# Extending the SweetDeal Approach for E-Procurement using SweetRules and RuleML

Sumit Bhansali and Benjamin Grosof

MIT Sloan School of Management

Information Technologies group

http://web.mit.edu/people/bhansali http://ebusiness.mit.edu/bgrosof

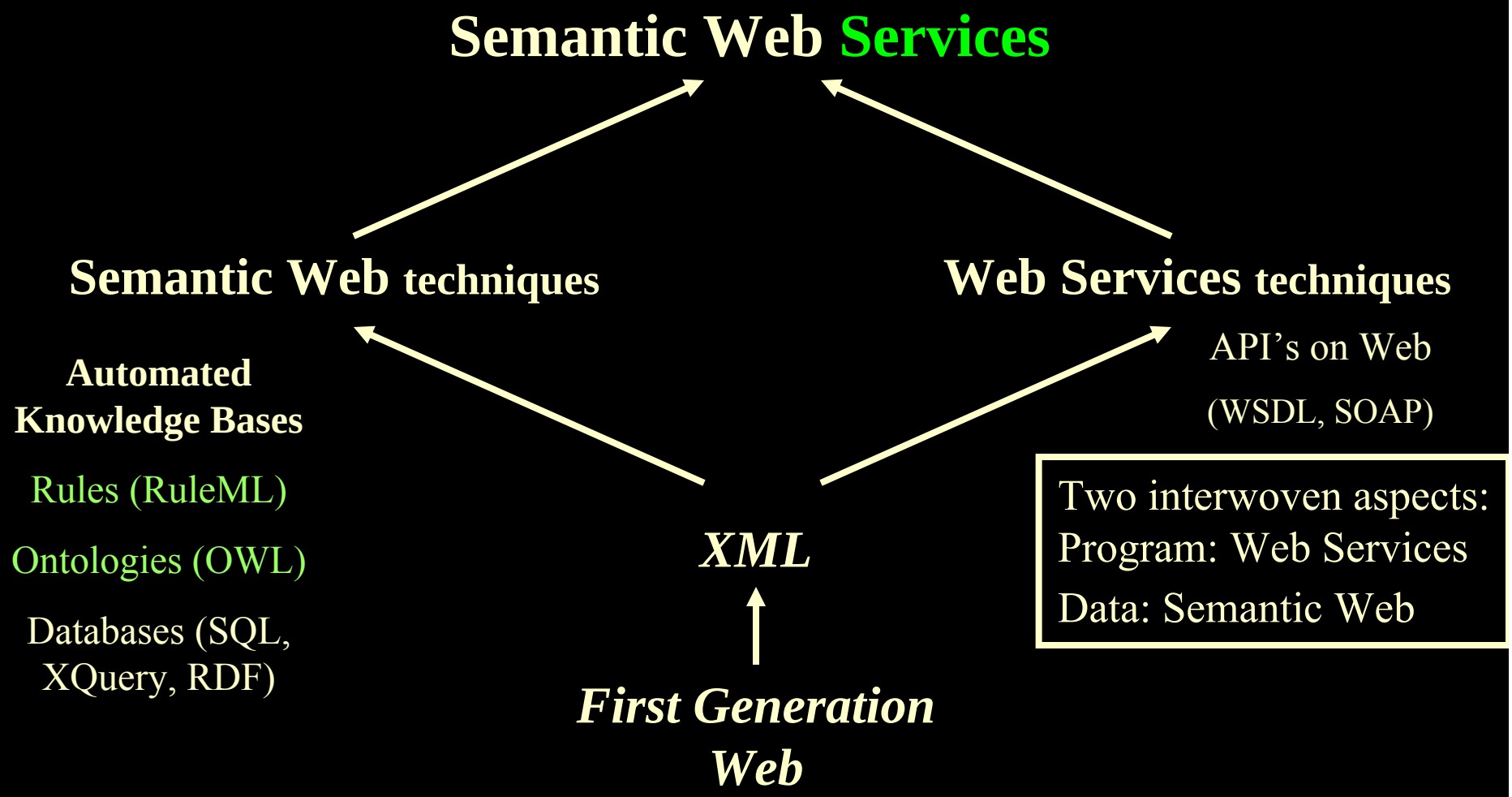
*Paper Presentation (30-minutes) at International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-2005), Galway, Ireland, Nov. 10-12, 2005* *http://2005.ruleml.org*

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work

# *Next Generation Web*



# *Our Research Aspects/Questions about the Semantic Web*

- Core technologies: Requirements, concepts, theory, algorithms, standards?
  - Rules in combination with ontologies; probabilistic, decision-/game-theoretic

- Business applications and implications: concepts, requirements analysis, techniques, scenarios, prototypes; strategies, business models, market-level evolution?
  - End-to-end e-contracting, finance, trust; …

# *Some Answers to: "Why does SW Matter to Business?"*

- 1. "Death. Taxes. Integration." - They're always with us.
- 2. "Business processes require communication between organizations / applications." - Data and programs cross org./app. boundaries, both intra- and inter- enterprise.
- 3. "It's the *automated knowledge* economy, stupid!" - The world is moving towards a knowledge economy. And it's moving towards deeper and broader automation of business processes. The first step is automating the use of structured knowledge.
  - Theme: *reuse* of knowledge across multiple tasks/app's/org's

# *Strategic Business Foci in our SW Research*

- Knowledge-based Services Engineering: intra- and inter- enterprise
- Target "killer app" known for 30 years: do better job of EDI
- Challenges:
  - Ease of development, deployment ↑
  - Reuse of knowledge ↑
  - ⇒ life cycle costs ↓ , agility ↑
- Starting with: Policies
  - Using recent theory breakthroughs in semantic rules
  - E.g., for end-to-end contracting and authorization (incl. security)
- Starting with: EAI as well as B2B

# *SW Rules: Use Cases from our research*

- Contracts/negotiation, advertising/discovery
  - E-procurement, E-selling
  - Pricing, terms & conditions, supplier qualification, …
- Monitoring:
  - Exception handling, e.g., of contract violations
    - Late delivery, refunds, cancellation, notifications
  - Notifications, personal messaging, and other workflow
- Trust Policies: authorization, confidentiality & privacy, security, access control
  - E.g., financial services, health care
    - *Extensive analysis of business case/value*
- Semantic mediation: rule-based ontology translation, context-based information integration

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work

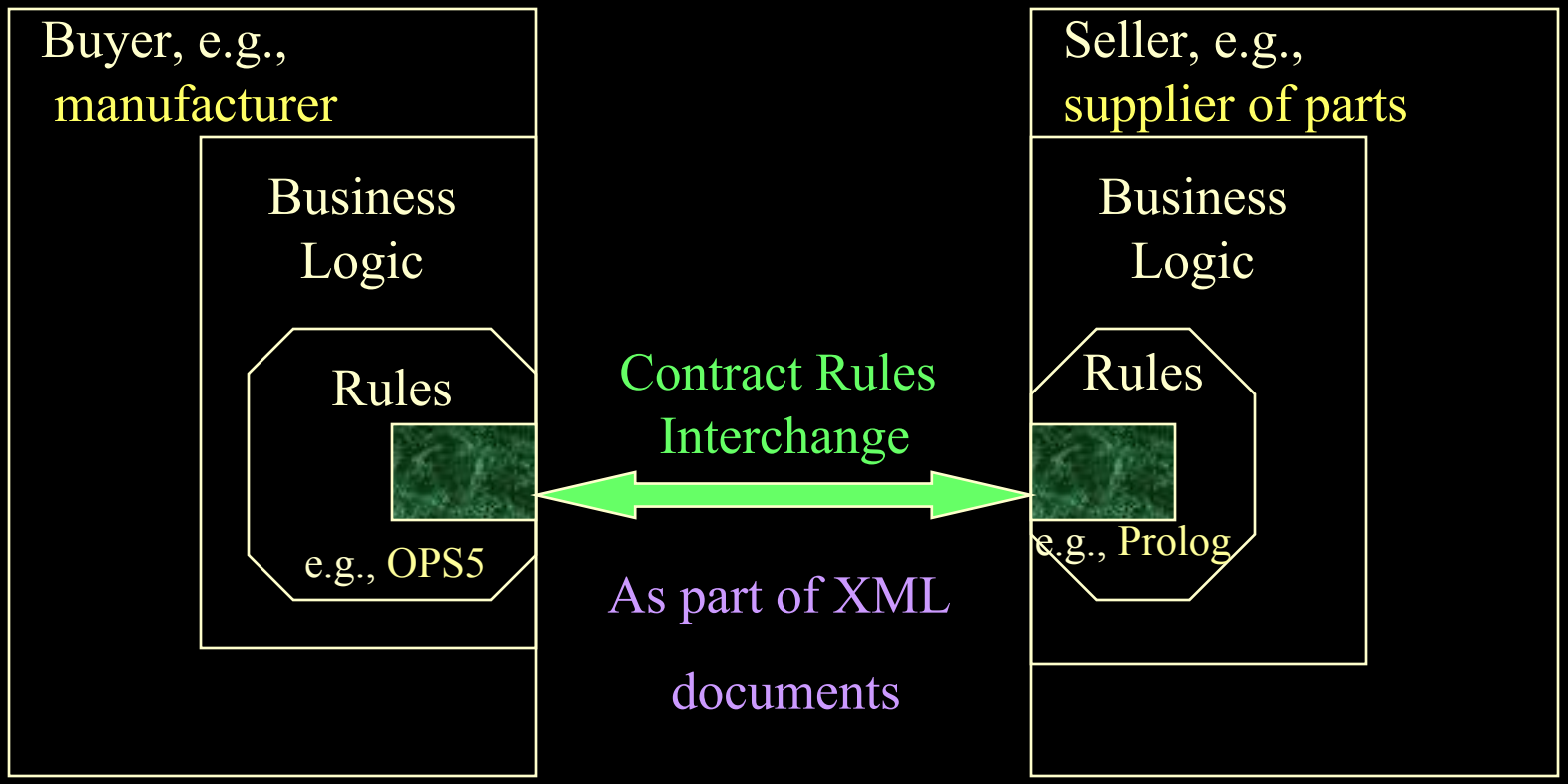
# *End-to-End E-Contracting Tasks*

- Discovery, advertising, matchmaking
  - Search, sourcing, qualification/credit checking
- Negotiation, bargaining, auctions, selection, forming agreements, committing
  - Hypothetical reasoning, what-if'ing, valuation
- Performance/execution of agreement
  - Delivery, payment, shipping, receiving, notification
- Problem Resolution, Monitoring
  - Exception handling

# *SweetDeal 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.

# *Contract Rules during Negotiation*

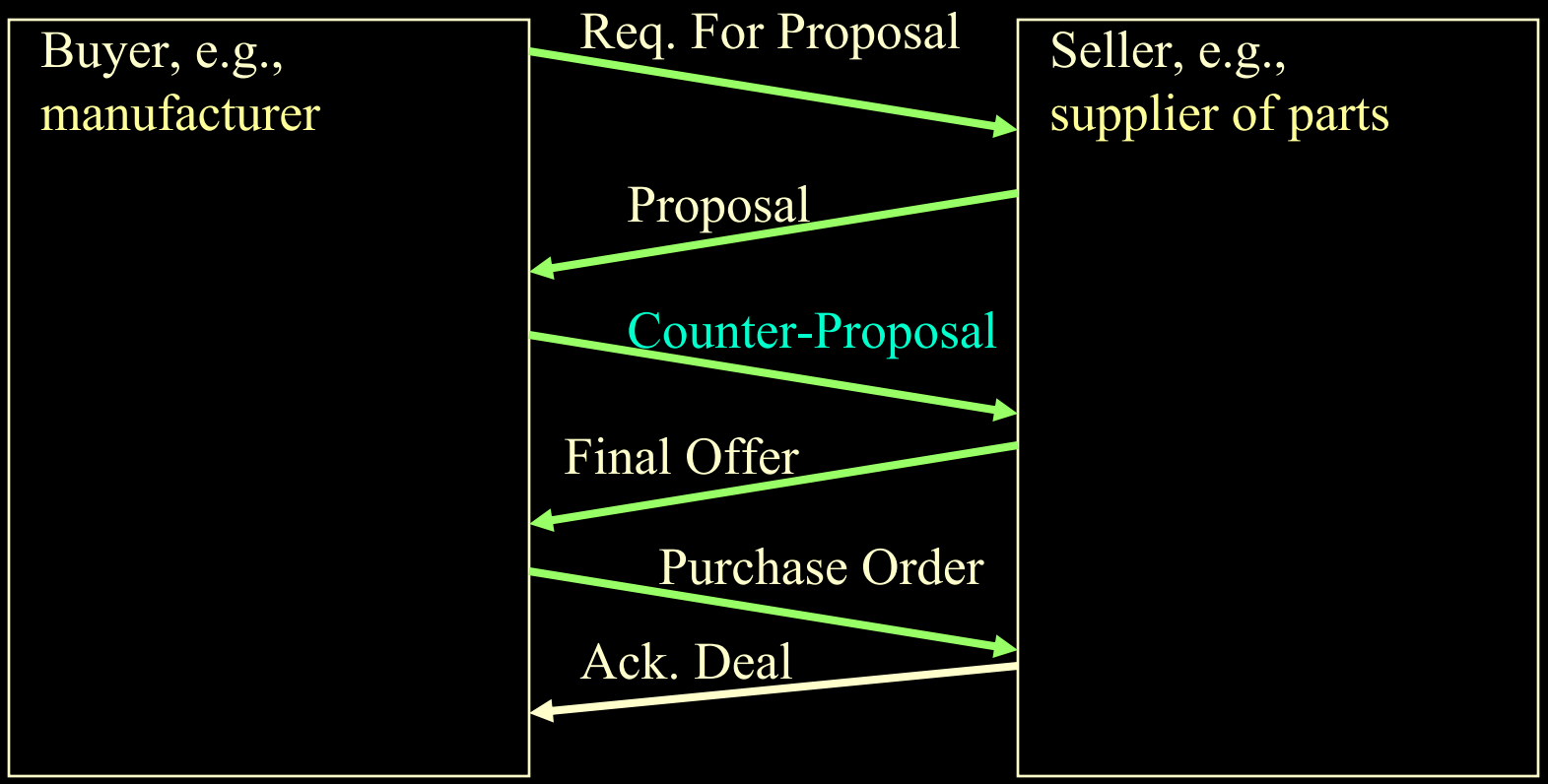



*Contracting parties NEGOTIATE via shared rules.*

# *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*

# *Exchange of Rules Content during Negotiation: example*



# *Example: E-Contract Proposal from supplierCo to manufCo*

- …
- {usualPrice} price(per_unit, ?PO, $60) ←
- purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧
- quantity_ordered( ?PO, ?Q) ∧ (?Q ≥ 5) ∧ (?Q ≤ 1000) ∧
- shipping_date(?PO, ?D) ∧ (?D ≥ 24Apr00) ∧ (?D ≤ 12May00).
- {volumeDiscount} price(per_unit, ?PO, $51) ←
- purchaseOrder(?PO, supplierCo, ?AnyBuyer) ∧
- quantity_ordered( ?PO, ?Q) ∧ (?Q ≥ 100) ∧ (?Q ≤ 1000) ∧
- shipping_date(?PO, ?D) ∧ (?D ≥ 28Apr00) ∧ (?D ≤ 12May00) .

overrides(volumeDiscount , usualPrice) .

- ⊥ ← price(per_unit, ?PO, ?X) ∧ price(per_unit, ?PO, ?Y) GIVEN (?X ≠ ?Y).
- ...

# *Negotiation Ex. Doc. Rules: Counter-Proposal* *from manufCo to supplierCo*

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

**Simply *added* rules!**

*Negotiation Example --*

# *XML Encoding of Rules in RuleML*

```
<rulebase>
 <imp>
  <rlab>usualPrice</_rlab>
  <head>
   <atom>
    <opr><rel>price</rel></_opr>
    <ind>per_unit</ind>
    <var>PO</var>
    <ind>$60</ind>
   </atom>
  </head>
  <body>   …  (see next page)   </_body>
 </imp>
…
</rulebase>
```

# *SweetDeal V2 Demo Outline*

- SweetDeal E-Contracting Application using SweetRules (supply chain)
  - SCLP RuleML that includes OWL ontologies
  - Contract proposals/final-agreements are SCLP RuleML rulebases that reference/include OWL ontologies
  - Humans edit & communicate, supported by automated agents
  - Proposal evaluation supported by inferencing
  - Agreed business process is executable via inferencing+action

# *What Can Be Done with the Rules in contracting, & negotiation, based on our SweetDeal approach to rule representation*

- Communicate: with deep shared semantics
  - via RuleML, inter-operable with same sanctioned inferences
  - ⇔ heterogeneous rule/DB systems / rule-based applications ("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)*

# *RECAP FOLLOWS*

# *Overview of SweetDeal Approach I*

SweetDeal approach: [EC-99, WWW-2003, IJEC 2004]; part of [SWSF 2005]

- Represent (parts of) e-contracts via webized semantic rules. Communicated.
- Contracts are partial or complete, proposed or final.

- Situated Courteous Logic Programs, in Webized syntax (RuleML) as representation. (Declarative. Similar
  - Prioritized defaults, with negation.
    - Modular modification. Default OO inheritance.
  - Procedural attachments for actions/effecting and queries/sensing. Execute business processes.
- With restricted OWL ontologies. The original use case & design pattern for Description Logic Programs (DLP).

# *Overview of SweetDeal Approach II*

- Handles "end-to-end" e-contracting tasks, i.e., knowledge-based services tasks:
  - advertising & discovery, negotiation & selection, agreement, authorization, monitoring & exception handling, modification & renegotiation, execution/enactment & business process automation.

- Many application scenarios, incl. of all these tasks:
  - pricing & discounting, refunds & customer service, ordering and lead time, late delivery, supply chain / B2B, retail / B2C, auctions, credit approval

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work

# *New Extensions to SweetDeal I*

- New particular procurement application scenario
  - Utilizes and exercises several advanced expressive features of the SweetDeal approach
    - Merge restricted DL ontologies: import OWL-DLP (DLP-fusion)
    - Effectors to perform business actions
    - Merge default-inheritance OO ontologies: as Courteous rules
  - Prototype, built using SweetRules V2.1 [demo'd 1st at DAML Winter 2004 PI Meeting]
    - Utilizes and exercises several new features of SweetRules
      - KB merging, e.g., OWL-DLP ontologies + Courteous policy RuleML rules.
      - Situated effecting incl. WSDL actions.

# *Overview of Procurement Scenario*

- B2B: Purchase of bunch of computers
  - Rebates; Financing Options (new aspects)
  - Pricing options, delivery details, …
- Buyer and sellers exchange contract proposals of increasing completeness
- Each does SCLP inferencing to evaluate implications of a given proposal, including relative to private info/criteria, and to trigger messaging sending/response

# *New Extensions to SweetDeal II*

- New expressive features and design understanding
  - Build on top of SW Rules tools: SweetRules
  - Merge heterogeneous KB's
  - Expressive Reasoning: Courteous, effecting, SCLP+DL
  - Reasoning & translation interoperably on/across heterogeneous rule-based systems/applications
  - … with strong semantics
  - Communication interaction protocol aspects represented using rules with effectors, e.g., to trigger contracting messages
  - Message contains a KB consisting of *queries* + rules
  - *Fact-queries* as expressive feature, natural & convenient
    - Experimental extension of RuleML markup

# *New Extensions to SweetDeal III*

- Initial steps of analysis/design of how fit as "letters content" within important e-business communication standards/platforms
  - RosettaNet, ebXML.
  - Fits nicely within their messages = "envelopes".

# *RULE/ONT EXAMPLES*

- Walk thru of some example SCLP RuleML and OWL-DLP contract knowledge from the RuleML-2005 proceedings paper

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work

# *Overview of SweetRules V2.1:*
## *Tools for Semantic Web Rules and Ontologies, including Translation, Inferencing, Analysis, and Authoring*

*http://sweetrules.projects.semwebcentral.org*

*Presentation (15-min.) by Benjamin Grosof* and Mike Dean***
*Tools by multi-institutional team (MIT, UMBC, BBN, …)*

*Invited Poster at RuleML-2005 (International Conference on Rules and Rule Markup Languages for the Semantic Web) http://2005.ruleml.org*
*held Galway, Ireland, Nov. 10-12, 2005.*
**MIT Sloan School of Management, http://ebusiness.mit.edu/bgrosof*
***BBN Technologies, http://www.daml.org/people/mdean*

# *SweetRules V2.1 Overview*

Key Ideas: **http://sweetrules.projects.semwebcentral.org**

- Unite the commercially most important kinds of rule and ontology languages via a a new, common knowledge representation (SCLP) in a new standardized syntax (RuleML), including to cope with *heterogeneity* and resolve contradictory *conflicts*.
  - Capture most of the useful expressiveness, interoperably and scalably.
- Provide an open source tool platform to combine a large *distributed* set of rule and ontology knowledge bases that each are *active:* each has a different *associated engine* for reasoning capabilities (inferencing, authoring, and/or translation ).
- Based on recent fundamental KR theory advances, esp. Situated Courteous Logic Programs (SCLP) and Description Logic Programs.
  - Including semantics-preserving translations between different rule languages/systems/families, e.g., Situated LP ↔ production rules

Application Areas (prototyped scenarios):

- Policies and authorizations; process monitoring; contracting, supply chain management; retailing, customer relationship management; business process automation and e-services; financial reporting and information; etc.

# *SweetRules Concept and Architecture*

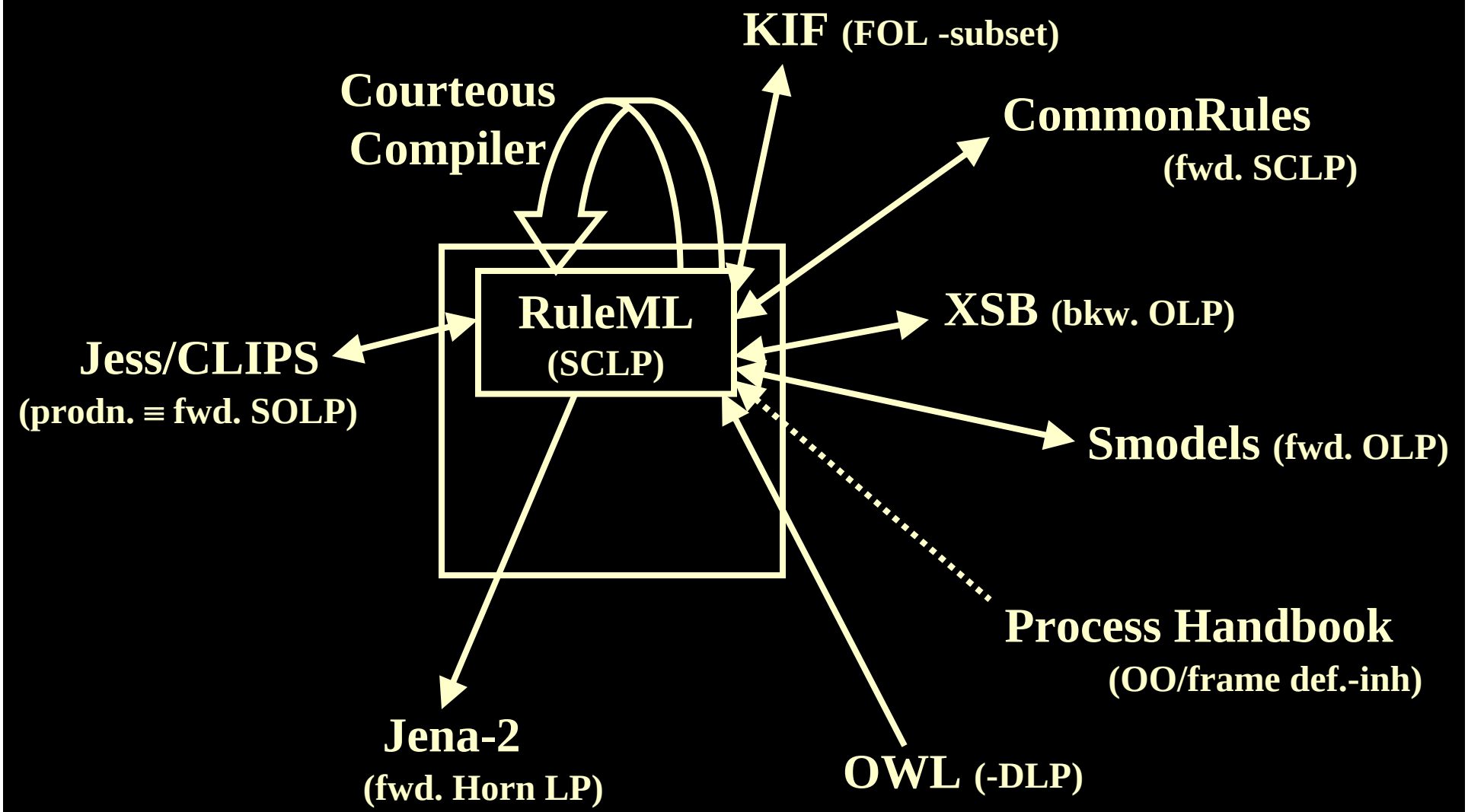
- Concept and Architecture: Tools suite for Rules and RuleML
  - Translation and interoperability between heterogeneous rule systems (forward- and backward-chaining) and their rule languages/representations
  - Inferencing including via translation between rule systems
  - Authoring, Analysis, and testing of rulebases
  - Open, lightweight, extensible, pluggable architecture overall

  - Merge knowledge bases
    - Combine rules with ontologies, incl. OWL
  - SWRL rules as special case of RuleML
  - Focus on kinds of rule systems that are commercially important

# *SweetRules V2.0+ Fundamental KR Today*

- Fundamental KR: Situated Courteous Logic Programs (SCLP)
  - Horn
  - + Negation-As-Failure (NAF) = Ordinary LP
  - + Courteous prioritized conflict handling
    - overrides relation on rule labels, classical negation, mutex integrity constraints
  - + Situated sensing & effecting
    - Invoke external procedural attachments
    - Sensing = tests/queries; e.g., built-ins
    - Effecting = side-effectful actions, triggered by conclusions

# *SweetRules V2.1 Translators Graph*

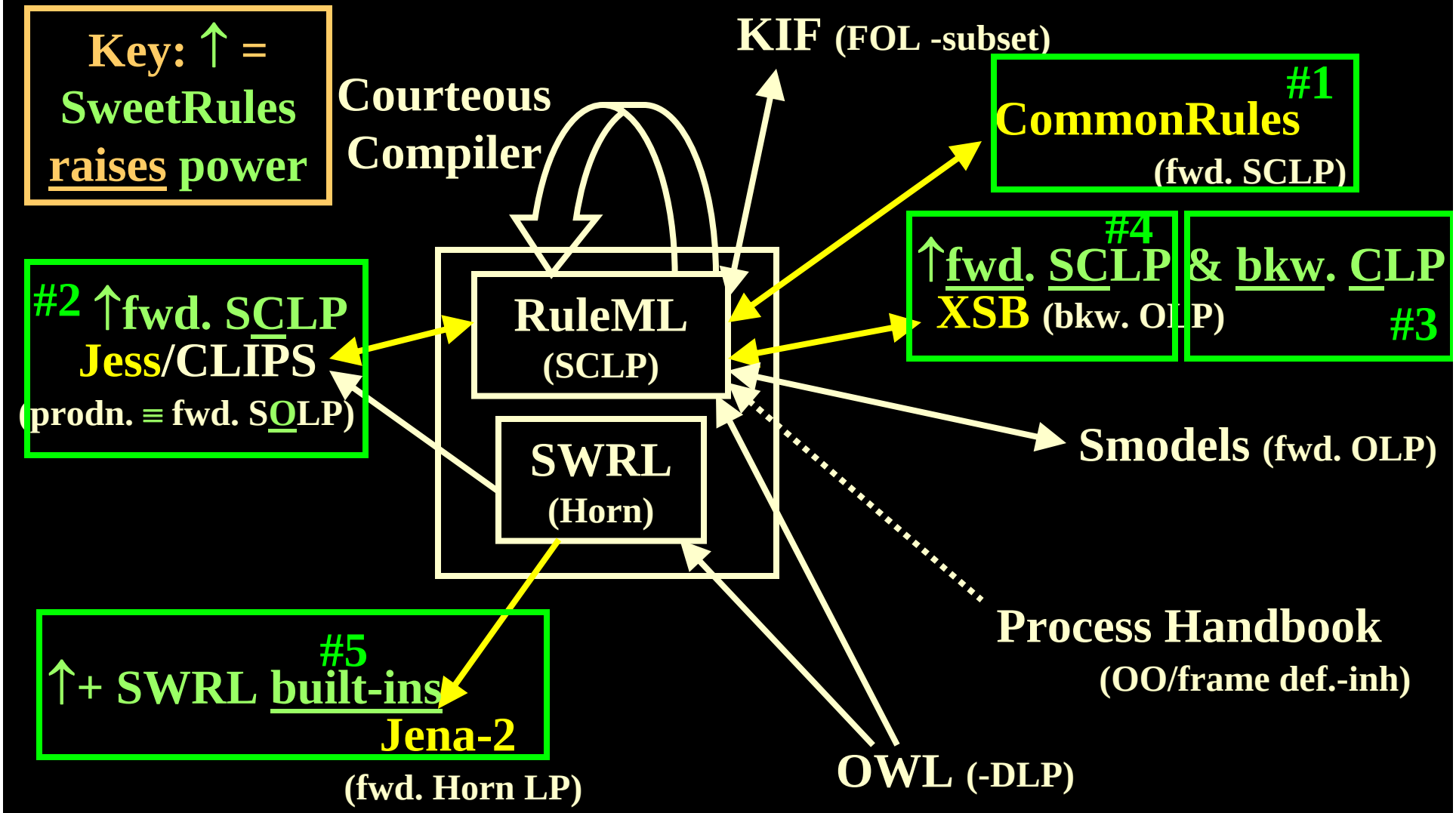



**( SCLP = Situated Courteous Logic Programs. OLP = Ordinary LP (plain NAF) )**

# *SweetRules Inferencing Capabilities Today: Overview*

- Inferencing engines in RuleML/SWRL via translation:
  - Indirect inferencing:
    1. translate to another rule system, e.g., {XSB, Jess, CommonRules, or Jena}
    2. run inferencing in that system's engine
    3. translate back
  - Can use composite translators

# *SweetRules V2.0+ New Inferencing Engines*



# *Novel Capabilities I*

- RuleML-based interoperability, knowledge-merging, reasoning for commercially important kinds of rules, e.g.,
  - Production rules ↔ Prolog, with strong semantic equivalence
  - Platform with pluggability and automatic tool composition
- Supports Correct Negation-As-Failure in Production Rules, via new techniques
- Newly Uses Courteous Compiler to support Courteous feature (prioritized conflict handling) even in systems that don't directly support it, as long as they support negation-as-failure
  - E.g., in XSB Prolog, Jess, Smodels
- New Include-a-KB mechanism, similar to owl:imports (prelim. RuleML V0.9)
  - Include a remote KB that is translatable to RuleML
- Uses New Action Launcher component to support Situated effecting feature (actions triggered by conclusions) even in systems that don't directly support it. Facts input, actions output.
  - E.g., in SweetXSB forward inferencing

# *Additional Firsts in Implementation*

- Forward Situated Courteous LP inferencing+action with intrinsically highly scaleable run-time performance, and moreover with general non-stratified NAF
  - Both XSB/Prolog and Jess/Rete/production-rules reportedly scale very well to very large rulebases (~100K+ non-fact rules, many Millions facts)
- Backward Courteous LP inferencing for general non-stratified NAF, and scaleably in above sense
- RuleML Presentation Syntax Support:
  - Includes Situated feature
  - Generator. *Parser is in testing.*
- WSDL Web Services permitted as procedural attachments
  - Initially, only for effecting not yet sensing. Dynamic.

# *Novel KB Merging of Rules + Ontologies*

- Combine:
  - Multiple SCLP RuleML (/ SWRL) rulebases
    - Or any knowledge base that is translatable into RuleML
  - Heterogeneous kinds of rules
    - E.g., originally XSB rules + Jess facts
    - These get translated and union'd into a single RuleML rulebase (possibly virtual)
  - OWL ontologies
    - Translate Description Logic Programs (DLP) subset of OWL into RuleML
    - Hybrid reasoning via DLP-fusion, i.e., LP inferencing after translate
  - OO/Frame ontologies with default inheritance
    - E.g., Process Handbook ontologies
    - … which get translated to (S)CLP rules

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work

# *F(act)-Queries*

- Queries that return facts, rather than bindings
  - I.e., the facts that correspond to substituting ground bindings
- It's convenient and natural to send such queries, and expect such fact-set answers, in e-contracts.
- E.g., "please tell me your price", "please give me your billing address".
- Experimental extension of RuleML syntax to support this

# *Communicated KB of Queries+Rules*

- Include queries along with rules in communicated KB's
  - E.g., in exchanged contract proposals
- Experimental extension of RuleML syntax to support this

# *Represent Default-Inheritance Object-Oriented Ontologies Via Courteous LP*

- Default-inheritance object-oriented ontologies are ubiquitous in business process realm:
  - Java, C++ frameworks
  - Frame-based systems
- Override or cancel inheritance at subclass.
- OWL, Description Logic, FOL cannot represent default behavior: monotonic only.
- Nonmonotonic/default character increases reuse as compared to monotonic-only.
- Courteous LP can represent them nicely.
  - E.g., SweetPH represents Process Handbook OO businss process ontology (5000 processes, 38000 axioms) [Grosof & Bernstein 2003]

# *Example of Default-Inheritance OO Ontologies in Courteous LP*

```
{buyRegular} paymentMode(?quoteID,invoice) :- Buy(?quoteID).

  /* BuyWithCredit is a subclass of Buy */
Buy(?quoteID) :- BuyWithCredit(?quoteID).

{buyCredit} paymentMode(?quoteID,credit)
                              :- BuyWithCredit(?quoteID).
overrides(buyCredit, buyRegular).
```

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work

# *Advantages of Standardized SW Rules for Policies, e.g., Authorization/Security*

- Easier Integration: with rest of business policies and applications, business partners, mergers & acquisitions
  - Enterprise integration, B2B
- Familiarity, training
- Easier to understand and modify by humansChange management
- Quality and Transparency of implementation in enforcement
  - Provable guarantees of behavior of implementation
- Reduced Vendor Lock-in
- Expressive power
  - Principled handling of conflict, negation, priorities
- $\Rightarrow$ Agility, change management $\uparrow$

# *Advantages of SW Rules, cont'd:*
# *Loci of Business Value in Policy Management*

- Reduced system dev./maint./training costs
- Better/faster/cheaper policy admin.
- Interoperability, flexibility and re-use benefits
- Greater visibility into enterprise policy implementation $\Rightarrow$ better compliance
- Centralized ownership and improved governance by Senior Management
- Rich, expressive policy management language allows better conflict handling in policy-driven decisions
- Strategic agility, incl. wrt business model

# *SWS and Rules Summary*

*** SWS Tasks Form 2 Distinct Clusters,*

*each with associated Central Kind of Service-description Knowledge and Main KR*

1. Security/Trust, Monitoring, Contracts, Advertising/Discovery, Ontology-mapping Mediation
   - Central Kind of Knowledge: Policies
   - Main KR: Nonmon LP (rules + ontologies)
2. Composition, Verification, Enactment
   - Central Kind of Knowledge: Process Models
   - Main KR: FOL (axioms + ontologies)
     - + Nonmon LP for ramifications (e.g., cf. Golog)

- Thus RuleML & SWSF specify both Rules, FOL
  - Fundamental KR Challenge: "Bridging" Nonmon LP with FOL
    - SWSF experimental approach based on hypermon. [Grosof & Martin]

# *SW Rules: Use Cases from our research*

- Contracts/negotiation, advertising/discovery
  - E-procurement, E-selling
  - Pricing, terms & conditions, supplier qualification, …
- Monitoring:
  - Exception handling, e.g., of contract violations
    - Late delivery, refunds, cancellation, notifications
  - Notifications, personal messaging, and other workflow
- Trust Policies: authorization, confidentiality & privacy, security, access control
  - E.g., financial services, health care
    - *Extensive analysis of business case/value*
- Semantic mediation: rule-based ontology translation, context-based information integration

# *Future Work Directions*

- More scenarios, esp. in SWS policy/SCAMP task cluster
- Integration of more expressive ontologies from OWL, FOL (beyond DLP)
  - Extend DLP in various ways
  - Use hypermonotonic reasoning approach (new KR theory) [SWSF 2005]
    - Map FOL ↔ Courteous LP
    - View nonmon LP as weakened FOL: sound, incomplete
- More integration into e-business communication and Web Services, following our SWS vision

# *Outline*

- Introduction and Context: Semantic Web Services for E-Business; Policies
- Overview: SweetDeal Approach, New Extensions
- More Details: SweetDeal, SCLP, KB merging, SweetRules
- Procurement Scenario
- Fact-queries, as part of communicated KB's
- OO default inheritance ontologies, as Courteous LP
- Relationship to E-Business Messaging Standards / Platforms
- Business Value Analysis
- Conclusions & Future Work