Combining Semantic Web Rules with Ontologies: New KR Theory and Tools

Benjamin Grosof

MIT Sloan School of Management

Information Technologies group

http://ebusiness.mit.edu/bgrosof

Slides of 1-hour Invited Talk at PPSWR04

(Workshop on Principles and Practice of Semantic Web Reasoning)

http://www.pms.ifi.lmu.de/PPSWR04

Sponsored by REWERSE (Reasoning on Web with Rules and Semantics), a European Union Network of Excellence; http://www.rewerse.net

Held Saint-Malo, France, Sept. 8, 2004

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL

- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL

- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

My 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

- Pilot Business Application Scenarios
  - End-to-end e-contracting, e.g., in manufacturing supply chain
    - SweetDeal approach using rules
  - Trust policies for security authorization etc., incl. in financial
  - Financial information and reporting:
    - ECOIN approach mapping ontologies
  - Other: travel, …

- Analyze Prospective Early Adopter Areas
  - Strategy: Adoption Roadmap; Market Evolution
  - Entrepreneurial Opportunities

Core SW/KR Research Challenges on Rules and Ontologies

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

- Describing business processes & web services via rules + ontologies
  - Capture object-oriented process ontologies
    - Default inheritance via rules (e.g., Courteous Inheritance)
    - Wrapper/transform to legacy C++, Java, UML
    - Develop open source knowledge bases (e.g., MIT Open Process Handbook Initiative)

  - Also:
    - Rules query web services (e.g., in RuleML Situated feature)
    - Rules trigger actions that are web services (e.g., ditto)
    - Event triggering of rules (e.g., capture ECA rules in RuleML)
    - Rules in process models, e.g., cf. OWL-S, PSL

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:
    - Pioneered e-contracting/supply-chain; finance; trust; …
  - Overall knowledge representation, e-commerce, intelligent agents
- Pioneered key SW rule and ontology KR approaches:
  - Declarative LP in XML as Web rules interlingua
  - Courteous LP as practical tractable well-behaved expressive prioritized conflict handling extension
  - Situated LP as declaratively clean, practical, expressive, and disciplined extension for procedural attachments
  - Description LP as bridge between OWL/DL and LP.
- Co-Founder, Rule Markup Language Initiative – the leading emerging standards body in semantic web rules (http://www.ruleml.org)
- Contracts/Rules Area Editor, Semantic Web Services Initiative – which coordinates world-wide SWS research and early standards (http://www.swsi.org)

Quickie Bio II: REWERSE roles

- Co-led first preproposal phase of creation of REWERSE
- On advisory board of REWERSE.

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

URI Ontological Reference Approach

- A RuleML predicate (or individual / logical function) is specified as a URI, that refers to a predicate (or individual / logical function, respectively) specified in another KB, e.g., in OWL.

- Application pilot and first use case: in SweetDeal e-contracting system (design 2001, prototype early 2002).

- Approach was then soon incorporated into RuleML and adopted in SWRL design (which is based mainly on RuleML), and used heavily there.

- Issue: want to scope precisely which premises in an overall ontological KB are being referenced.
  – Approach in our current work: define a KB (e.g., a subset/module) and reference that KB.

URI Ontological Reference Approach Example *SCLP TextFile Format for RuleML*

```
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:_opr> <drm:tup>
          <drm:ind>co123</drm:ind> <drm:var>Cust</drm:var>
         </drm:tup> </drm:atom>

... </drm:andb> </drm:_body>  </drm:imp>
```

drm = namespace for RuleML

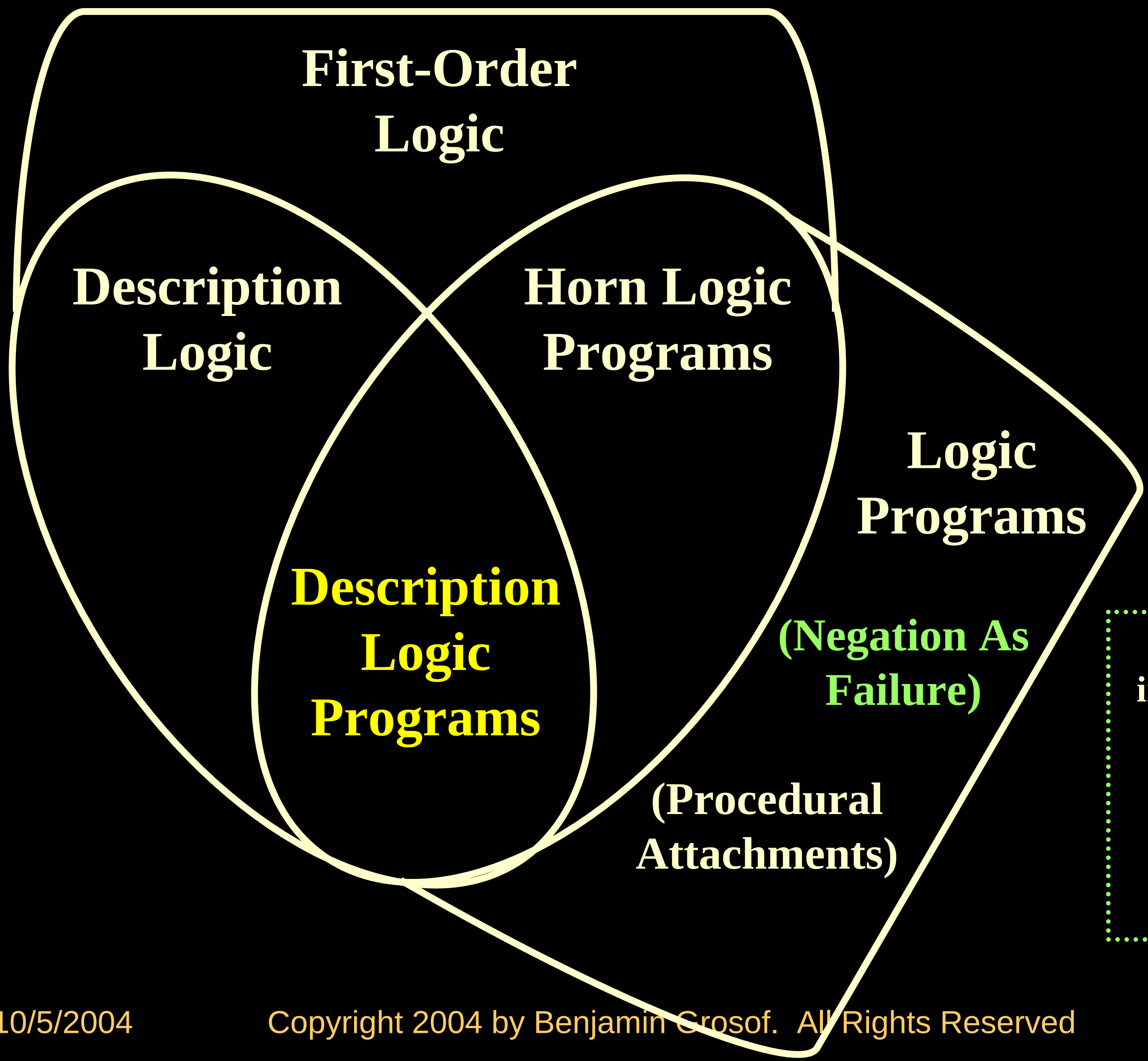
Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL

- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL

- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Venn Diagram: Expressive Overlaps among KR's



NB: Nonmon LP, including Courteous, relies on NAF as fundamental underlying KR expressive mechanism

Overview of DLP KR Features

- DLP captures completely a subset of DL, comprising RDFS & 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 more DL statements beyond RDFS:
  - Using Intersection connective (conjunction) in class descriptions
  - Stating that a property (or inverse) is Transitive or Symmetric
  - Using Disjunction or Existential in a *subclass* expression
  - Using Universal in a *superclass* expression

  - ∴ "OWL Feather" – subset of OWL Lite
    - Update summer 2004: New Related Effort is "OWL Lite Minus" by WSMO

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
  - $\Rightarrow$ Exploit: Scaleability of LP/DB engines >> DL engines , as |instances| $\uparrow$.

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

- Facilitate rule-based mapping between ontologies / “contexts”

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Expressiveness of SWRL (V0.6)

SWRL expressiveness =

1. OWL-DL (i.e., SHOIQ Description Logic (DL) which is an expressive subset of FOL)
2. + <u>Horn</u> FOL rules, with no logical functions, where each predicate may be:
   - OWL named class (thus arity 1)
     - More generally, may use a complex class, but this is expressively inessential – can just replace by a named class and define that named class as equivalent to the complex class.
   - OWL property (thus arity 2)
   - OWL data range (thus arity 1)
     - RDF datatype
     - set of literal values, e.g., {3} or {1,2,3,4,5} or {“Fred”,“Sue”}
3. + some <u>built-ins</u> (mainly XML-Schema datatypes and operations on them)
   - This is new with V0.6
   - (All have arity 1 or 2.)
   - Plan: the set of built-ins is extensible

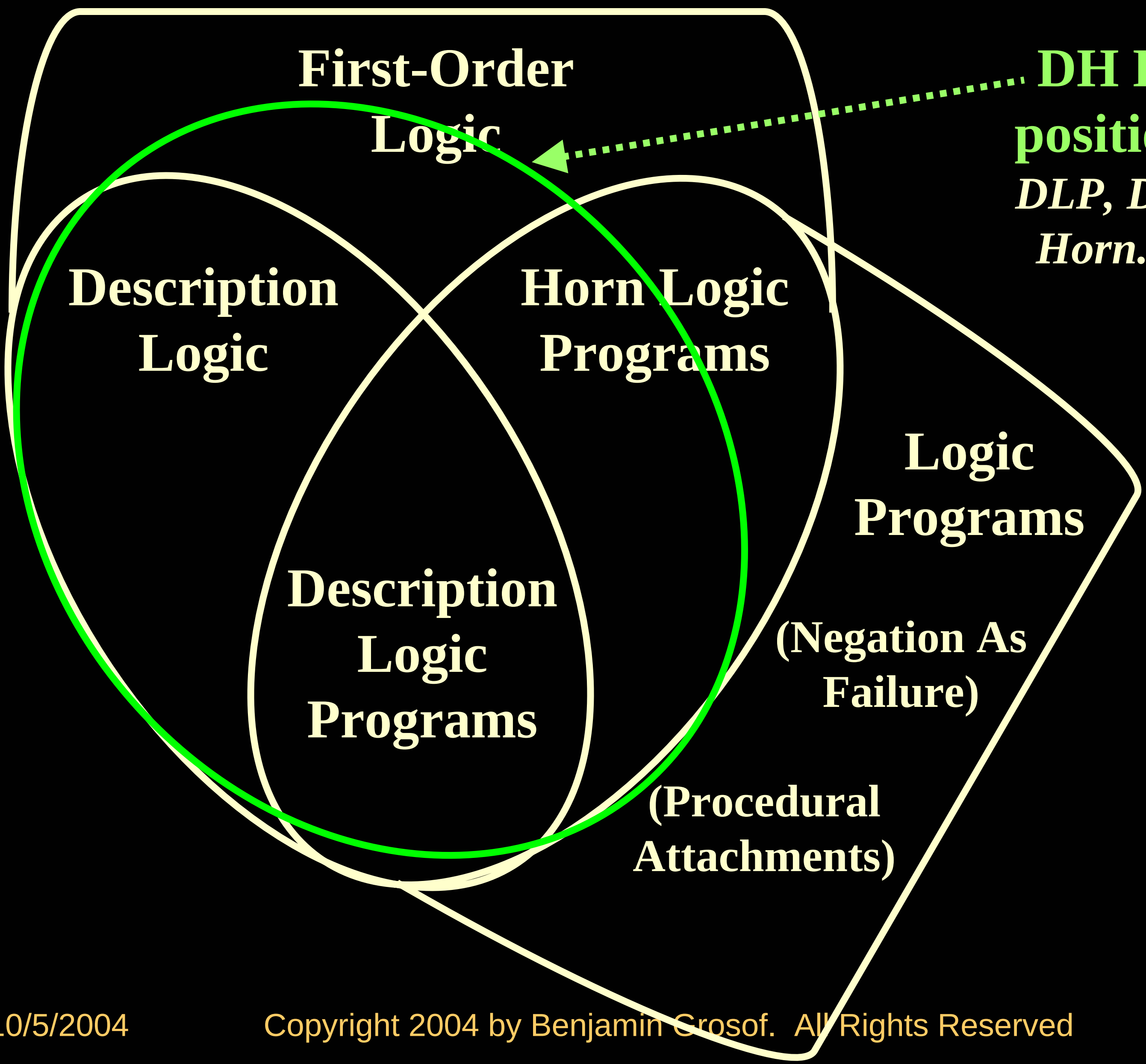
- The fundamental KR is an expressive subset of FOL
  - We’ll call it “DH” here. (It doesn’t have a real name yet.)
  - Its expressiveness is equivalent to: DL + function-free Horn.

"Warning Label"

1. The Theory of DH is Little Explored Territory as a KR.
   - In its full generality, DH is a relatively unstudied fragment of FOL.
   - Its worst-case computational complexity is undecidable and is not known to be better than that of full FOL (e.g., for the propositional case).
   - There are not yet efficient algorithms known for inferencing on it "natively" as a KR.

2. To ensure extensibility of SWRL rulebases to include LP features that go beyond Horn expressiveness, restrict the OWL ontologies used within SWRL to be in the DLP subset of OWL-DL. E.g.:
   - If you want to use nonmonotonicity / negation-as-failure / priorities in your rules
   - If you want to use procedural attachments that go beyond the SWRL built-ins
     - E.g., effectors/actions with side effects

Venn Diagram: Expressive Overlaps among KR's



DH KR's rough position. ***Subsumes DLP, DL, and part of Horn. Subsumed by FOL.***

Design Perspective

Alternative points in design space:

1. partial LP + full DL = SWRL V0.6

versus

2. full LP + partial DL = SCLP RuleML V0.8+

(with DLP OWL2RuleML)

(SCLP = Situated Courteous Logic Programs KR)

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Need for Other Kinds of Ontologies besides OWL

- Kinds of ontologies practically/commercially important in the world today*:
  - SQL DB schemas, E-R, UML, OO inheritance hierarchies, LP/FOL predicate/function signatures; equations and conversion-mapping functions; XML-Schema
- OWL is still emerging.
- Overall relnsh of OWL to the others is as yet largely unclear
  - There are efforts on some aspects, incl. UML
- OWL cannot represent the nonmon aspects of OO inh.
- OWL does not yet represent, except quite awkwardly:
  - n-ary signatures
  - ordering aspects of XML-Schema

- (*NB: Omitted here are statistically flavored ontologies that result from inductive learning and/or natural language analysis.)

Need for Other Kinds of Ontologies besides OWL, cont.'d

- Particularly interesting:
  - OO-ish nonmon taxonomic/frames
  - Equations and context mappings cf. ECOIN – can be represented in FOL or often in LP
  - OWL DL beyond DLP

- Builtins (sensed) are a relatively simple kind of shared ontology
  - SWRL V0.6 and forthcoming RuleML V0.9

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Review: Situated and Courteous extensions of LP

- 1. 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
- 2. Courteous Logic Programs:

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

- These extensions combine essentially orthogonally.
  - Sensors may be the subject of prioritized conflict handling, so it is useful to give them labels.

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.)

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.
- 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).

EECOMS Example of Conflicting Rules: Ordering Lead Time

- Vendor’s rules that prescribe how buyer must place or modify an order:
- A) 14 days ahead if the buyer is a qualified customer.
- B) 30 days ahead if the ordered item is a minor part.
- C) 2 days ahead if the ordered item’s item-type is backlogged at the vendor, the order is a modification to reduce the quantity of the item, and the buyer is a qualified customer.

- Suppose more than one of the above applies to the current order? **Conflict!**

- Helpful Approach: **precedence** between the rules. Often only *partial* order of precedence is justified. E.g., C > A.

Courteous LP's: Ordering Lead Time Example

- <leadTimeRule1> orderModificationNotice(?Order,14days)
- ← preferredCustomerOf(?Buyer,?Seller) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- <leadTimeRule2> orderModificationNotice(?Order,30days)
- ← minorPart(?Buyer,?Seller,?Order) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- <leadTimeRule3> orderModificationNotice(?Order,2days)
- ← preferredCustomerOf(?Buyer,?Seller) ∧
- orderModificationType(?Order,reduce) ∧
- orderItemIsInBacklog(?Order) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- overrides(leadTimeRule3 , leadTimeRule1) .
- (⊥ ← orderModificationNotice(?Order,?X) ∧
- orderModificationNotice(?Order,?Y)) ← (?X ≠?Y) .

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}

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL

- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL

- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

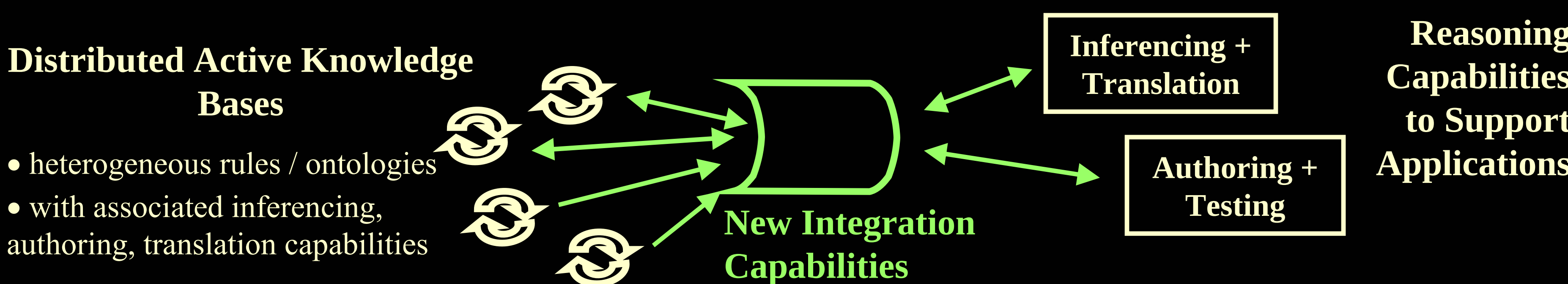
SweetRules Overview

Key Ideas:

- 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.
- 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.
  - Plus semantics-preserving translations between different rule languages/systems/families

Application Areas (prototyped scenarios):

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



SweetRules *Concept, Architecture, and Goals*

- 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 and testing of rulebases
  - Open, lightweight, extensible, pluggable architecture overall
- Goals:
  - Research vehicle: embody ideas, implement application scenarios (e.g., contracting, policies)
    - Situated Courteous Logic Programs (SCLP) KR
    - Description Logic Programs (DLP) KR which is a subset of SCLP KR
  - Proof of concept for feasibility, including of translations between heterogenous families of rule systems
    - Encourage others: researchers; industry esp. vendors
  - Catalyze open source communal toolset efforts
    - *Initial open-source release on SemWebCentral.org ~Nov. 2004*

SweetRules Context and Players

- Part of SWEET = "Semantic WEb Enabling Tools" (2001 –)
  - Other parts:
    - SweetDeal for e-contracting
      - Which uses SweetRules
- Cross-institutional. Collaborators invited!
  - Originated and coordinated by MIT since 2001
  - Code by MIT, UMBC, U. Karlsruhe, U. Zurich
  - Uses code by IBM, SUNY Stonybrook, Sandia Natl. Labs, Helsinki
  - More loosely, several other institutions cooperating: BBN, NRC/UNB, Stanford
  - Many more are good targets: subsets of Flora, cwm, Triple, Hoolet, ?Jena, DRS, ?ROWL, KAON (main), JTP, SWI Prolog, ...

RuleML KR Expressiveness

- SweetRules supports: RuleML in its highly expressive Situated Courteous Logic Programs (SCLP) extension, V0.8
  - Horn LP …
  - + Negation-As-Failure = "Ordinary" LP (OLP)
  - + Courteous feature: prioritized conflict handling (partially ordered priorities, mutual exclusion integrity constraints, e.g., for partial-functionality; limited classical negation of atoms, e.g., p vs. not-p in heads)
  - + Situated feature: procedural attachments
    - Sensors: external queries when rule body atoms are tested
      - Built-ins in SWRL V0.6 correspond to sensors.
    - Effectors: external actions triggered when rule head atoms are concluded
- RuleML also supports *referencing* OWL (or other) ontologies
    - URI predicate name (in RuleML rule) refers to class or property (in OWL axioms)
  - This was pioneered in SweetDeal using SweetRules
  - The same approach was then taken in SWRL V0.5+

Rule and Ontology Languages/Systems That Interoperate via SweetRules and RuleML, Today

1. RuleML
   - SCLP extension, V0.8
2. XSB (the pure subset of it = whole Ordinary LP)
   - Backward. Prolog. Fast, scalable, popular. Good support of SQL DB's (e.g., Oracle) via ODBC backend. Full well-founded-semantics for OLP. Implemented in C. By SUNY Stonybrook. Open source on sourceforge. Well documented and supported. Papers.
3. Jess (a pure subset of it = a large subset of Situated Ordinary LP)
   - Forward. Production Rules (OPS5 heritage). Flexible, fast, popular. Implemented in Java. By Sandia National Labs. Semi-open source, free for research use. Well documented and supported. Book.
   - *Uses recent novel theory for translation between SOLP and Production Rules.*
4. IBM CommonRules (whole = large subset of stratified SCLP)
   - Forward. SCLP. Implemented in Java. Expressive. By IBM Research. Free trial license, on IBM AlphaWorks (since 1999). Considerable documentation. Papers. Piloted.
   - Implements the Courteous Compiler (CC) KR technique.
     - which reduces (S)CLP to equivalent (S)OLP, tractably.
   - Includes bidirectional translators for XSB, KIF, Smodels.
   - Its overall concept and design was point of departure for several aspects of SweetRules

Rule and Ontology Languages/Systems That Interoperate via SweetRules and RuleML, Today, continued

5. Knowledge Interchange Format (KIF) (a subset of it = an extension of Horn LP)
   - First Order Logic (FOL). Semi-standard, morphing into Simple Common Logic ISO standard. Several tools support, e.g., JTP. Research language to date.
     - Note: FOL is superset of DLP and of SWRL's fundamental KR.
6. OWL (the Description Logic Programs subset)
   - Description Logic <u>ontologies</u>. W3C standard. Several tools support, e.g., FACT, RACER, Jena, Hoolet, etc.
   - *Uses recent novel DLP theory for translation between Description Logic and Horn LP.*
7. Process Handbook (large subset = subset of SCLP)
   - Frame-style object-oriented <u>ontologies</u> for business processes design, i.e., for services descriptions. By MIT and Phios Corp. (spinoff). Large (5000 business processes). Practical, commercial. Good GUI. Open source license in progress. Available free for research use upon request. Includes extensive textual information too. Well documented and supported. Papers. Book. Dozens of research users.
   - *Uses recent novel SCLP representation of Frames with multiple default inheritance.*
8. Smodels (NB: somewhat old version; large subset = finite OLP)
   - Forward. Ordinary LP. Full well-founded-semantics or stable semantics. Implemented in C. By Helsinki univ. Open source. Research system.

SweetRules Capabilities and Components Today

- Translators in and out of RuleML:
  - RuleML ↔ {XSB, Jess, CommonRules, KIF, Smodels}
  - RuleML ← {OWL, Process Handbook} (one-direction only)
  - SOLP RuleML ← SCLP RuleML (Courteous Compiler)
- Inferencing engines in RuleML via translation:
  - Simple drivers translate to another rule system, e.g., CommonRules, Jess, or XSB, then run inferencing in that system's engine, then translate back.
  - Observation: Can easily combine components to do other kinds of inferencing, in similar indirect style, by combining various translations and engines.
- Authoring and Testing front-end: currently rudimentary, partial
  - Command-line UI + Dashboard GUI with set of windows
  - Edit rulebases. Run translations. Run inferencing. Compare.
  - Edit in RuleML. Edit in other rule systems' syntaxes. Compare.
  - View human-oriented presentation syntax. View XML syntax. (Future: RDF.)

SweetRules Capab. & Compon. Today, cont.'d

- 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., XSB Prolog, Jess, Smodels
  - Uses Courteous Compiler component from IBM CommonRules
- Has Include-a-KB mechanism, similar to owl:includes (prelim. RuleML V0.9)
  - Include a remote KB that is translatable to RuleML
- Uses IBM CommonRules translators: CommonRules ↔ {XSB, KIF, Smodels}
- Some components have distinct names (for packaging or historical reasons):
  - SweetJess translation and inferencing RuleML ↔ Jess
  - SweetOnto translation RuleML ← OWL
- Code base: Java, XSLT; convenience shell scripts (for testing drivers)
- *In-development:* Pluggability Architecture with detailed interfaces
  - Add your own translator/inferencing-engine/authoring/testing tools
  - Compose tools: translator1 ⊗ translator2; translator ⊗ inferencing-engine; etc.

SweetRules: More Goals

- Additional Goals:
  - More meat to pluggable architecture
  - More authoring/UI capabilities
  - More SWRL support
  - More wrt additional kinds of rule systems:
    - ECA rules, SQL (needs some theory work, e.g., events for ECA)
    - RDF-Query and XQuery
  - More wrt connections-to / support-of web services:
    - Importing knowledge bases / modules, procedural attachments, translation/inferencing, events, …
  - Explore applications in services, e.g., policies, contracts
- More Collaborators Invited!
  - Many more rule/ontology systems are good targets for interoperation/translation:
    - Flora, cwm, Triple, Hoolet, Jena, DRS, ROWL, KAON, JTP, SWI Prolog, …

More about Combining Rules with Ontologies

There are several ways to use SweetRules to combine rules with ontologies:

1. By reference: via URI as name for predicate
2. Translate DLP subset of OWL into RuleML
   - Then can add SCLP rules
     - E.g., add Horn LP rules and built-in sensors $\Rightarrow$ interesting subset of the SWRL V0.6 KR
     - E.g., add default rules or procedural attachments
3. Translate non-OWL ontologies into RuleML
   - E.g., object-oriented style with <u>default inheritance</u>
     - E.g., Courteous Inheritance for Process Handbook ontologies
4. Use RuleML Rules to map between ontologies
   - E.g., in the spirit of the Extended COntext Interchange (ECOIN) approach/system.
   - SWRL V0.6 good start for mapping between non-DLP OWL ontologies.

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL

- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL

- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

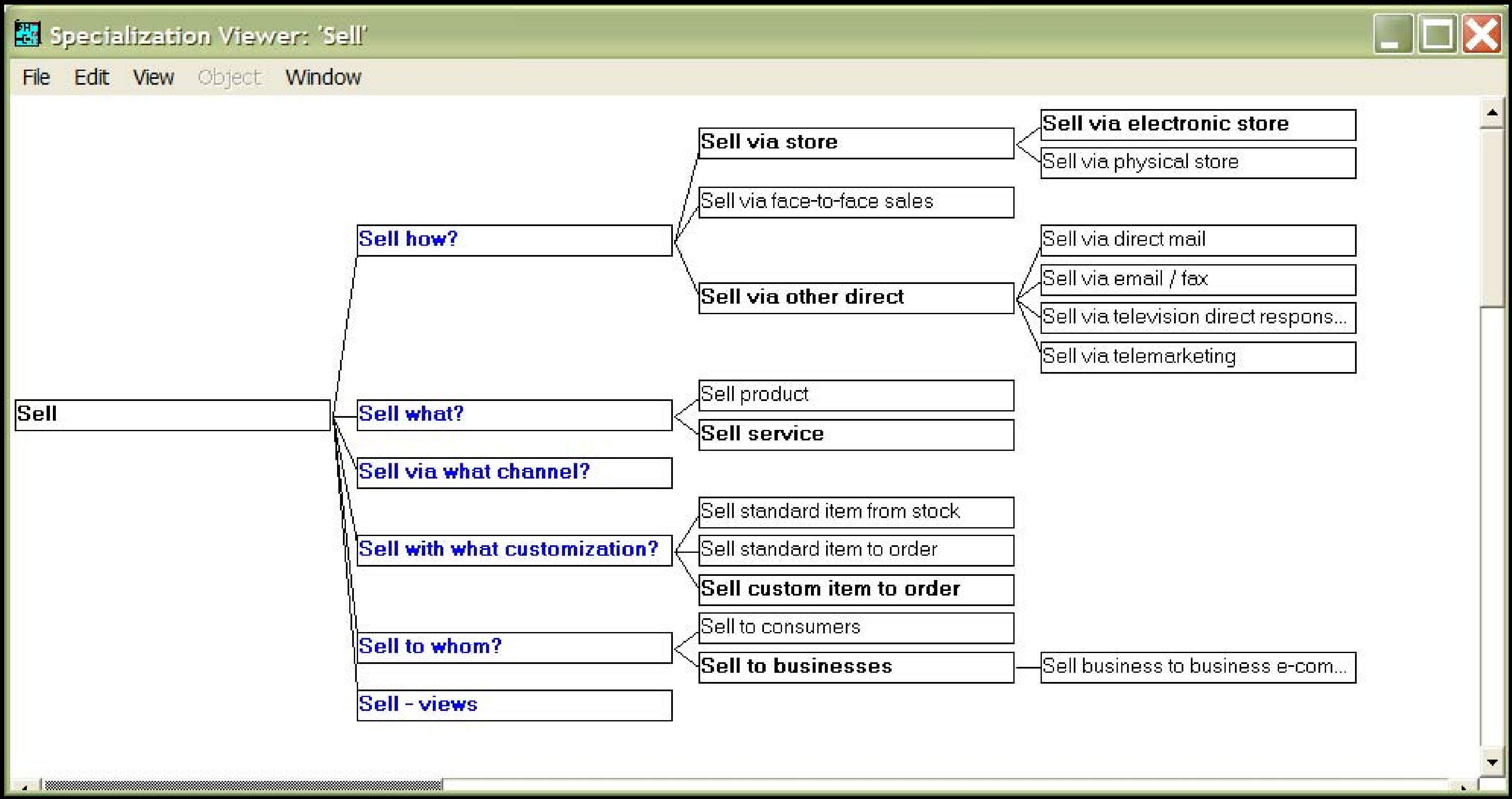
Outline of Courteous Inheritance Section

- 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

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
  - Open Process Handbook Initiative: *an open-source version, is in progress.* (http://ccs.mit.edu/ph)
- 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)



New Technical Approach: Courteous Inheritance in the Process Handbook

- 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. Esp. pages 7-10.
  - Example: selling process
- *In-development currently:* Optimized version
  - generates a linear -size/-time OLP after courteous compilation

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Standards News: SWSL-Rules, FOL RuleML and Joint Committee

- Issue of FOL vs. (nonmon) LP as foundation KR:
  - JC struggled for ~18 months with, decided to go initially with FOL
  - SWSL struggled with that issue too for ~15 months, decided to go initially with both
  - But: Issue: overall, the bridge between FOL and LP is undesirably thin
    - e.g., plan to specify service-concept ontologies in FOL,
    - but it's unclear how to utilize those in LP;
    - ideally want to avoid a separate spec in LP

Standards News, cont.'d: SWSL-Rules

- SWSL-Rules design (primarily by Grosof & Kifer) is highly expressive – it includes a novel combination of two KR feature sets:

1. {HiLog, frames syntax, reification} (from F-Logic and Flora)
2. {Courteous, Situated sensing, URIreferences} (from SCLP, DLP, and RuleML)

- (1.) and (2.) combine essentially orthogonally
  - In-development: defining needed minor restrictions combo with Flora features
- Initially it is just a human-editing string presentation syntax, not yet a markup syntax
- Plan to make this an extension in/of RuleML
- The added expressiveness aids convenience in specifying ontological aspects of knowledge
  - e.g., OWL-Full aspects, frame syntax, prioritized override in OO subclassing.

Standards News, cont.'d: SWSL-Rules

- Extension of RuleML to FOL syntax features (e.g., quantifiers) is being developed by Joint Committee + RuleML Initiative. This is aimed to supersume SWRL.
- Joint Committee will probably hibernate when that is done (in a few months).
- SWSL is requesting RuleML to carry the ball after that on SWSL's LP and FOL needs.

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL

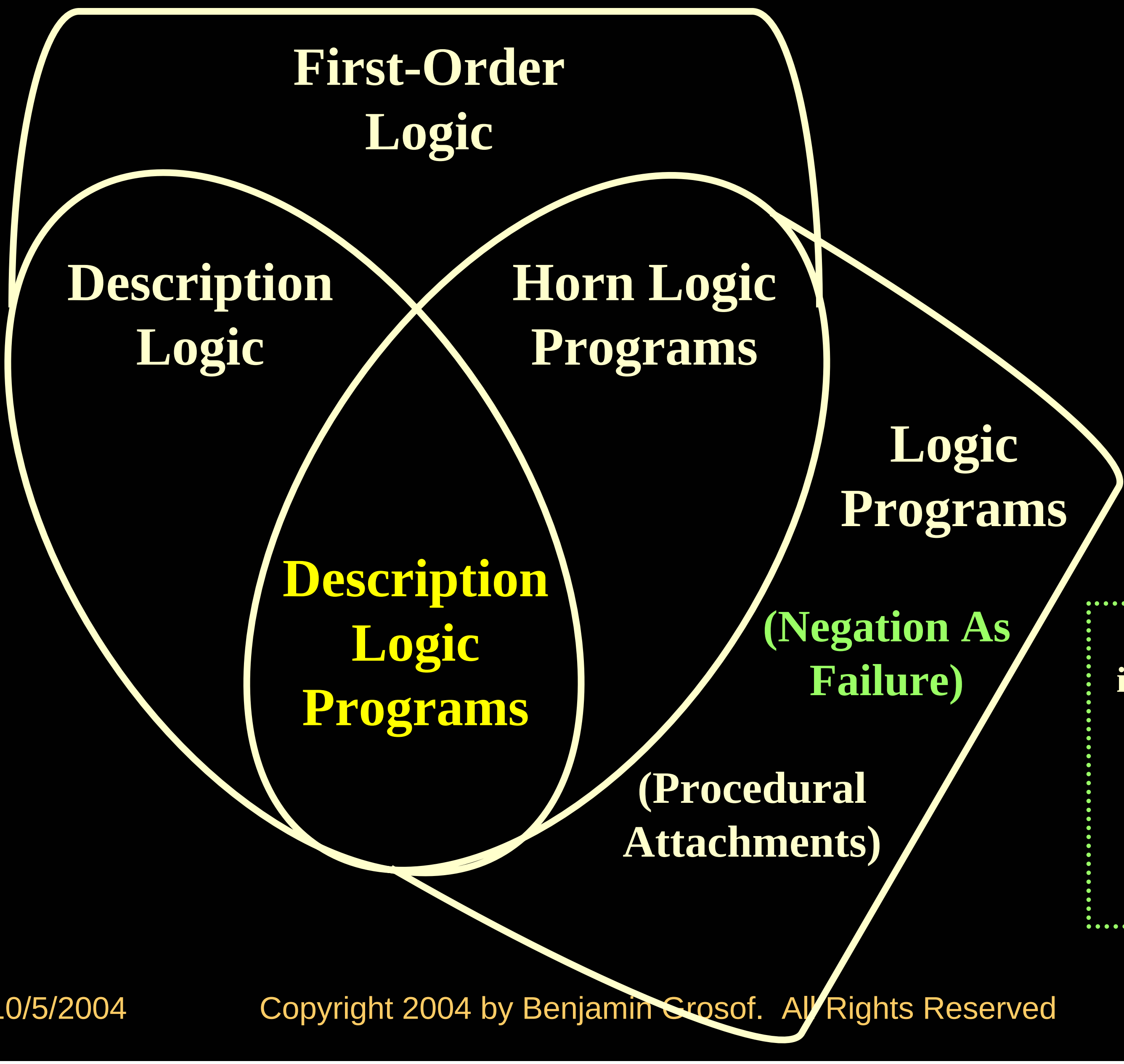
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL

- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Aspiration: Unifying FOL and Nonmon LP

- A challenge, a holy grail:
  - Wouldn't it be nice to have a single KR that unifies <u>all</u> of FOL and nonmon LP?
  - … or at least <u>more</u> of FOL and nonmon LP?

Venn Diagram: Expressive Overlaps among KR's



NB: Nonmon LP, including Courteous, relies on NAF as fundamental underlying KR expressive mechanism

Motivations I: Some Potential Uses for
Unifying FOL and Nonmon LP KR's for Rules+Ontologies

- Tightly integrate full OWL ontologies (OWL-DL and OWL-Full) with nonmon LP rules. Increase expressiveness of DLP to all of OWL.
  - Semantics; algorithms; ensure consistency
- Cope robustly with conflict between ontologies, e.g., merging OWL ontologies from many sources
- Permit FOL for ontologies beyond DL/OWL
  - E.g., ECOIN work on equational ontologies and context integration
- Integrate nonmon frame/OO ontologies with mon DL/FOL ontologies

Motivations II: Some Potential Uses for
Unifying FOL and Nonmon LP KR's for Rules+Ontologies

- Integrate SWSL's 2 "wings":
  - LP rules language & service-concept ontologies for contracts, policies, ads, mappings, etc. (SCAMP tasks)
  - FOL language & service-concept ontologies for process model, synthesizing composition, verification, etc. (e.g., cf. PSL)
    - Actually also desire default reasoning to minimize ramifications in reasoning about actions (e.g., cf. Golog)

- Unify the KR foundation of the Semantic Web
  - Represent all the current* major pieces:
    - Rules, ontologies, databases, RDF, queries
    - Semantic Web Services   service descriptions
  - Overcome what has been a major hang-up for Joint Committee and Semantic Web Services Initiative efforts on SW standards design.

(*NB: SW in future should also include probabilistic/statistical KR.)

Logical Hypermonotonicity

The following is current work (paper is in progress).

- New Definition: logical hypermonotonicity:
- A KR S is "hypermonotonic", wrt monotonic KR B, when:

1. S is nonmonotonic
2. Each premise (respectively, conclusion) expression in S can also be viewed* as a premise (respectively, conclusion) expression in B.
3. S is sound wrt B, but (in general) incomplete wrt B
   - I.e., Let X stand for a set of premises, Conc(X,Y) stand for the set of conclusions that are entailed in KR Y by the set of premises X. Then:
     - Forall P. Conc(P,S) $\subseteq$ Conc(P,B) but not vice versa

- More Details:
  - A KR Y is defined as a triple (LP, LC, $\Rightarrow$), where LP is a formal language of sets of premises (i.e., premise expressions), LC is a formal language of sets of conclusions (i.e., conclusion expressions), and $\Rightarrow$ is the entailment relation. We assume here that $\Rightarrow$ is a functional relation.
  - *More generally, one can generalize to have a mapping T from the premises/conclusions of S to the premises/conclusions of B.

Hypermon: Discussion of Definition

- Hypermon is a restricted case of nonmon, in which the nonmon KR's entailed conclusions can be viewed as always unobjectionable, i.e., sanctioned, by an associated mon KR that provides a background "reference" semantics for the premises in the nonmon KR.
- *By contrast, in the previously typical perspective on nonmon, the nonmon reasoning is viewed as unsound, i.e., it goes beyond what the mon semantics sanctions, e.g., as a way to handle "incompleteness" of the (mon semantics of the) available premise info.*
- The spirit of conflict handling is a good match to the hypermon concept. E.g., suppose B is FOL.
  - When P is inconsistent according to FOL, then it's arguably often quite desirable that S is incomplete wrt FOL, since FOL produces a global meltdown in which all sentences are entailed.
  - Even if P is consistent according to FOL, then it's "not so bad" that S is incomplete. In practical inferencing over FOL, since that is computationally and/or algorithmically complex, incompleteness is often acceptable. I.e., many practical FOL tools are (in general) incomplete.

Nonmon LP as Hypermon wrt FOL

Caveat: The following results are in preliminary and summary form.

- Let OLP stand for Ordinary LP (a.k.a. Normal LP, a.k.a. "General" LP), and CLP stand for Courteous LP.

- We assume here the semantics of OLP and CLP is based on the Well Founded Semantics. (NB: this assumption can be generalized.) A conclusion is entailed iff it has truth value *t* in the WFS.

- Obs.: OLP is unsound wrt FOL, if NAF is mapped to classical negation. I.e., Closed World is required as an extra assumption, essentially. Thus OLP is not (directly) hypermon wrt FOL. (NB: If instead NAF is not viewed as a classically interpretable expressive construct, then also OLP is not hypermon wrt FOL.)

Nonmon LP as Hypermon wrt FOL, cont.'d

- However, with some cleverness and the use of Courteous LP, we can establish a hypermon relationship of nonmon LP to FOL. This relationship will actually encompass both OLP and CLP.
- Let CLP2 stand for NAF-free Courteous LP, i.e., CLP restricted to prohibit (explicit) NAF. (NB: CLP2 does include the classical negation operator $\neg$, however.)
- CLP2 has a straightforward very simple mapping (T) to FOL: each rule is viewed as a clause in FOL; likewise, each mutex (mutual exclusion integrity constraint).
- Theorem: CLP2 is hypermon wrt FOL.

Nonmon LP as Hypermon wrt FOL, *cont.'d more*

- Theorem: OLP is expressively reducible to CLP2 via a relatively simple transformation on the premises.

  The transformation M is local and linear-time. E.g.*, let M be defined as:

  1. Replace every NAF'd atom ~p(t) by fp(t). Here, p is a predicate, t is a tuple of terms of appropriate arity for p, ~ is the NAF operator, and fp is a newly introduced predicate.
  2. Add the two rules:
     a. fp(t) ← .
     b. ¬fp(t) ← p(t). Here ¬ is the classical negation operator.

  *There are multiple similar alternative such transformations.

- Theorem: CLP is expressively reducible to CLP2, in like fashion.

- OLP and CLP are thus in this sense <u>indirectly</u> hypermon wrt FOL.
  – View as: their essential expressiveness is hypermon wrt FOL.

Nonmon LP as Hypermon wrt FOL, *cont.'d more*

- Theorem: CLP is always consistent from the viewpoint of FOL. (I.e., it has a consistent set of conclusions.)

- Can thus view conflictful merging/updating in CLP2 as sound, consistent, and incomplete from FOL viewpoint.

- The fundamental KR relationships can be used in more ways too:
  - Import FOL axioms (e.g., ontologies) to become (nonmon) LP rules, mutex's
    - As initial LP premises
    - As sensed facts
  - Export (nonmon) LP conclusions as facts to become FOL axioms

Nonmon LP as Hypermon wrt FOL, cont.'d yet more

- Provides path to formally define and investigate:
  - Merging of LP KB's with FOL KB's, in terms of conclusions or premises, when conflict is absent or present.

- *Further Results in Development, e.g.:*
  - Special cases when (nonmon) LP is consistent, or its updates are monotonic, wrt a given FOL or LP sub-theory/background-theory.
  - Other interesting hypermonotonic KR's:
    - Identify previous ones; tweak or design new ones
  - Extend fundamental CLP expressiveness.

Meaning of the Name: Inspiration behind "Hyper"

- Naming inspirations for why call CLP2 "hyper"monotonic:
  1. Fun … and hopefully catchy too ☺
  2. Available: hardly appears in previous literature.
  3. Analogy: hyperspace

     Overcomes the apparent barrier/limitation of how inconsistency behaves (global fragilility/propagation) in classical logic. "Tunnels through a wormhole" to a consistent, typically contenful, set of conclusions (with localized propagation scope for unresolved conflicts). Enters a regime with different characteristics – monotonic relative to the FOL semantics, but nonmonotonic wrt updates within the LP semantics. Transcends the previous category boundary of monotonicity vs. nonmonotonicity.

     - In science-fiction (and in tachyon physics), traveling through hyperspace overcomes the apparent barrier/limitation of how speed behaves (bounded by speed of light) in normal space-time. Enters a regime with different characteristics – respects the behavior of normal space-time but moves information/matter/energy faster in its own regime. Transcends the previous category boundary framed by relativity theory.

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

Summary of Objectives Motivating SweetRules and Hypermon: Integrating Distributed Rules and Ontologies

Address "the 5 D's" of real-world reasoning ⇒ *desired improvements*:

1. **D**iversity – Existing/emerging kinds of ontologies and rules have heterogeneous KR's. *Handle more heterogeneous systems.*
2. **D**istributedness - of ownership/control of ontology/rule active KB's. *Handle more source active KB's.*
3. **D**isagreement - Conflict (contradiction) will arise when merging knowledge. *Handle more conflicts.*
4. **D**ynamism - Updates to knowledge occur frequently, overturning previous beliefs. *Handle higher rate of revisions.*
5. **D**elay - Computational scaleability is vital to achieve the promise of knowledge integration. *Achieve Polynomial-time (~ databases).*

Summary of Objectives Motivating SweetRules and Hypermon:

Integrating Distributed Rules and Ontologies, cont.'d

BEFORE

AFTER

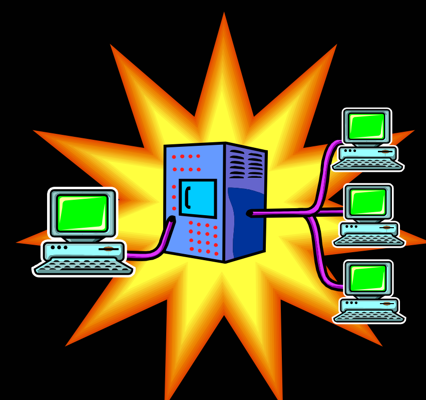
Contradictory conflict is globally contagious, invalidates all results.

⇒

Contradictory conflict is contained locally, indeed tamed to aid modularity.

Knowledge integration tackling the 5 D's (esp. diversity and distributedness) is labor-intensive, slow, costly.

⇒



Knowledge integration is highly automated, faster, cheaper.

Acknowledgements

- Many people have contributed to the work described here.
  - Can't mention everyone here
- Thanks especially to:
  - Shashidhara Ganjugunte, UMBC (SweetRules lead implementer)
  - Said Tabet, RuleML (SweetRules)
  - Chitravanu Neogy, MIT (SweetRules)
  - Abraham Bernstein, U. Zurich (Process Handbook, SweetPH)
  - Boris Motik, U. Karlsruhe (SweetOnto first phase lead implementer)
  - Ian Horrocks, U. Manchester (DLP theory, SWRL)
  - Raphael Volz, U. Karlsruhe (DLP, SweetOnto first phase)
  - Mike Dean, BBN (SWRL, Joint Committee chair)
  - Michael Kifer, SUNY Stonybrook (SWSL-Rules)

Some Resources

- See papers, talk slides, and links at http://ebusiness.mit.edu/bgrosof
- ../#RecentPapersByTopic : (for most below, there are earlier versions too)
  - "Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML", *Electronic Commerce Research and Applications*, 2004.
  - "SweetDeal: Representing Agent Contracts With Exceptions using Semantic Web Rules, Ontologies, and Process Descriptions", *International Journal of Electronic Commerce*, 2004.
  - "Description Logic Programs: Combining Logic Programs with Description Logic", WWW-2003.
  - "SweetJess: Inferencing in Situated Courteous RuleML via Translation to and from Jess Rules", 2003 working paper updating RuleML-2002 Workshop paper.
  - "A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML", EC-99.
  - "Beyond Monotonic Inheritance: Towards Semantic Web Process Ontologies", 2003.
  - "SWRL: A Semantic Web Rules Language Combining OWL and RuleML", 2004.
- ../#RecentSoftware : Links to SweetJess, SweetOnto, CommonRules (where can download)
- RuleML http://www.ruleml.org
- Joint Committee http://www.daml.org/committee
- SemWebCentral http://www.semwebcentral.org
- SWSI/SWSL http://www.swsi.org

Discussion Kickoff

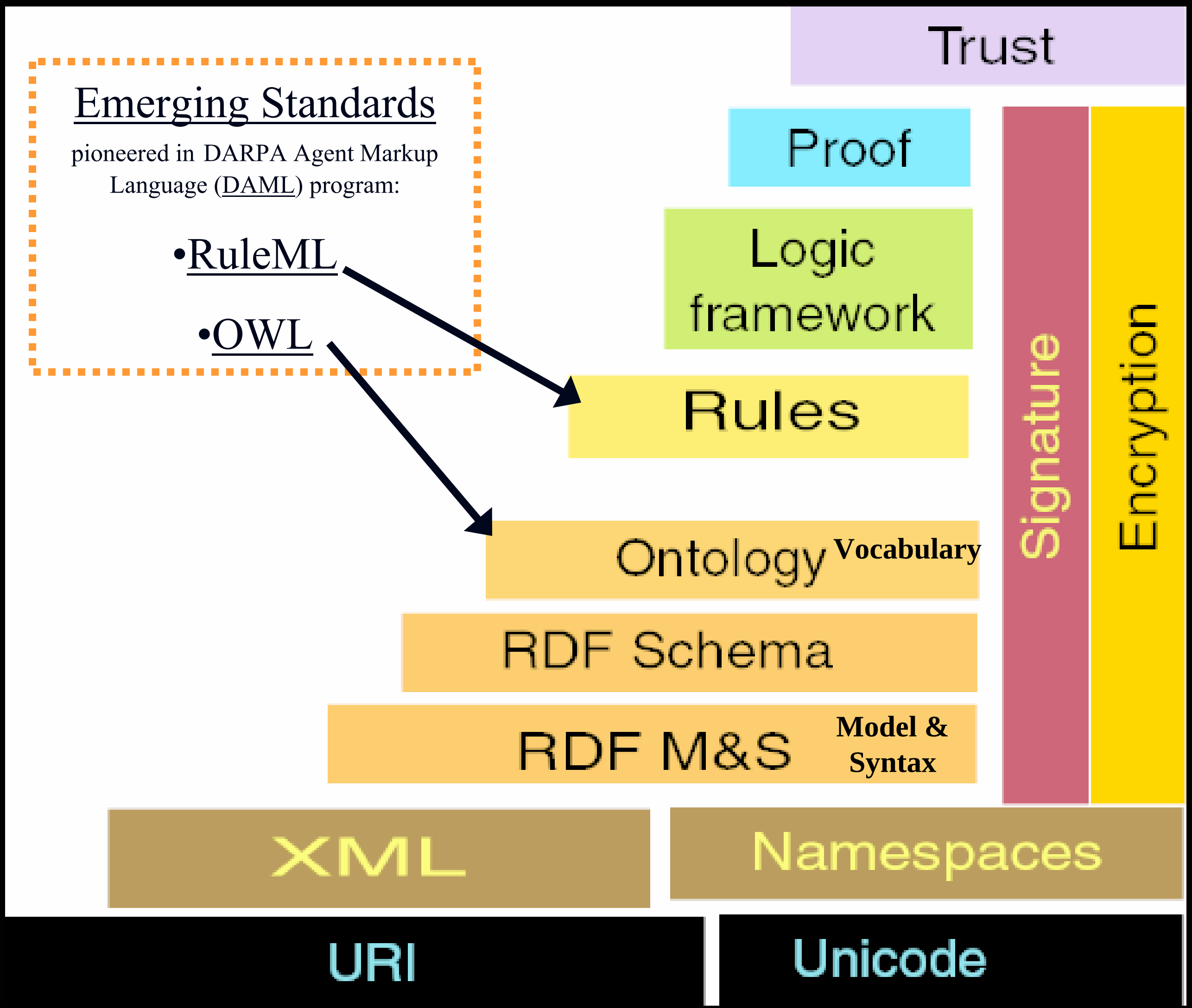
- Thanks!
- Focused here on latest stuff, in workshop spirit
- Would love your comments/feedback, e.g., on:
  - SweetRules Toolset for RuleML: Vision, Design, Status
  - Need for other kinds of Ontologies besides OWL
  - Hypermonotonicity as a Fundamental Approach to Unifying FOL and Nonmon LP

Outline

- Introduction
  - Overall research agenda
  - *Focus here on latest stuff, in workshop spirit – would love your feedback*
  - Review of ontological URI reference approach, Description Logic Programs
- Analysis of SWRL wrt KR relationships
- Analysis of Need for Other kinds of Ontologies besides OWL
- SweetRules Toolset for RuleML: Vision, Design, Status
  - *begin with:* Review of Situated Courteous LP *(skim)*
  - *follow with:* Courteous Inheritance representing MIT Process Handbook ontologies *(skim)*
- Standards News: SWSL-Rules, FOL RuleML and SWRL
- A Fundamental Approach to Unifying FOL and Nonmon LP
  - Hypermonotonicity: Nonmon LP as Incomplete FOL
- Windup; Discussion

OPTIONAL SLIDES FOLLOW

W3C Semantic Web "Stack": Standardization Steps



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

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; $\rightarrow$ 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.

Where Rules Shine in
Goals wrt Key SWS Tasks

- Knowledge reuse in knowledge-based service descriptions:
  - … Across the Key Tasks in our Requirements:
    - Contracts (proposals, request-for-proposals, selection, negotiation, advertising); Discovery; Enactment, Composition; Monitoring, Problem resolution, Exception handling; Verification
    - Business/Trust/Security/Privacy Policies
    - Semantic Interoperability (mappings, specializations)
    - Underlying: Hypothetical Reasoning

Where are the Holdups? … and Challenges for Research

- KR & standards to integrate Rules with Ontologies more expressively
- KR, & later standards, to represent Services descriptions using Rules and Ontologies.
  - A step is our SweetDeal approach; much current work in SWSI.
- KR & strategy to leverage legacy content, e.g., OO service/process ontologies
  - A rich research area. We are doing much current work on that.
    - Preliminary-version approach is available as paper "Beyond Monotonic Inheritance: Towards Semantic Web Process Ontologies" at http://ebusiness.mit.edu/bgrosof
- 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

