DAML
Tools for Rules
Next-Phase Plan

Benjamin Grosof
MIT Sloan School of Management,
http://ebusiness.mit.edu/bgrosof

Presented at DAML PI Mtg., May 25, 2004, New York City
WWW-2004 DevDay last week

• Way cool!
• Lots of tools and use cases now!

• Themes:
  – Mostly LP/RuleML expressible rules
  – Many combine LP or HornFOL rules with OWL ontologies or OO syntax
DAML Rules Plan Overview

- **Vision**: studio for developers, studio for rule authors and users
- **Approach**: Composable Tools Suite supporting RuleML/SWRL
  - inferencing, translation/interoperability, authoring, testing
- **Infrastructure**: SemWebCentral, SWeDE
- **MIT Sloan** (Benj. Grosof lead): SweetRules RuleML tools:
  - translation and inferencing; architecture for suite integration
- **BBN** (Mike Dean lead): SWRL tools:
  - translator to Jena2; editor; validator
- **Stanford** (Mark Musen lead): Protégé support for rule authoring
- **More about Implementing SWRL**
- **Later**: More about FOL
- **Others Very Much Invited!**
  - some good candidates: those presented at WWW-2004 Developers Day Rules on the Web Track

5/28/2004 Copyright 2004 by Benjamin Grosof. All Rights Reserved
MIT Sloan Plan:

SweetRules:
Tools for RuleML Inferencing and Translation
Outline

• Concept, Architecture, and Goals
• Rule and Ontology Languages/Systems involved
• Capabilities and Components Today
• More about Combining Rules with Ontologies
• Application Scenarios and Examples
• Plans
• Motivations, revisited: Conclusions and Directions
• Acknowledgements
• Resources
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, Hoolet, ROWL, Triple, Jena, DRS, KAON (main), JTP, SWI Prolog, ...
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 heterogeneous families of rule systems
    • Encourage others: researchers; industry esp. vendors
**SweetRules Overview**

**Key Ideas:**

- Unite the commercially most important kinds of rule and ontology languages via 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.
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/DAML+OIL 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)

3. **Jess** (a pure subset of it = a large subset of Situated Ordinary LP)
   - *Uses recent novel theory for translation between SOLP and Production Rules.*

4. **IBM CommonRules** (whole = large subset of stratified SCLP)
   - Implements the Courteous Compiler (CC) KR technique.
     - which reduces (S)CLP to equivalent (S)OLP, tractably.
   - Includes bidirectional translators for XSB, KIF, Smols.
   - 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 ontologies. 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)
   - Uses recent novel SCLP representation of Frames with multiple default inheritance.

8. Smodels (NB: somewhat old version; large subset = finite OLP)
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 in RuleML. Edit in other rule systems’ syntaxes. Compare.
  - View human-oriented presentation syntax. View XML syntax. (Future: RDF.)
Capabilities and Components 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
- **Uses IBM CommonRules translators:** CommonRules ↔ {XSB, KIF, Smodels}
- **Some components have distinct names (for packaging or historical reasons):**
  - **SweetJess** translation and inferencing RuleML ↔ Jess
    - Available upon request free for research use as download.
  - **SweetOnto** translation RuleML ← OWL
    - Available currently as part of KAON open-source code base, called “DLP” component there
- **Code base:** Java, XSLT, shell scripts (for testing drivers)
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 default inheritance
     - 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.
Venn Diagram: Expressive Overlaps among KR’s
Some New Research Application Scenarios for Rule-based Semantic Web Services

- **SweetDeal** [Grosof & Poon WWW-2003] configurable reusable e-contracts:
  - Represents modular modification of proposals, service provisions
  - **LP rules** as KR. E.g., prices, late delivery exception handling.
  - **On top of DL ontologies** about business processes from MIT Process Handbook
  - Evolved from EECOMS pilot on agent-based manufacturing SCM ($51M NIST ATP 1996-2000 IBM, Boeing, TRW, Vitria, others)

- **Financial** knowledge integration (ECOIN) [Firat, Madnick, & Grosof 2002]
  - Maps between contexts using LP rules, equational ontologies, SQL DB’s.

- **Business Policies**:
  - **Trust** management (Delegation Logic) [Li, Grosof, & Feigenbaum 2003]:
SweetRules Tools Available Now

• Available currently:
  – SweetJess
  – SweetOnto = KAON’s DLP component

• Rest of Suite being updated and prepared for release on SemWebCentral
SweetRules Plans

• Update, integrate, and polish suite overall
• Support latest versions of RuleML and CommonRules
• Open source on semwebcentral.org
• Scenarios: Explore applications in SW Services, e.g., trust policies, contracting, monitoring, semantic interoperability mappings
• Requirements analysis
SweetRules Plans, cont.'d

• Pluggable architecture for Rules tools
  – SemWebCentral aspects
  – SWeDE aspects
  – Eclipse wrappers for tools
  – Ontology of tools
  – Composition patterns, high-level interfaces design
SweetRules Plans, cont.'d

• Additional Goals:
  – *Via suite integration:* More interoperability between SWRL and RuleML
  – *Ongoingly:* Update RuleML spec in synch with SWRL spec (in RuleML Initiative, Joint Committee)
  – *Via suite integration:* More authoring/UI capabilities
SweetRules Plans, longer-term

– *Later:* Justifications and proof, e.g., via suite integration with InferenceWeb

– *Later:* More wrt additional kinds of rule systems:
  - **ECA rules, SQL** (needs some theory work, e.g., events for ECA)
  - **RDF-Query and XQuery**

– *Later:* More wrt connections-to / support-of web services:
  - Importing knowledge bases / modules, procedural attachments, translation/inferencing, events, …
SweetRules Groups/People

• Collaborators: Said Tabet, RuleML; Mike Dean, BBN; Mark Musen, Stanford; Harold Boley, NRC/UNB

• More Collaborators Invited!
  – Many more rule/ontology systems are good targets for interoperation/translation:
    • Flora, cwm, ROWL, Hoolet, Triple, DRS, KAON, JTP, SWI Prolog, …
Resources

- See papers, talk slides, and links at http://ebusiness.mit.edu/bgrososf
- ../#RecentSoftware: Links to SweetJess, SweetOnto, CommonRules (where can download)
- ../#RecentPapersByTopic: (for most below, there are earlier versions too)
  - "Description Logic Programs: Combining Logic Programs with Description Logic", WWW-2003.

- RuleML http://www.ruleml.org
- DAML Rules http://www.daml.org/rules
- Joint Committee http://www.daml.org/committee
- SemWebCentral http://www.semwebcentral.org

5/28/2004 Copyright 2004 by Benjamin Grososf. All Rights Reserved
DAML Rules Plan Overview

- **Vision**: studio for developers, studio for rule authors and users
- **Approach**: Composable Tools Suite supporting RuleML/SWRL
  - inferencing, translation/interoperability, authoring, testing
- **Infrastructure**: SemWebCentral, SWeDE
- **MIT Sloan (Benj. Grosof lead)**: SweetRules RuleML tools:
  - translation and inferencing; architecture for suite integration
- **BBN (Mike Dean lead)**: SWRL tools:
  - translator to Jena2; editor; validator
- **Stanford (Mark Musen lead)**: Protégé support for rule authoring
- **More about Implementing SWRL**
- **Later**: More about FOL
- **Others Very Much Invited!**
  - some good candidates: those presented at WWW-2004 Developers Day Rules on the Web Track
More about Implementing SWRL
Venn Diagram: Expressive Overlaps among KR’s

- **Description Logic**
- **Horn Logic Programs**
- **Logic Programs (Negation as Failure)**
- **Logic Programs (Procedural Attachments)**
- **First-Order Logic**

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)
More SWRL Implementation Strategy

- **Named-classes-only restriction** on SWRL rules simplifies implementation including translation to rule systems (e.g., RuleML, Jess, XSB), yet does not sacrifice fundamental expressiveness.
  - Both current implementations of SWRL do this.

- Can translate full SWRL / DH $\Rightarrow FOL$ for which “native” (general-purpose) reasoners are indeed available.
  - E.g., OTTER or Simple Common Logic / KIF
  - The Manchester implementation of SWRL does this.
  - **Drawbacks:**
    - General-purpose FOL reasoners are often not very efficient.
    - Today, they also usually don’t directly support Webized syntax.
More SWRL Implementation Strategy, cont.'d

- Can translate subset of SWRL / DH into a KR for which “native” reasoners are indeed available. E.g.:

1. Horn LP expressible subset $\Rightarrow$ LP, e.g., RuleML, Jess, XSB
   - E.g., Horn LP SWRL rules + DLP OWL ontologies
     - Horn LP restriction on the SWRL rules means that:
       - rules are named-classes-only (no complex class expressions appear)
       - rules are definite (consequent is non-empty); and
       - ground atomic conclusions suffice.
   - The BBN implementation does this (Horn rules $\Rightarrow$ Jess)

2. DL-expressible subset of DH $\Rightarrow$ DL, e.g., OWL
   - E.g., DLP SWRL rules + any OWL-DL
     - E.g., SWRL rules are used to define some ontologies
     - No implementation of this is yet available.
More about SWRL V0.6 Built-Ins

- The built-ins (3.) can be viewed as predicates/relations that have a fixed extension.
  - Alternatively, the set of tuples satisfied by calls to the built-ins can be viewed as corresponding to a virtual fact set adjoined to the FOL theory.
- These are similar to sensors in Situated Logic Programs RuleML.

- The built-ins can be implemented via procedural attachments that are purely informational (free of side effects)
  - Intuitively, they are typically evaluated when rule body is tested.
Punchline on Near-term Implementation Strategy

• (Unless you can invent a whole new technique…)

1. If you want full SWRL expressiveness, translate to some FOL syntax and then use a FOL theorem-prover to do inferencing.

2. If you want to translate to LP to exploit one of the many LP rule engines available (e.g., RuleML, Jess, XSB), or to exploit beyond-Horn LP expressive features (e.g., nonmon or actions), then restrict the SWRL ontologies to DLP.
   • RuleML is the obvious choice of translation target: it’s SWRL’s extension in direction of fuller LP expressiveness, and facilitates translations to multiple other rule languages’ engines (e.g., Jess, XSB).
   • SweetOnto tool (a.k.a. KAON DLP package) translates DLP OWL to RuleML. (There are other DLP implementations too.)
DAML Rules Plan Overview

• Vision: studio for developers, studio for rule authors and users
• Approach: Composable Tools Suite supporting RuleML/SWRL
  – inferencing, translation/interoperability, authoring, testing
• Infrastructure: SemWebCentral, SweDE
• MIT Sloan (Benj. Grosof lead): SweetRules RuleML tools:
  – translation and inferencing; architecture for suite integration
• BBN (Mike Dean lead): SWRL tools:
  – translator to Jena2; editor; validator
• Stanford (Mark Musen lead): Protégé support for rule authoring
• More about Implementing SWRL
• Later: More about FOL
• Others Very Much Invited!
  – some good candidates: those presented at WWW-2004 Developers Day Rules on the Web Track
Getting Involved!

- Please contact Benjamin Grosof and Mike Dean (DAML Rules co-chairs) with your rules …
- Tools
- Ideas
- Rulebases
- Use cases
- Other resources
- Relevant plans