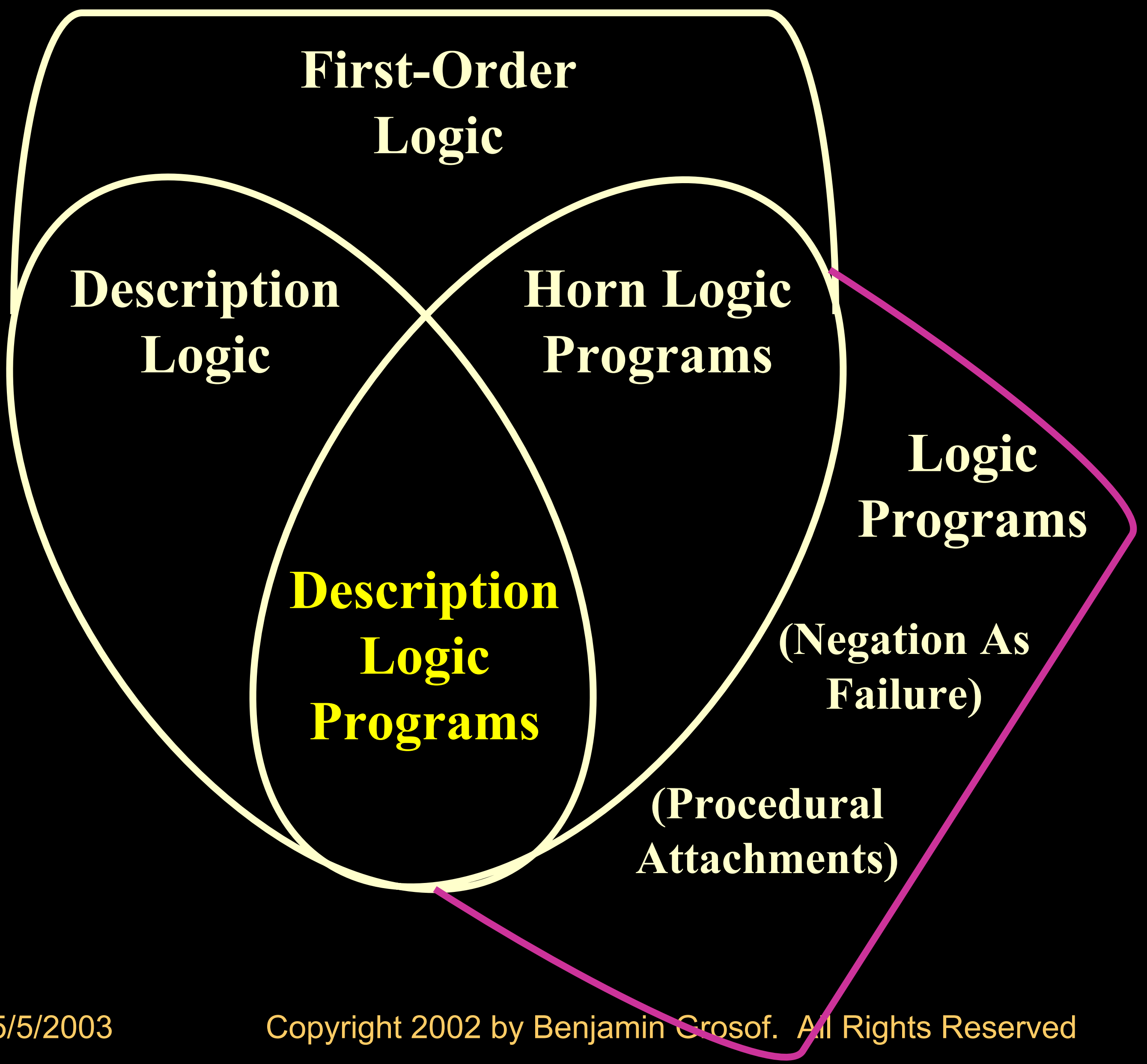
Outline of Talk

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

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

Venn Diagram: Expressive Overlaps among KR's



Overview of DLP KR Features

- DLP captures a complete subset of DL, containing RDFS plus more
- RDFS subset of DL permits the following statements:
  - Subclass, Domain, Range, Subproperty (also SameClass, SameProperty)
  - instance of class, instance of property
- DLP also completely captures following DL statements beyond RDFS:
  - Using the <u>Intersection</u> connective (conjunction) in class descriptions
  - Stating that a property (or inverse) P is <u>Transitive</u> or <u>Symmetric</u>.
  - (Some other stuff)
  - "OWL Feather"
- DLP can *largely but partially* capture: most other DL features.
  - Use skolemization, explicit equality, integrity constraints.
- Translation simpler to define from DL $\Rightarrow$ LP than DL $\Leftarrow$ LP.
- Bridge easily to Relational DBMS (SQL) – which is LP-based.
  - *Scaleability of LP/DB engines >> DL engines , as |instances|* $\uparrow$.

LP as a superset of DLP

- "Full" LP, including with non-monotonicity and procedural attachments, can thus be viewed as including an "ontology sub-language", namely the DLP subset of DL.

Technical Capabilities Enabled by DLP

- LP rules "on top of" DL ontologies.
  - E.g., LP imports DLP ontologies, with completeness & consistency
  - Consistency via completeness and use of Courteous LP
- Translation of LP rules to/from DL ontologies.
  - E.g., develop ontologies in LP (or rules in DL)
- Use of efficient LP rule/DBMS engines for DL fragment.
  - E.g., run larger-scale ontologies

- Translation of LP conclusions to DL.
- Translation of DL conclusions to LP.

- Facilitate rule-based mapping between ontologies / "contexts"

Outline of Talk

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

Heavy Reliance on Procedural Attachments in Currently Commercially Important Rule Families

- E.g., in OO app's, DB's, workflows.

- Relational databases, SQL: Built-in sensors, e.g., for arithmetic, comparisons, aggregations. Sometimes effectors: active rules / triggers.

- Production rules (OPS5 heritage): e.g., Jess
  - Pluggable (and built-in) sensors and effectors.
- Event-Condition-Action rules:
  - Pluggable (and built-in) sensors and effectors.

- Prolog: e.g., XSB.
  - Built-in sensors and effectors. More recent systems: more pluggability of the built-in attached procedures.

Situated LP's: Overview

- Point of departure: LP's are pure-belief representation, but most practical rule systems want to invoke external procedures.
- Situated LP 's feature a semantically-**clean** kind of **procedural attachments**. I.e., they hook beliefs to drive procedural API's outside the rule engine.
- Procedural attachments for **sensing** (queries) when testing an antecedent condition or for **effecting** (actions) upon concluding a consequent condition. Attached procedure is invoked when testing or concluding in inferencing.
- Sensor or effector **link** statement specifies an association from a predicate to a procedural call pattern, e.g., a method. A link is specified as part of the representation. I.e., a SLP is a conduct set that includes links as well as rules.

Situated LP's: Overview (cont.'d)

- phoneNumberOfPredicate ::s:: BoeingBluePagesClass.getPhoneMethod . *ex. sensor link*
- shouldSendPagePredicate ::e:: ATTPagerClass.goPageMethod . *ex. effector link*
- Sensor procedure may require some arguments to be ground, i.e., bound; in general it has a specified binding-signature.
- Enable dynamic or remote invocation/loading of the attached procedures (exploit Java goodness).
- Overall: cleanly separate out the procedural semantics as a declarative extension of the pure-belief declarative semantics. Easily separate chaining from action. (Declarative = Independent of inferencing control.)

Overview: Semantics of Situated Logic Programs

- Definitional: complete inferencing+action occurs during an "episode" – intuitively, run all the rules (including invoking effectors and sensors as go), then done.
- Effectors can be viewed as all operating/invoked after complete inferencing has been performed.
  - Independent of inferencing control.
    - But often intuitively less appropriate if only doing backward inferencing.
  - Separates pure-belief conclusion from action.

Overview: Semantics of Situated LP, continued

- Sensors can be viewed as accessing a virtual knowledge base (of facts). Their results simply augment the local set of facts. These can be saved (i.e., cached) during the episode.
  - Independent of inferencing control.
- The sensor attached procedure could be a remote powerful DB or KB system, a web service, or simply some humble procedure.
- Likewise, an effector attached procedure could be a remote web service, or some humble procedure. An interesting case for SW is when it performs updating of a DB or KB, e.g., "delivers an event".

Overview of Semantics of Situated LP, continued

- Conditions:
  - Effectors have only *side* effects: they do not affect operation of the (episode's) inferencing+action engine itself, nor change the (episode's) knowledge base.
  - Sensors are purely informational: they do not have side effects (i.e., any such can be ignored).
  - Timelessness of sensor and effector calls: their results are not dependent on when they are invoked, during a given inferencing episode.
  - "Sensor-safeness": Each rule ensures sufficient (variable) bindings are available to satisfy the binding signature of each sensor associated with any of its body literals – such bindings come from the other, non-sensor literals in the rule body. During overall "testing" of a rule body, sensors needing such bindings can be viewed as invoked after the other literals have been "tested".

SweetJess [Grosof, Gandhe, & Finin 2002]: First-of-a-kind Translation Mapping/Tool between LP and OPS5 Production Rules

- Requirement for rules interoperability:

  Bridge between multiple families of commercially important rule systems: SQL DB, Prolog, OPS5-heritage production rules, event-condition rules.
- Previously known: SQL DB and Prolog are LP.
- Theory and Tool Challenge: bring production rules and event-condition-action rules to the SW party
- Previously not known how to do even theoretically.
- Situated LP is the KR theory underpinning SweetJess, which:
  - Translates between RuleML and Jess production rules system
- SweetJess V1 implementation available free on Web

SweetJess: Translating an Effector Statement

```
<damlRuleML:effe>
  <damlRuleML:_opr>
   <damlRuleML:rel>giveDiscount</damlRuleML:rel>
  </damlRuleML:_opr>
  <damlRuleML:_aproc>
   <damlRuleML:jproc>
     <damlRuleML:meth>setCustomerDiscount</damlRuleML:meth>
     <damlRuleML:clas>orderMgmt.dynamicPricing</damlRuleML:clas>
     <damlRuleML:path>com.widgetsRUs.orderMgmt
       </damlRuleML:path>
   </damlRuleML:jproc>
  </damlRuleML:_aproc>

</damlRuleML:effe>
```

Associates with predicate P : an attached procedure A that is side-effectful.

- Drawing a conclusion about P triggers an action performed by A.

jproc = Java attached procedure.

meth, clas, path = its methodname, classname, pathname.

Equivalent in JESS: key portion is:

```
(defrule effect_giveDiscount_1
  (giveDiscount ?percentage ?customer)
  =>
  (effector setCustomerDiscount orderMgmt.dynamicPricing
                 (create$ ?percentage  ?customer) ) )
```

Example: Notifying a Customer when their Order is Modified

- See extended version of B. Grosof WITS-2001 conference paper
  - "Representing E-Business Rules on the Semantic Web: Situated Courteous Logic Programs in RuleML"
  - Available at http://ebusiness.mit.edu/bgrosof

Outline of Talk

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

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!

Courteous LP's: the What

- Updating/merging of rule sets: is crucial, often generates conflict.
- Courteous LP's feature prioritized handling of conflicts.
- Specify scope of conflict via a set of *pairwise* mutual exclusion constraints.
  - E.g., $\perp \leftarrow$ discount(?product,5%) $\wedge$ discount(?product,10%) .
  - E.g., $\perp \leftarrow$ loyalCustomer(?c,?s) $\wedge$ premiereCustomer(?c,?s) .
  - Permit classical-negation of atoms: $\neg$p means p has truth value *false*
    - implicitly, $\perp \leftarrow p \wedge \neg p$ for every atom p.
- **Priorities** between rules: partially-ordered.
  - Represent priorities via reserved predicate that compares rule labels:
    - overrides(rule1,rule2) means rule1 is higher-priority than rule2.
    - Each rule optionally has a rule label whose form is a functional term.
    - overrides can be reasoned about, just like any other predicate.

Priorities are available and useful

- Priority information is naturally available and useful. E.g.,
  - recency: higher priority for more recent updates.
  - specificity: higher priority for more specific cases (e.g., exceptional cases, sub-cases, inheritance).
  - authority: higher priority for more authoritative sources (e.g., legal regulations, organizational imperatives).
  - reliability: higher priority for more reliable sources (e.g., security certificates, via-delegation, assumptions, observational data).
  - closed world: lowest priority for catch-cases.
- Useful to infer prioritization
  - From knowledge about sources, incl. meta-data about web sources.
- Many practical rule systems employ priorities of some kind, often implicit, e.g.,
  - rule sequencing in Prolog and production rules.
    - courteous subsumes this as special case (totally-ordered priorities), plus enables: merging, more flexible & principled treatment.

Courteous LP's: Advantages

- Facilitate updating and merging, modularity and locality in specification.
- Expressive: classical negation, mutual exclusions, partially-ordered prioritization, reasoning to infer prioritization.
- Guarantee consistent, unique set of conclusions.
  - **Mutual exclusion is enforced**. E.g., never conclude discount is both 5% and that it is 10%, nor conclude both p and ¬p.
- Efficient: low computational overhead beyond ordinary LP's.
  - Tractable given reasonable restrictions (Datalog, bound v on #var's/rule):
    - extra cost is equivalent to increasing v to (v+2) in ordinary LP's.
  - By contrast, more expressive prioritized rule representations (e.g., Prioritized Default Logic) add NP-hard overhead.
- Modular software engineering: via courteous compiler: CLP → OLP.
  - A radical innovation. Add-on to variety of OLP rule systems. O(n^3).

Outline of Talk

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

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
  - Forming Industrial Advisory Board: 100 target 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 1^{st} 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, lower 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

- Introduction
  - Semantic Web Services (SWS)
- 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
- Conclusions

Acknowledgements

- Description Logic Programs: collaborators: Ian Horrocks, Stefan Decker, and student Raphael Volz
- SweetDeal e-contracting: student: Terrence Poon
- Situated Logic Programs: collaborator on implementation: Hoi Chan
- SweetJess collaborators on implementation: Tim Finin, student Mahesh Gandhe
- 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

OPTIONAL SLIDES FOLLOW

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) langauges 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.

Hybrid DL+LP Task Scenarios/Use-Cases

- 1. Service descriptions combining LP rules and DL ontologies

- 2. Rules for knowledge translation: e.g.,
  - translating/merging ontologies (or rules)

Overview: Semantics of Situated LP, Continued

- Generalizations possible:
  - permit multiple sensors or effectors per predicate.
  - sense functions (or terms) not just predicates.
  - permit sensor priority – i.e, specify the prioritization of the facts that result from a particular sensor .
  - associate sensing with atoms/literals (or terms), but this is reducible to sensing predicates (or functions) – by rewriting of the rules.
- Challenge: error handling info returned from attached procedures

Prioritized argumentation in an opposition-locale.

Conclusions from opposition-locales previous to this opposition-locale {p1,...,pk}

(Each pi is a ground classical literal. k ≥ 2.)

Run Rules for p1,...,pk

Set of Candidates for p1,...,pk:
Team for p1, ..., Team for pk

Prioritized Refutation

Set of Unrefuted Candidates for p1,...,pk:
Team for p1, ..., Team for pk

Skepticism

Conclude Winning Side if any: at most one of {p1,...,pk}

Courteous LP's: Keys to Tractability

- Overall: mutex's & conflict locales $\rightarrow$ keep tractability.
- LP's: disallow disjunctive conclusions, essentially. Classical allows $\Rightarrow$ NP-hard.
- LP's: disallow contraposition (= $\{\neg a \leftarrow .\, , a \leftarrow b \wedge c.\} \Rightarrow (\neg b \vee \neg c)\}$) which requires disjunctive conclusions. "Directional". Classical allows $\Rightarrow$ NP-hard.
- Highly expressive prioritized rule representations (e.g., Prioritized Default Logic, Prioritized Circumscription) allow minimal conflict sets of arbitrary size $\Rightarrow$ 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 $\perp \leftarrow p \wedge \neg p$, 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.

The Semantic Web

The 1st generation, the Internet, enabled disparate machines to exchange data.

- The 2nd generation, the World Wide Web, enabled new applications on top of the growing Internet, making enormous amounts of information available, in <u>human-readable</u> form, and allowing a revolution in new applications, environments, and <u>B2C</u> e-commerce.

- The next generation of the net is an “agent-enabled” resource (the “Semantic Web”) which makes a huge amount of information available in <u>machine-readable</u> form creating a revolution in new applications, environments, and <u>B2B</u> e-commerce.

…by enabling “agent” communication at a Web-wide scale.

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*

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.

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

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)]