DAML Rules
Report for PI Mtg. Nov.-Dec. 2004

by Benjamin Grosof and Mike Dean***
(DAML Rules Co-Chairs)
**MIT Sloan School of Management, http://ebusiness.mit.edu/bgrosof*
***BBN Technologies, http://www.daml.org/people/mdean*

Presented at DARPA Agent Markup Languages program (DAML)
Principal Investigators Meeting (held Nov. 30 – Dec. 2),
Dec. 1, 2004, San Antonio, Texas, USA
http://www.daml.org

NOT INCLUDED in this Slideset: OTHER PRESENTATIONS ON RULES IN TODAY'S SESSION

- SWRL FOL (by Peter Patel-Schneider)
- RuleML incl. SWRL, FOL (by Harold Boley)
- Rei and Security (by Tim Finin)
- Integrating OWL-DL with Rules (by Boris Motik)

- (plus some stuff is pointed-at via URL's, e.g., Outbrief)

Intro

presentation by Mike Dean and Benjamin Grosof

Outline of Rules Plenary Session

~ Time		Presenters
8:00	Intro, ISWC-2004	M. Dean, B. Grosof
8:05	SWRL Update	M. Dean
8:15	SWRL-FOL	P. Patel-Schneider
8:30	RuleML Update incl. SWRL, FOL	H. Boley
8:45	SweetRules Toolset for RuleML/SWRL incl. Demos, Discussion	B. Grosof, M. Dean
9:30	BREAK	
9:40	SweetRules, continued	
10:05	Rei and Security	T. Finin
10:15	Integrating OWL-DL with Rules	B. Motik, B. Grosof
10:25	SWSL and Rules: Update	B. Grosof, D. Martin
10:40	Next Steps in Standardization	B. Grosof, M. Dean
10:45	Additional Discussion	

ISWC News

presentation by Benjamin Grosof and Mike Dean

ISWC-2004 Rules News I

- ISWC-2004 Tutorial (half-day)
  - "Semantic Web Rules with Ontologies, and their E-Business Applications" (by B. Grosof & M. Dean)
    - Core -- KR Languages and Standards
    - Tools -- SweetRules, Jena, cwm, and More
    - Applications -- Policies, Services, and Semantic Integration
  - Quite successful, ~50 attendees.
  - Tutorial Material Is Available Free on Web: http://ebusiness.mit.edu/bgrosof/#RulesTutorial (continuingly updated), or http://www.daml.org/2004/11/tutorial

ISWC-2004 Rules News II

- RuleML-2004, the ISWC-2004 Rules Workshop (full-day)
  - "Rules and Rule Markup Languages for the Semantic Web" (co-chairs G. Antoniou, H. Boley; other organizers M. Dean, B. Grosof, B. Spencer, S. Tabet, G. Wagner)
  - 3rd in series, one held at each ISWC
    - Planned again for next ISWC (2005)
  - Quite successful, ~50 attendees
  - Proceedings Available:
    - Springer-Verlag published volume
    - http://2004.ruleml.org

ISWC-2004 Rules News III

- Themes of Workshop Papers and Invited Talks:
  - RuleML/SWRL, and More
  - Approaches to Combining Rules with Ontologies
    - Use Cases
      - SWRL and Ontology Translation [M. Dean] [C. Golbreich]
    - Theory on Combining DL with Nonmon LP
      - E.g., in Defeasible Logic (similar to Courteous LP)
    - Constraints/FOL [A. Preece *et al.*]

- **Rules also in ISWC-2004 Main-Conference Talks:**
  - E.g., [M. Rousset] invited, [B. Motik & R. Studer] paper

Ongoing Rule Efforts

- RuleML Initiative
- Joint Committee
- SWSL-Rules

- WonderWeb
- REWERSE
  - PPSWR04 Workshop
- WSML
  - FORUM

SWRL Update

presentation by Mike Dean

SWRL

- extends OWL and RuleML
  - sublanguage of RuleML
- several releases
  - SWRL 0.5 (November 2003)
  - SWRL 0.6 (April 2004)
    - added builtins
    - added XML Schema
    - also W3C Member Submission
  - draft SWRL 0.7 (November 2004)

Recent SWRL Updates

- swrlb definitions of builtins
- builtin test cases
- sameIndividualAtom and differentIndividualsAtom now consistently take 2 arguments

SWRL Implementations

- see http://www.daml.org/rules/proposal/implementations
  - Many use "named classes only" subset of SWRL
  - More use SWRL RDF Concrete Syntax than XML Concrete Syntax

SWRL FOL

- focused effort based on feedback from last DAML PI Meeting
- extends SWRL toward First Order Logic
  - does not replace SWRL
- initial language released in early November
- see presentations to follow

SWRL-FOL

presentation by Peter Patel-Schneider

- See separate slideset

RuleML Update incl. SWRL, FOL

presentation by Harold Boley

- See separate slideset

SweetRules V2.0 Overview

presentation by Benjamin Grosof and Mike Dean

Overview of SweetRules V2.0:
Tools for Semantic Web Rules and Ontologies, including Translation, Inferencing, Analysis, and Authoring

by Benjamin Grosof and Mike Dean***

**MIT Sloan School of Management, http://ebusiness.mit.edu/bgrosof*
***BBN Technologies, http://www.daml.org/people/mdean*

Announcing…

- SweetRules V2.0 Initial Release was Monday Nov. 29 2004.
- Open-source on SemWebCentral.org
  - http://sweetrules.projects.semwebcentral.org
- *You're the first to hear* ☺

SweetRules V2 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.
  - Including semantics-preserving translations between different rule languages/systems/families, e.g., Situated LP ↔ production rules

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 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 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
  - RuleML/SWRL
- Proof of concept for feasibility, including of KR algorithms and translations between heterogenous families of rule systems
  - Encourage others: researchers; industry esp. vendors
- Catalyze/nucleate SW Rules communal efforts on:
  - Tools, esp. open-source
  - Application scenarios / use cases, esp. in services

SweetRules Website

- See http://sweetrules.projects.semwebcentral.org
  - Downloadable
  - Open-source code
  - Documentation
    - Javadoc
    - ISWC-2004 Tutorial on Rules+Ontologies+Ebiz
    - Overview, README, Rule Formats, ...

SweetRules Context and Players

- Part of SWEET = "Semantic WEb Enabling Tools" (2001 –)
  - Other parts: … these use SweetRules …
    - SweetDeal for e-contracting
    - SweetPH for Process Handbook ontologies
- Cross-institutional. Collaborators invited!
  - Originated and coordinated by MIT Sloan since 2001
  - Code base: Java, XSLT; convenience shell scripts (for testing drivers)
  - Code by MIT, UMBC, BBN, Stanford, U. Zurich
  - Cooperating other institutions: U. Karlsruhe, IBM, NRC/UNB, SUNY Stonybrook, HP, Sandia Natl. Labs; RuleML Initiative
    - Collaboration on design of code by Stanford, U. Karlsruhe
  - Uses code by IBM, SUNY Stonybrook, Sandia Natl. Labs, HP, Stanford, Helsinki
  - Many more are good targets: subsets of Flora-2, cwm, KAON, JTP, SWI Prolog, Hoolet, Triple, DRS, ROWL, ...

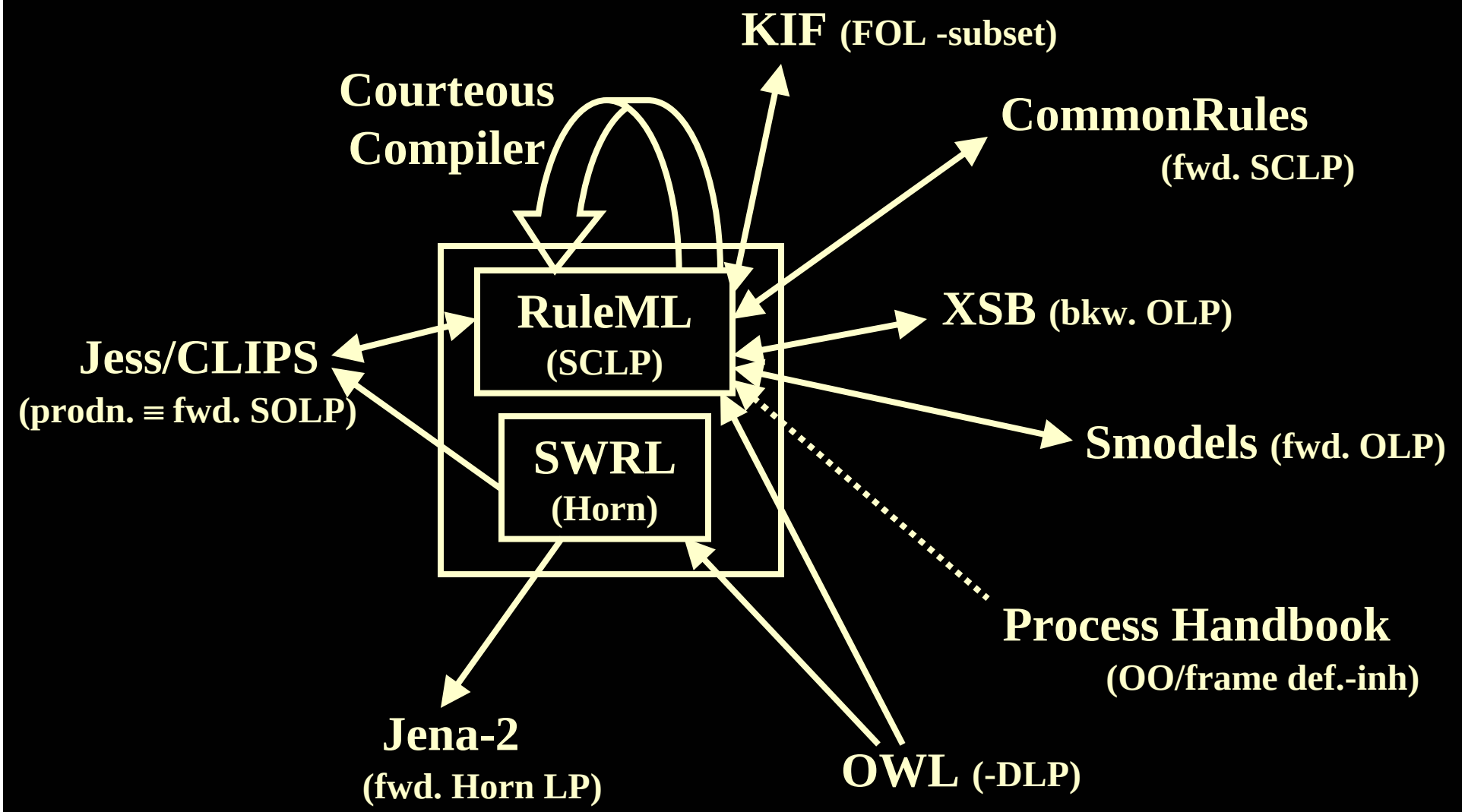
SweetRules V2.0 Fundamental KR Today

- 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 V2.0 KR Languages Supported

- RuleML (SCLP)
- SWRL rules (named-classes-only)
- OWL
  - Esp. Description Logic Programs subset
- Prolog (pure, plus informational built-ins) – bkw. OLP
  - XSB
- Production Rules -- fwd. ~ SOLP
  - Jess/CLIPS; Jena
- Other:
  - KIF (FOL subset), IBM CommonRules (fwd. SCLP), Smodels (fwd. Prolog)
  - *Soon to be integrated:* Process Handbook (OO/frame ontologies with default inheritance)

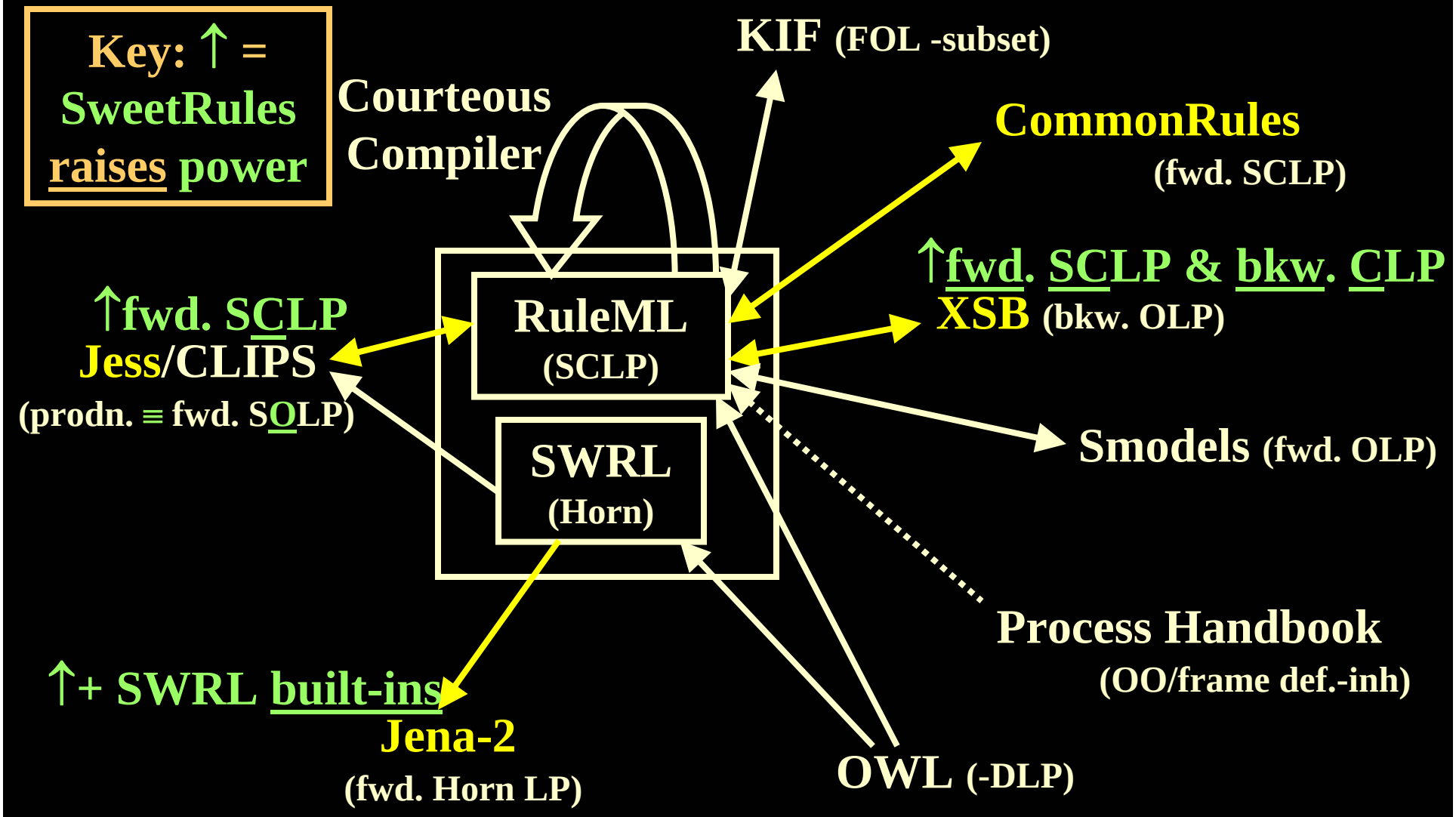
SweetRules Today: Translators Graph



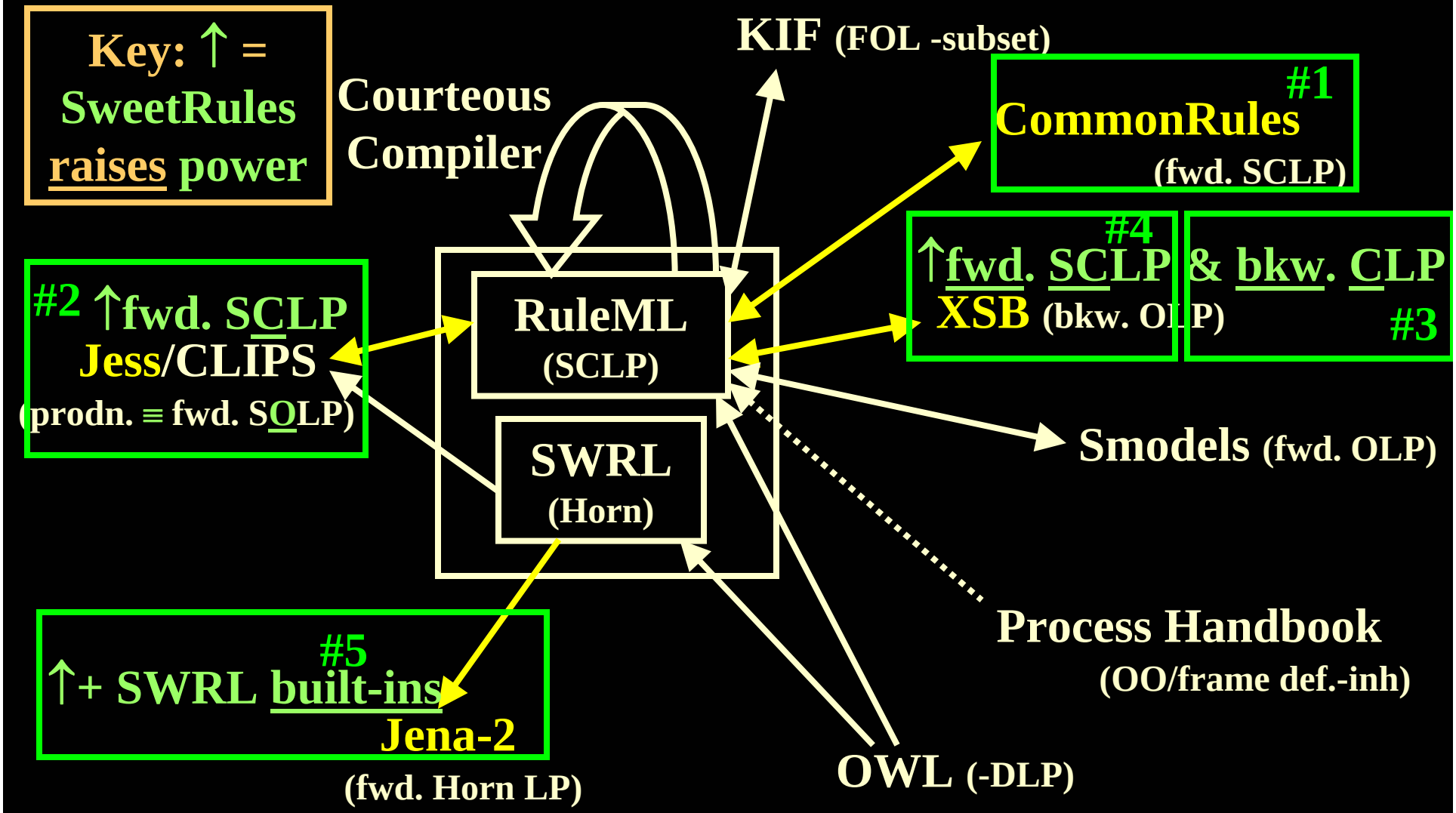
SweetRules Inferencing Capabilities Today: Overview

- 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: Indirect Inferencing Engines



SweetRules V2.0 New Inferencing Engines



SweetRules Capabilities Today Cont.'d

- Authoring and Testing front-end: *currently less mature, more partial*
  - Command-line UI
    - *Future: 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/RDF markup syntax.
  - Protégé OWL Plug-in Enhancement
    - SWRL Rule Editor (separate component from SweetRules)
- Analyzers incl. Validators: *currently less mature, more partial*
  - Detect violations of expressive restrictions, e.g., required syntax
  - Misc. other kinds of analyzers
    - e.g., DiffFacts for incremental reasoning
  - Some validators & analyzers as part of various translator & inferencing components
    - e.g., in SweetOnto, SweetXSB, SweetJess

SweetRules Components Today

- Some components have distinct names (for packaging or historical reasons): E.g.,
  - SweetCR translation & inferencing RuleML ↔ CommonRules
  - SweetXSB translation & inferencing RuleML ↔ XSB
  - SweetJess translation & inferencing RuleML ↔ Jess
  - SweetOnto translation {RuleML, SWRL} ← OWL + RDF-facts
  - SweetJena translation & inferencing SWRL → Jena-2
- Other Project Components: (separate codebases for licensing or other reasons)
  - SWRL Built-Ins library *Currently:* for Jena-2
  - SweetPH translation RuleML ← Process Handbook (OO/frame ontologies)
    - *Currently V1.2 is running. Separately downloadable V2 is in progress.*
  - Protégé OWL Plug-in authoring SWRL rules (Horn, referencing OWL)
    - Enhancement providing SWRL Rules authoring is part of the Plug-In.
  - SWRL Validator

Novel NAF Capability in Production Rules I

- Newly Supports Correct Negation-As-Failure in Production Rules
  - Problem: Jess does not correctly implement Negation-As-Failure
    - Conjecture: this problem is shared by all current production rule systems (OPS5-heritage family, based on Rete)
      - *Currently investigating this conjecture.*
  - Solution: We have developed two new techniques with associated KR proof/model theory
    - Stratified case of NAF: declare stratification-based salience in the production rules, when translating from RuleML
      - *Is implemented in SweetRules V2.0 (SweetJess component). Works correctly in all initial phase tests. More testing is in progress.*

Novel NAF Capability in Production Rules II

- General non-stratified case of NAF: new bottom-up algorithm for well founded semantics of OLP
  - *Currently detailed algorithm has been designed and is being implemented.*

- Observation on Additional Value-add: This eliminates the need for agenda meta-rules hacking to get NAF right in production rules, which is frequent in existing production rule applications (and is part of training/methodology)
  - *Interesting Question: How big a percentage of overall agenda meta-rules in typical applications are thus eliminated? Most?*

More Novel Capabilities

- 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., XSB Prolog, Jess, Smodels
  - Uses Courteous Compiler component from IBM CommonRules
- New Include-a-KB mechanism, similar to owl:imports Has 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.

Additional Firsts in Implementation

- SWRL/RuleML Built-Ins: (which are based largely on XML-Schema operations)
  - In SweetJena *(in progress: also in rest of SweetRules)*
- Forward Situated Courteous LP inferencing+action with intrinsically highly scaleable run-time performance
    - Both XSB/Prolog and Jess/Rete/production-rules reportedly scale very well to very large rulebases (~100K+ non-fact rules, many Millions facts)
  - Restrictions: Stratified NAF, function-free
  - SweetXSB forward-direction engine
    - Uses Query-All-Predicates, Action Launcher techniques
    - *Currently:* Restriction from XSB: sensing limited to built-ins
  - SweetJess engine
    - *Currently:* Restriction from Jess: all-bound-sensors (includes built-ins)
- Backward Courteous LP inferencing for general non-stratified NAF, and scaleably in above sense
  - SweetXSB backward-direction engine
    - *Currently:* Restriction from XSB: sensing limited to built-ins

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

Novel Integration Framework

- Pluggability & Composition Framework Architecture with detailed interfaces
  - Add your own translator/inferencing-engine/authoring/testing tools
    - We've used this to integrate previous existing translators, and some of our new translators
      - Found it to be easy! How about you?
  - Compose tools automatically, e.g.:
    - translator1 ⊗ translator2
    - translator ⊗ inferencing-engine
  - Search for tools

Object Models for Rules/Ontologies

- SweetRules uses popular API's & Tools Underneath to manipulate SW markup object models

API/Tool	Kind of Object Model
Jena	OWL, RDF
Protégé (API)	SWRL -RDF
JAXB	RuleML/SWRL -XML
XSLT	RuleML/SWRL -XML

E.g., the predicate-dependency graph and stratifier for SweetJess NAF handling was easily built out of the JAXB object model.

Measuring Power, Elegance and Reuse

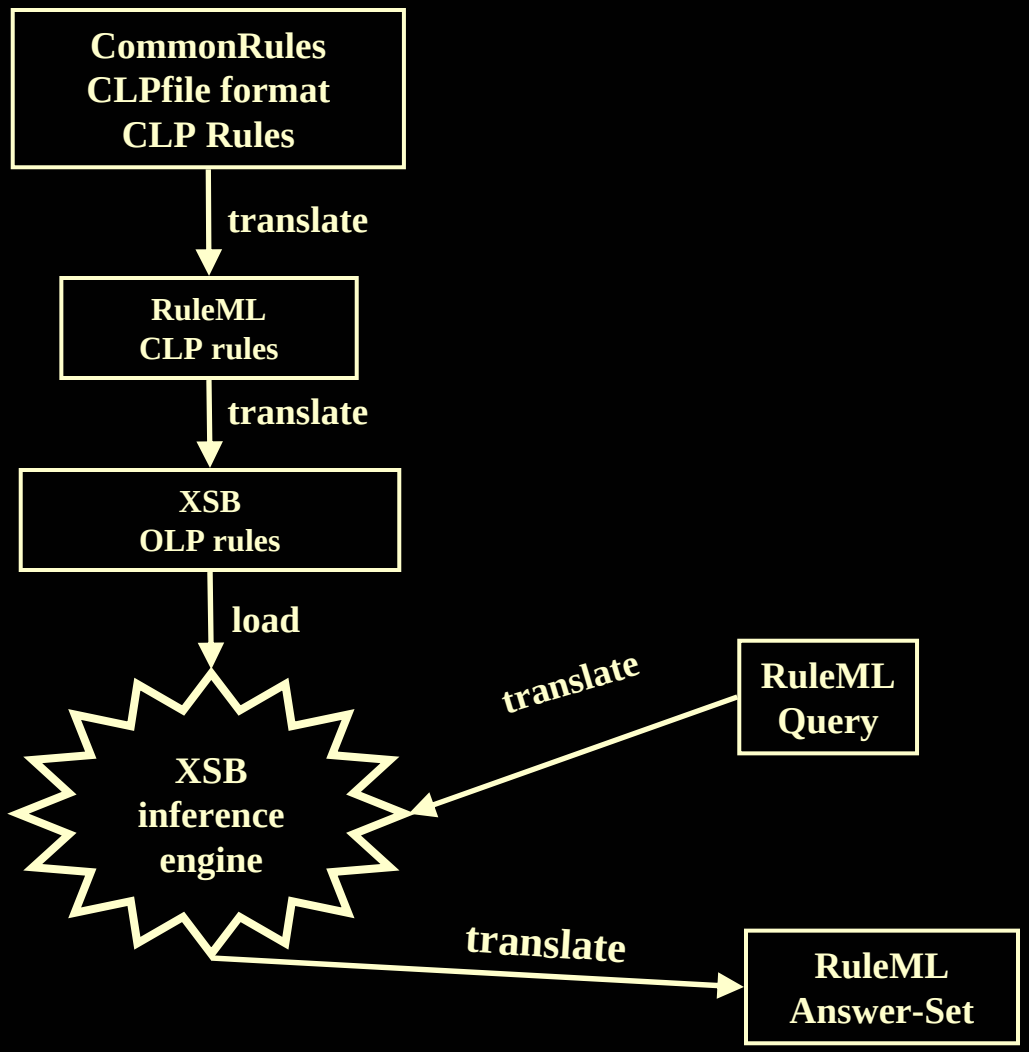
- Significant increases in KR expressiveness of (semantically correct) translation and inferencing relative to previous tools/approaches
  - Production rules join the party of SW and interoperability
  - Correct negation/nonmonotonicity in production rules without extensive agenda meta-rules hacking
  - Courteous extensions of commercial-grade inferencing engines for Prolog and production rules
- Significant increases in scaleability of forward and backward inferencing for (S)CLP
- Weighted coverage: Support the commercially most important kinds of rule systems (production rules, Prolog) for both translation and inferencing

- 10+ diverse KR languages/systems/formats supported
  - Half pre-SW, Half SW
- 20 simple translators; + composite translators
- 5 indirect inferencing engines
- All in code base of 23K Lines Of Code, built mostly in 6 months.
  - MUCH less than the total size of the interoperated systems

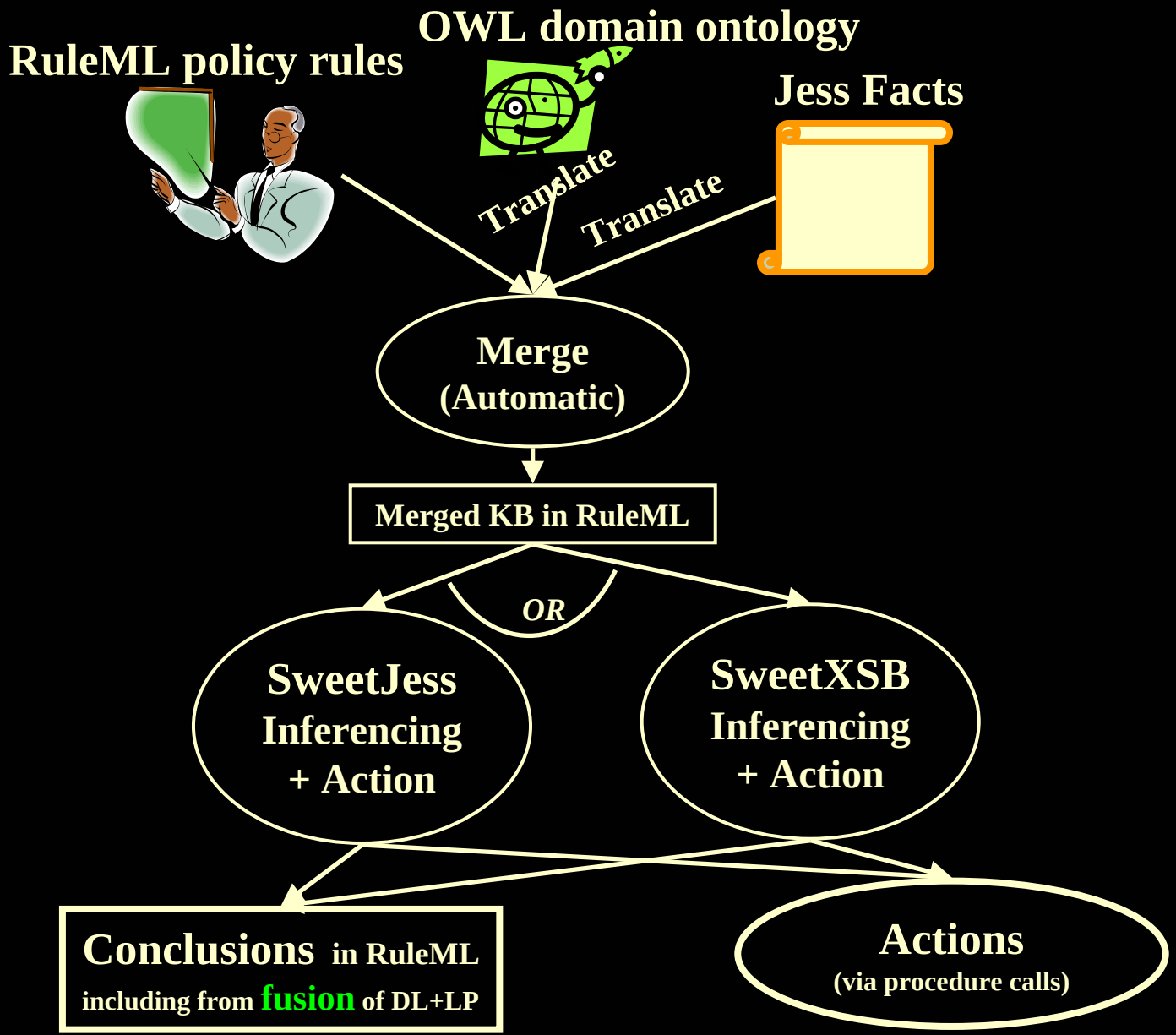
SweetRules V2 Demo Outline

- Pacifism (Quakers and Republicans)
  - Translation and CLP inferencing
  - SweetCR, SweetXSB backward (with RuleML answersets)
- Ordering Lead Time (e-commerce policies and notification)
  - KB Merging
  - Hybrid reasoning combining SCLP rules with DLP OWL ontologies
  - Effecting (actions)
  - SweetOnto, SweetJess, SweetXSB forward
- Search and compose translators within SweetRules repository
- Genealogy (family relationships, e.g., uncle-of)
  - Hybrid reasoning combining SWRL rules with DLP OWL ontologies, plus SWRL/RuleML built-ins and Protégé-created SWRL rules
  - SweetJena, Protégé SWRL editor, SWRL builtins, SweetOnto
- SweetDeal E-Contracting Application using SweetRules (supply chain)
  - SCLP RuleML rules that include DLP OWL ontologies

Quaker Example Demo Flow



OrderingLeadTime Example Demo Flow



SweetDeal V2 Demo Outline

- SweetDeal E-Contracting Application using SweetRules (supply chain)
  - SCLP RuleML that include DLP 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

SweetRules V2 Demo Examples

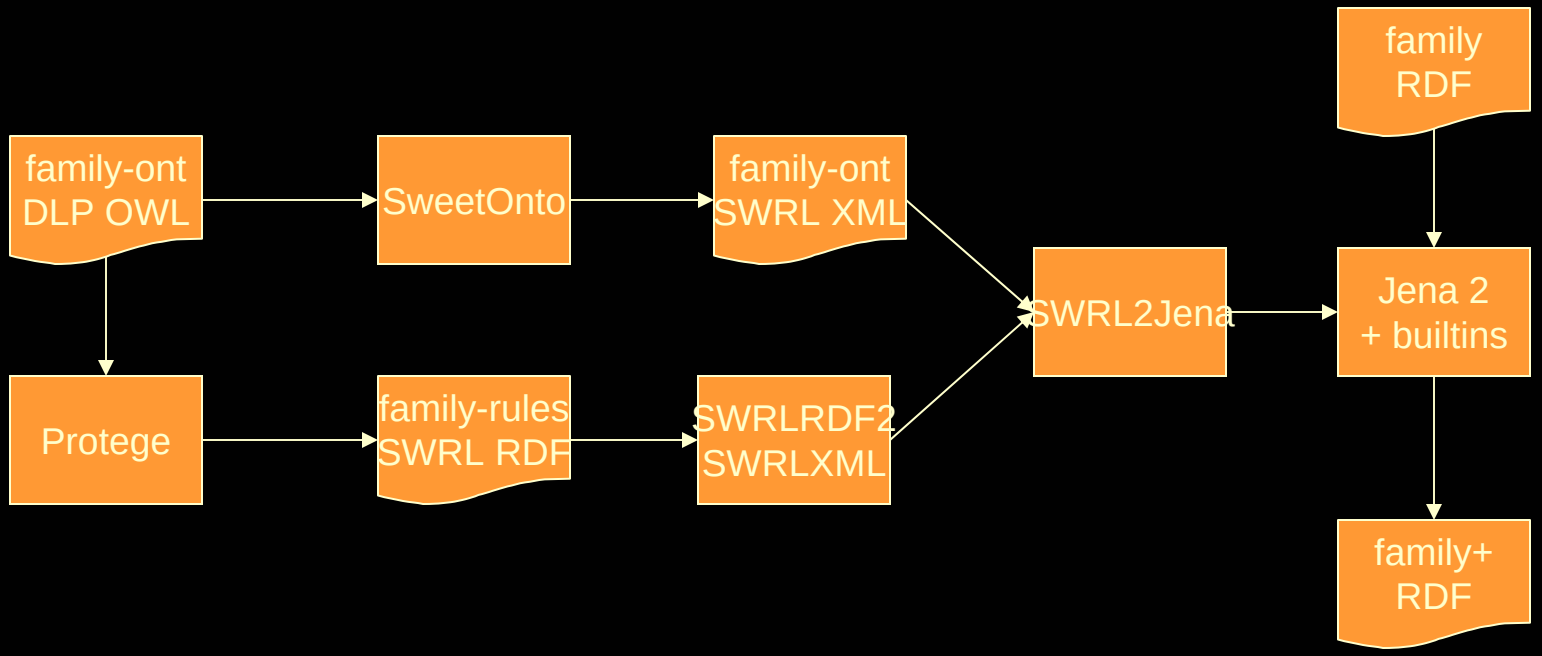
- See separate SweetRules V2 demo examples material.

SWRL-y SweetRules V2 Demo by Mike Dean

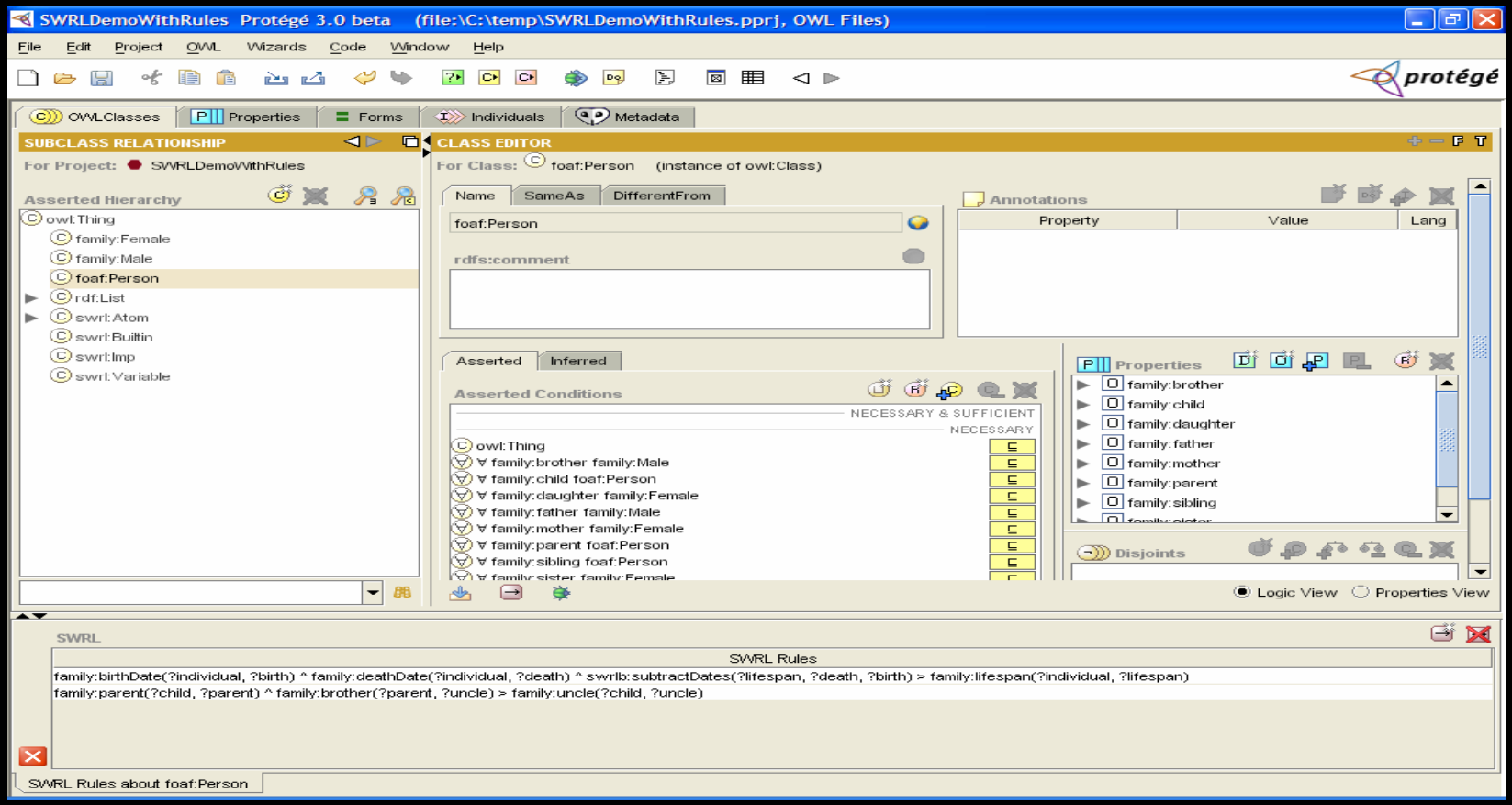
SLIDES FOLLOW

- And also see separate SweetRules V2 demo examples material.

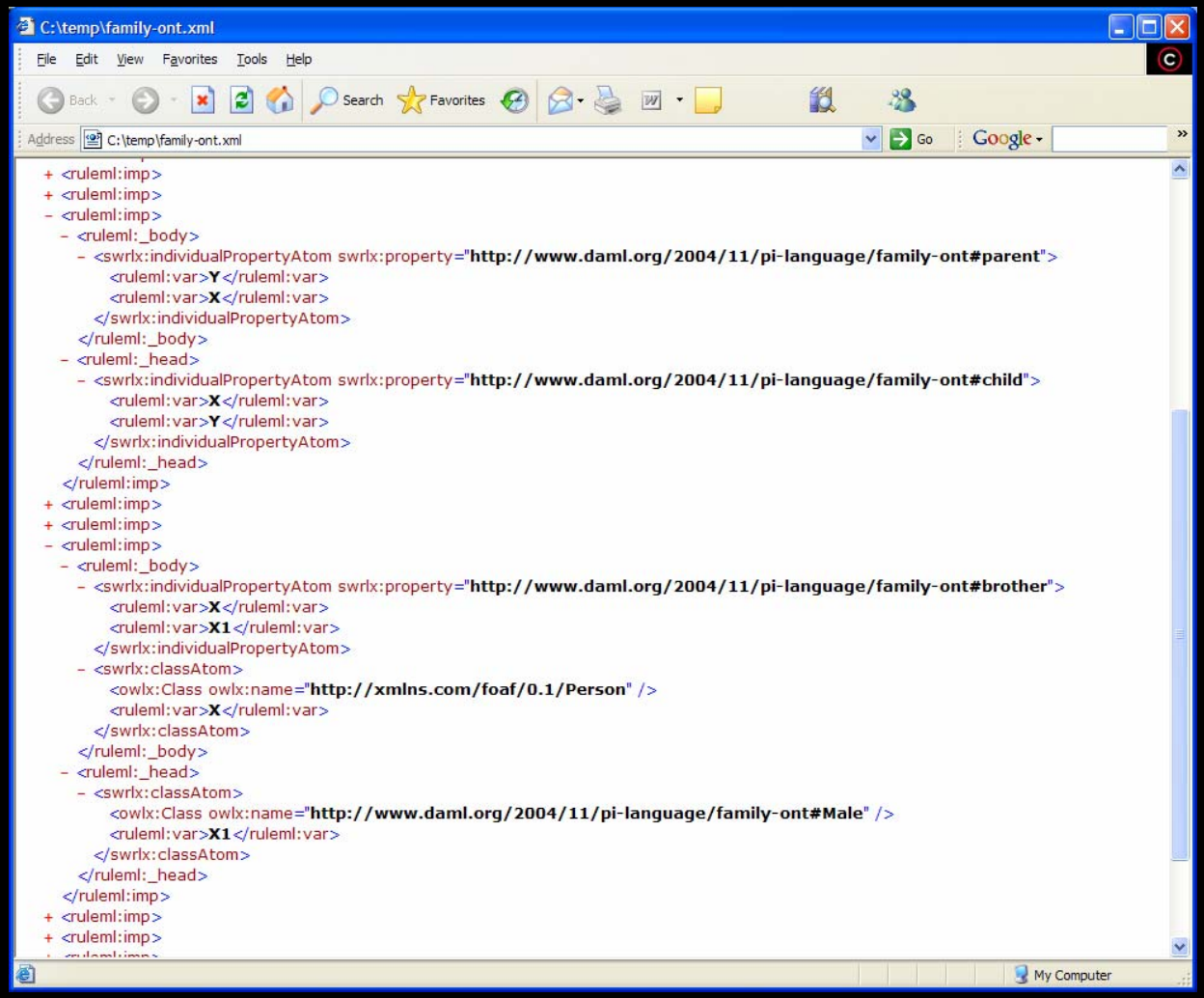
Protégé/SWRL/Jena Demo



Protégé Ontology and Rules



family-ont rules from SweetOnto



family

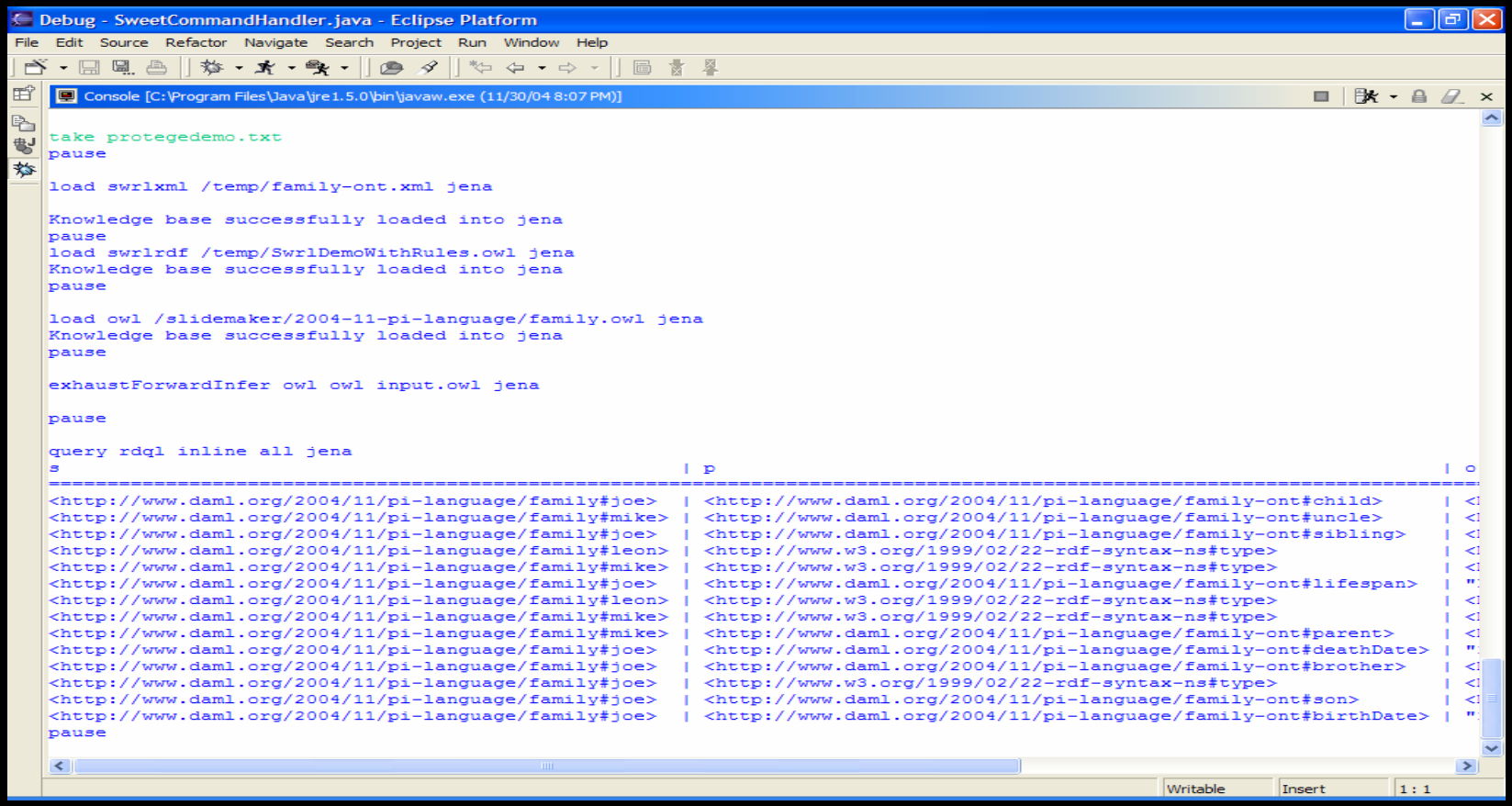
```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
        <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>

<rdf:RDF
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:family="http://www.daml.org/2004/11/pi-language/family-ont#"
  xml:base="http://www.daml.org/2004/11/pi-language/family">

<foaf:Person rdf:ID="joe">
  <family:birthDate rdf:datatype="&xsd;date">1923-10-23</family:birthDate>
  <family:deathDate rdf:datatype="&xsd;date">1999-03-17</family:deathDate>
  <family:son rdf:resource="#mike"/>
  <family:brother rdf:resource="#leon"/>
</foaf:Person>

</rdf:RDF>
```

SweetRules Execution



family+

family:joe	rdf:type	foaf:Person	
family:joe	family-ont:birthDate	"1923-10-23"^^xsd:date	
family:joe	family-ont:deathDate	"1999-03-17"^^xsd:date	
family:joe	family-ont:son	family:mike	
family:joe	family-ont:brother	family:leon	
family:joe	family-ont:child	family:mike	superproperty
family:joe	family-ont:sibling	family:leon	superproperty
family:joe	family-ont:lifespan	"P27539D"^^xsd:duration	rule
family:mike	rdf:type	family-ont:Male	allValuesFrom
family:mike	rdf:type	foaf:Person	allValuesFrom
family:mike	family-ont:parent	family:joe	inverse
family:mike	family-ont:uncle	family:leon	rule
family:leon	rdf:type	family-ont:Male	allValuesFrom
family:leon	rdf:type	foaf:Person	allValuesFrom

Demonstrated

- Hybrid reasoning with ontologies and rules
- SWRL editing with Protégé
- Transparent chained SweetRules translation
  - OWL DLP to SWRL
  - SWRL RDF to SWRL XML
  - SWRL XML to Jena 2
- Rule execution using Jena 2 with builtins

SweetDeal V2 Demo: Novelty Highlights

1. SweetDeal is the first e-contracting application scenario, and first real e-business application scenario, combining RuleML with OWL. It uses DLP-fusion combining the OWL with RuleML to do combined hybrid inferencing. It combines contract rulesets in RuleML with business process/contract ontologies in OWL.
2. Moreover, SweetDeal is the first to have such contracts contain rules that employ procedural attachments to perform actions (side-effectful) as part of the business processes that the contracts specify.
3. SweetDeal is the first previous application to be refitted to use SweetRules V2 – and the first to be refitted to use DLP-fusion.

- Deltas wrt the previous SweetDeal V1 prototype (of 2002):
  - Uses OWL (previous DAML+OIL); DLP-fusion; procedural attachments for actions; SweetRules as infrastructure

SweetRules: Use Cases Overview

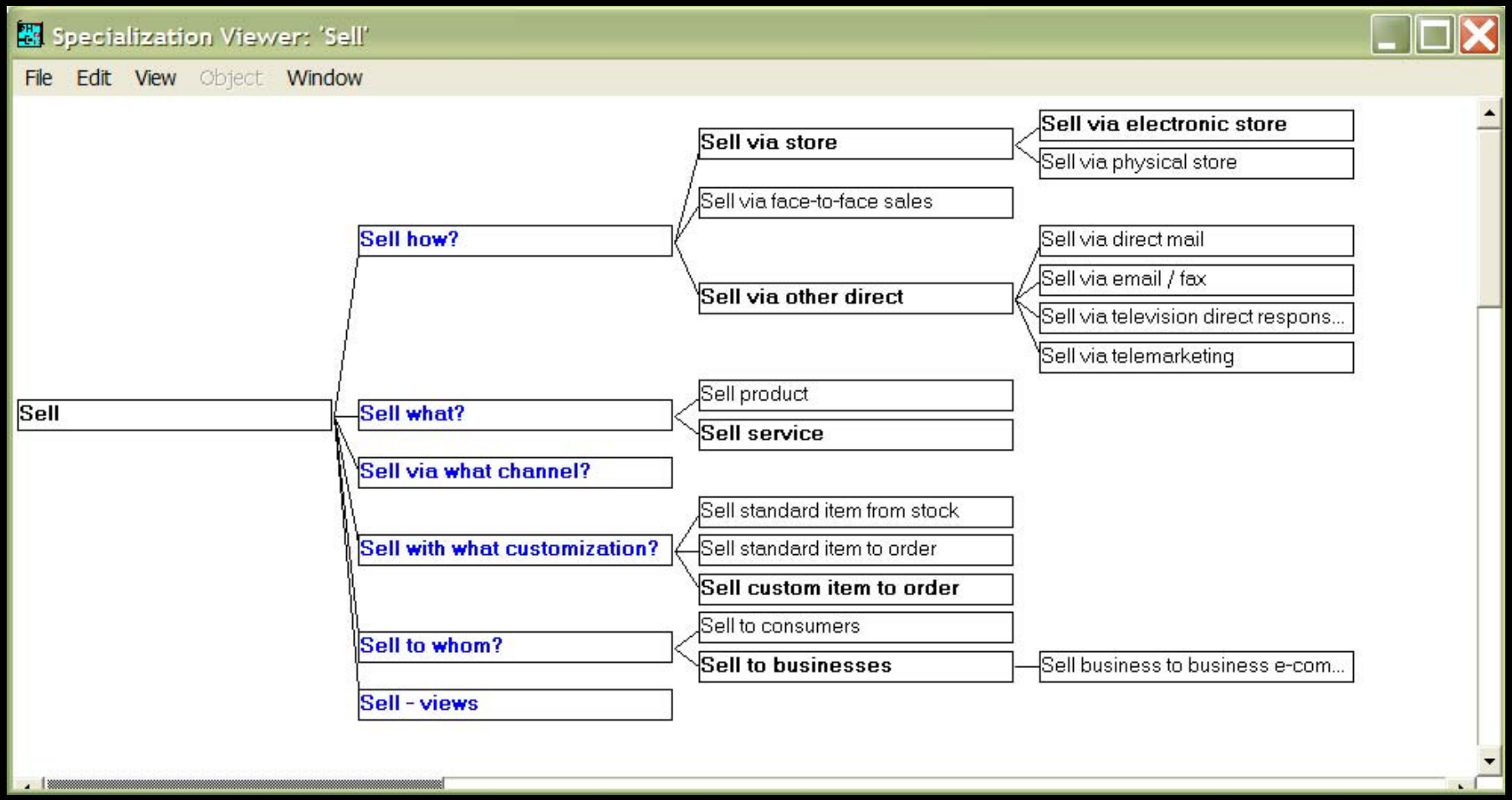
- Trust Policies: authorization, privacy, security, access control
  - E.g., financial services, health care
  - Extensive analysis of business case/value
- Semantic mediation: rule-based ontology translation, context-based information integration
- Contracts/negotiation, advertising/discovery
  - E-procurement,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

Opportunity for Process Handbook in SWS

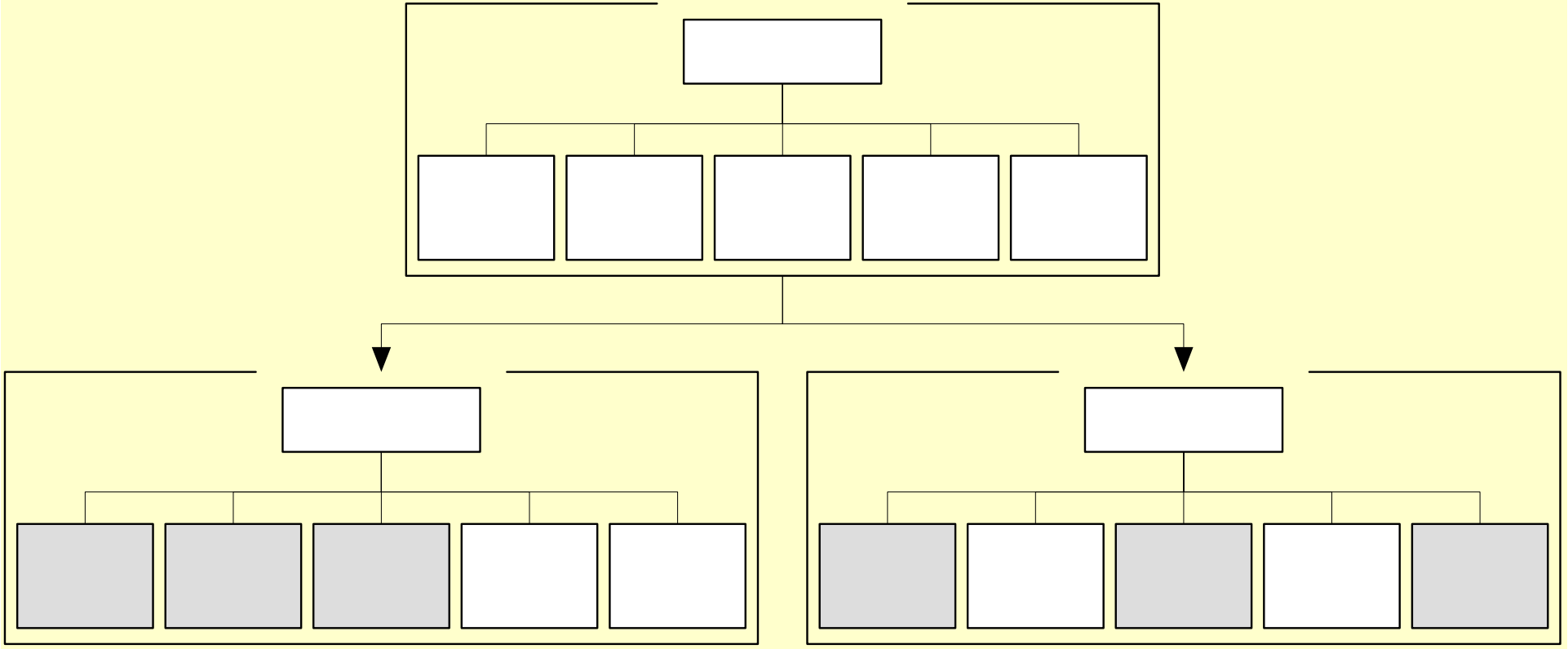
- Need for Shared Knowledge Bases about Web Services / Business Processes
  - For Semantic Web Services, etc.
- Want to leverage legacy process knowledge content
  - Go where the knowledge already is

- Process Handbook (PH) as candidate nucleus for shared business process ontology for SWS
  - 5000+ business processes, + associated class/property concepts, as structured knowledge (http://ccs.mit.edu/ph)
  - E.g., used in SweetDeal E-Contracting prototype
- Concept: Use Semantic Web KR and standards to represent Object-Oriented framework knowledge:
  - class hierarchy, types, generalization-specialization, domain & range, properties/methods' association with classes

Some Specializations of "Sell" in the Process Handbook (PH)



PH Example: Selling Processes



An activity (e.g., SellProduct) has sub-activities (steps).

Its specializations (e.g., SellByMailOrder) inherit its sub-activities by default.

Key: gray = modified (overridden). X = deleted (canceled).

ent

poten

SweetPH's New Technical Approach: Courteous Inheritance for PH & OO

- Surprise: use SW *rule* language not the main SW *ontology* language! I.e., use (SCLP) RuleML not OWL.
  - OO inheritance is *default* ⇒ more reuse in ontologies
  - OWL/FOL cannot represent default inheritance
  - RuleML/nonmon-LP can
- Courteous Inheritance approach translates PH to SCLP KR
  - A few dozen background axioms. Linear-size translation. Inferencing is tractable computationally.
- PH becomes a SWS OO process ontology repository
- *In progress: open source version of PH content*
- *In progress: extend approach to OO ontologies generally*

SweetRules: Plans within DAML program

- Polishing, generally, of doc and code

- SweetPH release
- Non-stratified NAF (WFS) in SweetJess
- More tightly integrate SWRL with RuleML: spec, code

- More application scenarios, esp. services
  - Policies, contracts, mediation, …

SweetRules: Directions *beyond DAML program*

- Hook up to Web Services
  - Importing knowledge bases / modules, procedural attachments, translation/inferencing, events, …
- More on authoring, UI, editors
- Support increased expressiveness of DLP
  - *Later in session:* new theory; services uses
- Support more rule/ontology engines/systems:
  - Tasks: translation, inferencing
  - Flora, cwm, Triple, Hoolet, DRS, ROWL, KAON, JTP, SWI Prolog, …
  - Systems of new/various kinds: ECA, RDF-Query/XQuery, …

SweetRules: Directions beyond the DAML program, cont.'d

- More support of SWSL-Rules, incl. for Hilog, frame syntax features
- More support of FOL
  - FOL RuleML / SWRL FOL / KIF / SCL
- More conflict analysis
- Incremental reasoning, events
- Scaleability performance testing/benchmarking

- *More Collaborators invited!*

SweetRules V2 Team

- Core Team:
  - B. Grosof (MIT Sloan), M. Dean (BBN), S. Ganjugunte (UMBC student), S. Tabet (MIT Sloan), C. Neogy (MIT Sloan)

- Project Lead: B. Grosof. Project Co-Lead: M. Dean.
- Lead designer of core including SCLP RuleML and DLP OWL aspects: B. Grosof
- Lead implementer of core: S. Ganjugunte
- Lead designer and implementer of SweetJena & several SWRL tools: M. Dean
- Lead implementer of SWRL built-ins: D. Kolas (BBN)
- Lead designers of Protégé Rules Editor enhancement: M. Musen (Stanford), M. O'Connor (Stanford); Project Lead: M. Musen; Lead Implementer: M. O'Connor.
- Lead designers of SweetPH: B. Grosof, A. Bernstein (U. Zurich)
- Lead implementer of SweetPH: A. Bernstein
- Lead designer of SweetDeal application scenario prototype: B. Grosof
- Lead implementer of SweetDeal: S. Bhansali (MIT Sloan student)

- Other Contributors: B. Motik (U. Karlsruhe student), R. Studer (U. Karlsruhe), R. Volz (U. Karlsruhe student); T. Finin (UMBC), A. Joshi (UMBC); J. Bonin (U. Zurich student); T. Poon (MIT student); H. Chan (IBM); H. Boley (NRC/UNB)
- * (This is a preliminary list, we may have forgotten to include someone; if so, apologies!)

Rei and Security

presentation by Tim Finin

- See separate slideset

Context Overview on: Integrating OWL-DL with Rule-based Systems

presentation by Benjamin Grosof

Directions in Extending DLP I

- DLP1 = the KR in the original DLP paper [Grosof, Horrocks, Volz, & Decker WWW-2003]. (DLP = Description Logic Programs)
- DLP1 translator (OWL → RuleML) implemented in SweetOnto (successor to KAON DLP component), tightly integrated within SweetRules V2.0.

- There are known extensions to handle:
  - Disjunction when inessential
  - Existentials via skolemization (in head): e.g., someValuesFrom in superclass of inclusion axiom
  - Equality (in head): e.g., sameIndividualAs
  - Integrity constraints: e.g., disjoint classes
  - These extensions haven't been packaged up yet in easy-to-digest form. They're in papers/theses/experimental-prototypes by {Grosof, Horrocks, Volz, Decker, Motik}.

Directions in Extending DLP II

- *But, actually, there's more to the story…*

- Further significant expressive extensions are available now from two directions of recent KR model/proof theory:
  1. DL ↔ Horn/Disjunctive LP: Results by B. Motik & R. Studer (see sub-section presentation on Integrating OWL-DL with Rule-based Systems)
  2. Ordinary/Courteous LP ↔ FOL: Results by B. Grosof (see later sub-section presentation on SWSL and Rules)

- Would be nice to have clearer picture of a family of one or more extended DLPs be available as well-understood theory -- and communal terminology -- in 2005.
  - *In-Progress:* MIT Sloan & U. Karlsruhe formulating collaboration

Integrating OWL-DL with Rule-based Systems

presentation by Boris Motik and Rudi Studer

- See separate slideset

SWSL and Rules

presentation by Benjamin Grosof and David Martin

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)

SWSL Strategy *[repeat from Services presentation]*

- Build out from OWL-S
  - to take advantage of more expressive languages
  - to extend the conceptual model
- Full-fledged use of FOL expressiveness
  - OWL-S can use SWRL and SWRL FOL in quoted contexts, in service descriptions (instances)
  - SWSL will use it throughout; both in ontology axioms and in all parts of service descriptions
- Leverage broad availability of LP-based languages, environments, tools, etc.
- Build on mature conceptual models
  - PSL, W3C architecture, Dublin core
- Maintain connections with the world of OWL
  - Layers of expressiveness

SWSL Components *[repeat from Services presentation]*

- Conceptual Model
  - Build on OWL-S, PSL, [W3C WS Architecture]
- Language
  - SWSL Rules – LP with NAF; Courteous, Hilog extensions
  - SWSL FOL
  - Shared presentation syntax; builds on F-Logic
  - Markup syntax – TBD probably with RuleML committee
- Ontology
  - Formal expression of conceptual model
  - Both in SWSL FOL and LP (as much as possible)
- Bridge (?)
  - What can we provide to enable coordinated use of FOL and LP reasoners
- Grounding
  - Like OWL-S Grounding, connects with WSDL

Technical Requirements for SWSL-Rules

- Presentation syntax (rather than markup) needed most urgently
  - To create and communicate examples to drive SWSI design
- Strong Consensus: Need Nonmonotonic LP. And FOL.
  - "SWSL-Rules" = the LP KR.
  - "SWSL-FOL" = the FOL KR.
- Expressive Features for SWSL are similar to those desired for SW rules in general, but with bit different near-term importance/urgency:
  - Important in both: Prioritization, NAF (cf. Courteous LP)
  - Important in both, more urgent in SWS than SW overall: Meta-power/convenience: Hilog, frame syntax (cf. F-Logic)
  - A bit more important in SWS than SW overall: Lloyd-Topor
  - Less important: triggering of side-effectful actions (cf. Situated LP effecting or Transaction Logic)

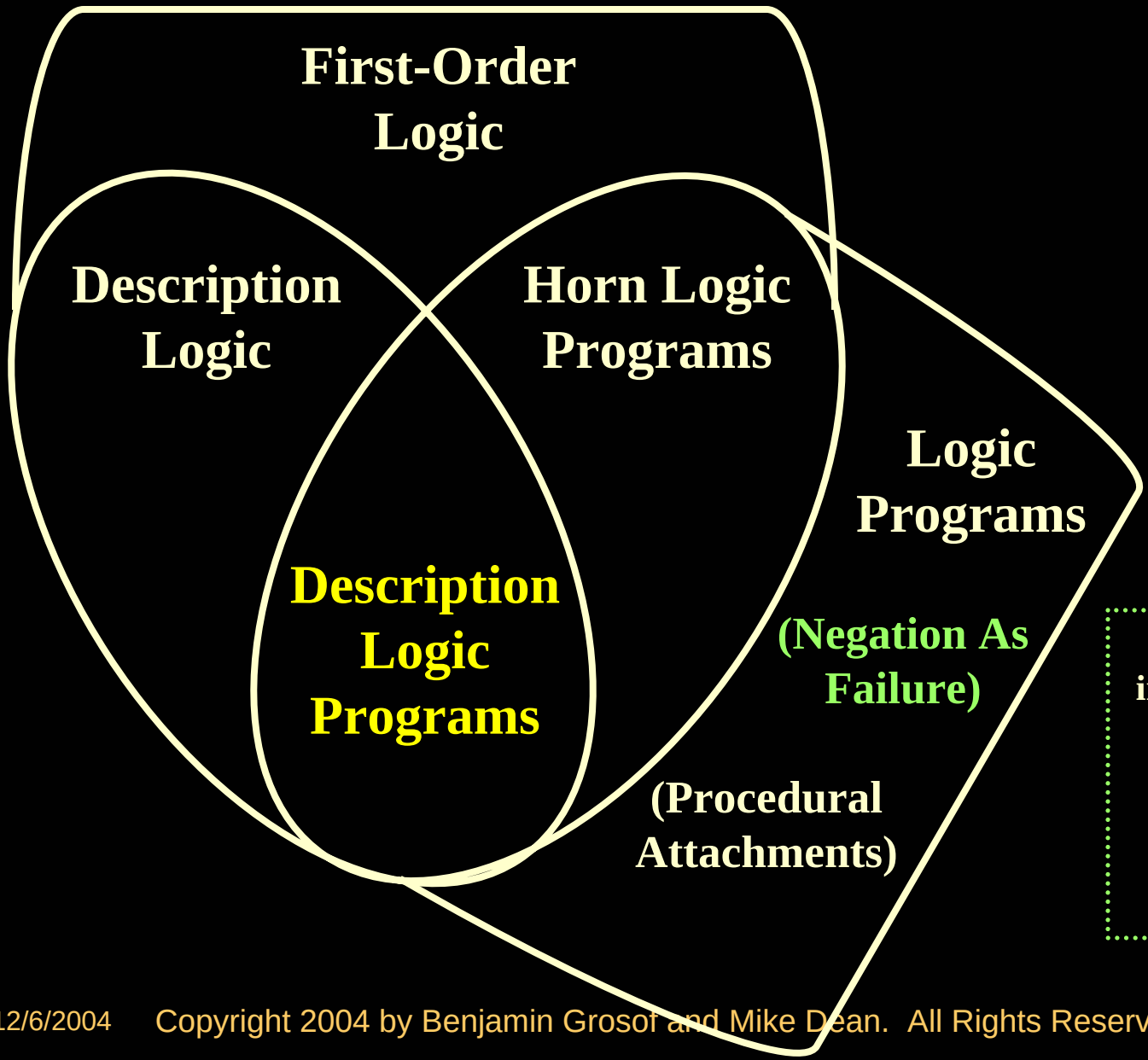
Markup Language Plan for SWSL-Rules I

- RuleML is the only serious candidate on the table for SWSL-Rules
  - Webized nonmon LP; some other key features
- SWRL does not meet basic requirements for SWSL-Rules
  - E.g., lacks nonmon
- CLP RuleML meets basic requirements for SWSL-Rules
- FOL RuleML meets basic requirements for SWSL-FOL
  - Unclear yet whether SWRL FOL is enough
    - E.g., result functions in situation calculus, extensibilty to predicates being terms in Hilog / frame syntax
- Nice match: FOL & Nonmon LP already in RuleML, as in SWSL
  - Full SWSL-Rules expressiveness would become extension of current SCLP RuleML, likewise full SWSL-FOL would become extension of current FOL RuleML
  - "A Package Deal" for {SWSL-Rules & SWSL-FOL}
  - Retains 90% Syntax Overlap
- Simplified Common Logic is another candidate for SWSL-FOL

Challenge for SWSL: Bridge LP & FOL

- Currently, SWSL is like a Butterfly:
  - 2 Beautiful Wings:
    - {LP;Policies;Trust etc.}
    - {FOL; Process Models; Composition etc.}
  - …Connected by only a thin fuzzy body:
    - Horn LP intersection KR
- New fundamental KR theory is needed to unify nonmon LP with FOL
  - A holy grail for SWS, and for SW generally
- *In-Progress:* Enhancements to DLP, e.g., Motik, Studer, Grosof, Horrocks
- *In-Progress:* New Approach: Hypermonotonic reasoning
  - Being discussed in SWSL (& presented at PPWSR04) [Grosof]
  - Theorem: Courteous/Ordinary LP is sound but incomplete relative to FOL, under simple translation mapping.
    - Reduce NAF-ful Courteous LP $\Rightarrow$ NAF-free Courteous LP $\Rightarrow$ FOL clauses.
  - Incompleteness often desirable if there's inconsistency, acceptable when not.
  - Provides basis for identifying new cases of consistent or monotonic KB fusion. Import/export premises/conclusions between KR's. Example: Rei rules.

Venn Diagram: Expressive Overlaps among KR's



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

Hypermonotonic Reasoning: Overview

- Definition: A KR S is "hyper"monotonic relative to FOL when S is nonmonotonic and S is sound but incomplete relative to FOL.
  - Premises (conclusions) of S are *viewable as premises (conclusions) of FOL.
  - Generalization: *Under a mapping T from premises/conclusions of S to premises/conclusions of FOL.
- The hypermon KR's entailed conclusions can be viewed as always unobjectionable, i.e., sanctioned, by FOL which provides a background "reference" semantics for the premises in the hypermon KR.

Hypermon: Discussion of Definition

- The spirit of conflict handling is a good match to the hypermon concept.
  - 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.
    - The hypermon KR can be viewed as a semantically characterized class of incomplete FOL reasoning tools.

- Analogy: jumping through hyperspace (similar to "hyper"text)
  - Overcomes the apparent barrier/limitation of how inconsistency behaves (global fragilility/propagation) in classical logic. "Tunnels through a wormhole" to a consistent, typically contentful, set of conclusions (with localized propagation scope for unresolved conflicts).

Nonmon LP as Hypermon

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

- 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.

- Theorem: NAF-free Courteous LP ("CLP2") is hypermon.
  - (Some other nonmon KR's are too.)

- Theorem: NAF-ful Courteous LP, and thus Ordinary LP, is hypermon under a simple mapping T1:
  - Replace every NAF'd atom ~p(t) by fp(t), where fp is a new predicate.
  - Add the two rules:
    a. $fp(t) \leftarrow .$
    b. $\neg fp(t) \leftarrow p(t).$

Nonmon LP as Hypermon, cont.'d

- 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 LP premises
      - E.g., as initial rules or as dynamically sensed facts
  - Export (nonmon) LP conclusions as facts to become FOL axioms
    - An early usage: provide KR semantic analysis of Rei as CLP rules conservatively extending (non-Horn-expressible) DL.

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.
    - E.g., $\exists x.q(x)$ in FOL is consistent with CLP in which all rules with q in head mention q positively. E.g., Rei rules consistent with the ontologies it uses.
  - Identify, tweak, extend, design hypermon KR's

SWSL 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)

- SWSL spec. of Rules, FOL presentation syntax, expressiveness
- Handoff issue on markup syntax: ?RuleML, SWRL FOL, SCL?
- Challenge: "Bridging" Nonmon LP with FOL. As weakening?

Next Steps in Standardization

presentation by Benjamin Grosof and Mike Dean

Standardization Routes for Rules

- W3C
  - Exploring possible Workshop on Rules 2Q2005, possibly followed by formation of Working Group
  - Rules apply beyond Semantic Web Activity, e.g., services, policy
- Oasis
  - Lower threshold and lead time than W3C to form Technical Committee
  - Exploring possible TC in Rules
  - Very interested in Rules & RuleML incl. for policies
  - Rules apply to several existing activity areas, e.g., services, policies
- OMG
  - Has Production Rules activity
  - Meta-model focus, complementary to above markup and semantics
    - Very interested in RuleML incl. for markup
- ISO
  - Has FOL activity (Simplified Common Logic, successor to KIF)

END PLENARY SESSION PRESENTATION

presentation by Benjamin Grosof and Mike Dean

Rules Working Group

Discussion Agenda Topics Overview

presentation by Benjamin Grosof and Mike Dean

Agenda Topics I

- Directions for Extending DLP
- SWSL and Rules: Update, Handoff, Bridging
- Next Steps in Standardization for Rules
- Feedback on SweetRules design and directions
  - Features, pluggability/composition
- Feedback on SWRL and RuleML, generally
  - E.g., include-a-kb design
- Feedback on FOL. E.g., adequacy of SWRL-FOL subsetfeatures
- Ideas on use cases and application scenarios
- Implementation plans by all, generally
- Planning for rules tools efforts beyond May 2005

SWSL 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)

- SWSL spec. of Rules, FOL presentation syntax, expressiveness
- Handoff issue on markup syntax: ?RuleML, SWRL FOL, SCL?
- Challenge: "Bridging" Nonmon LP with FOL. As weakening?

Rules WG Outbrief

presentation by Benjamin Grosof and Mike Dean

- See http://www.daml.org/2004/11/pi-rules-outbrief/Overview.html

