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

http://sweetrules.projects.semwebcentral.org

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

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

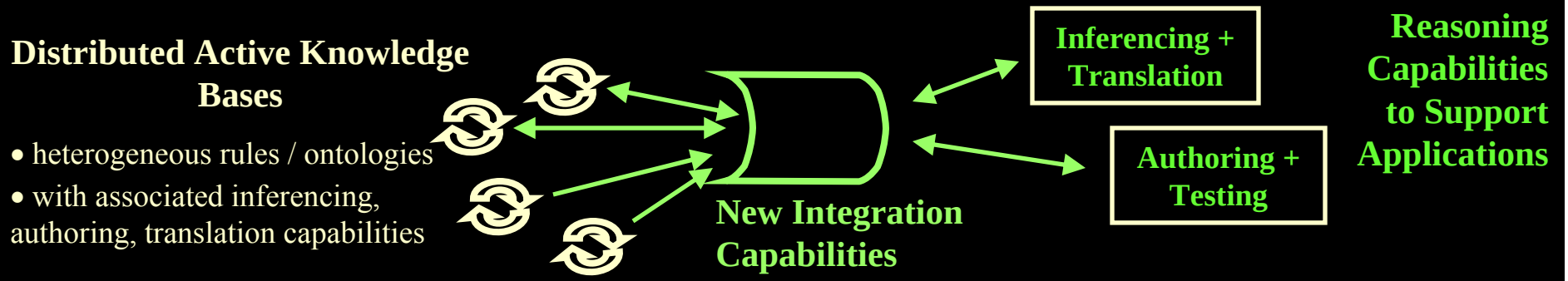
SweetRules V2.1 Overview

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

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

Application Areas (prototyped scenarios):

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



SweetRules Concept and Architecture

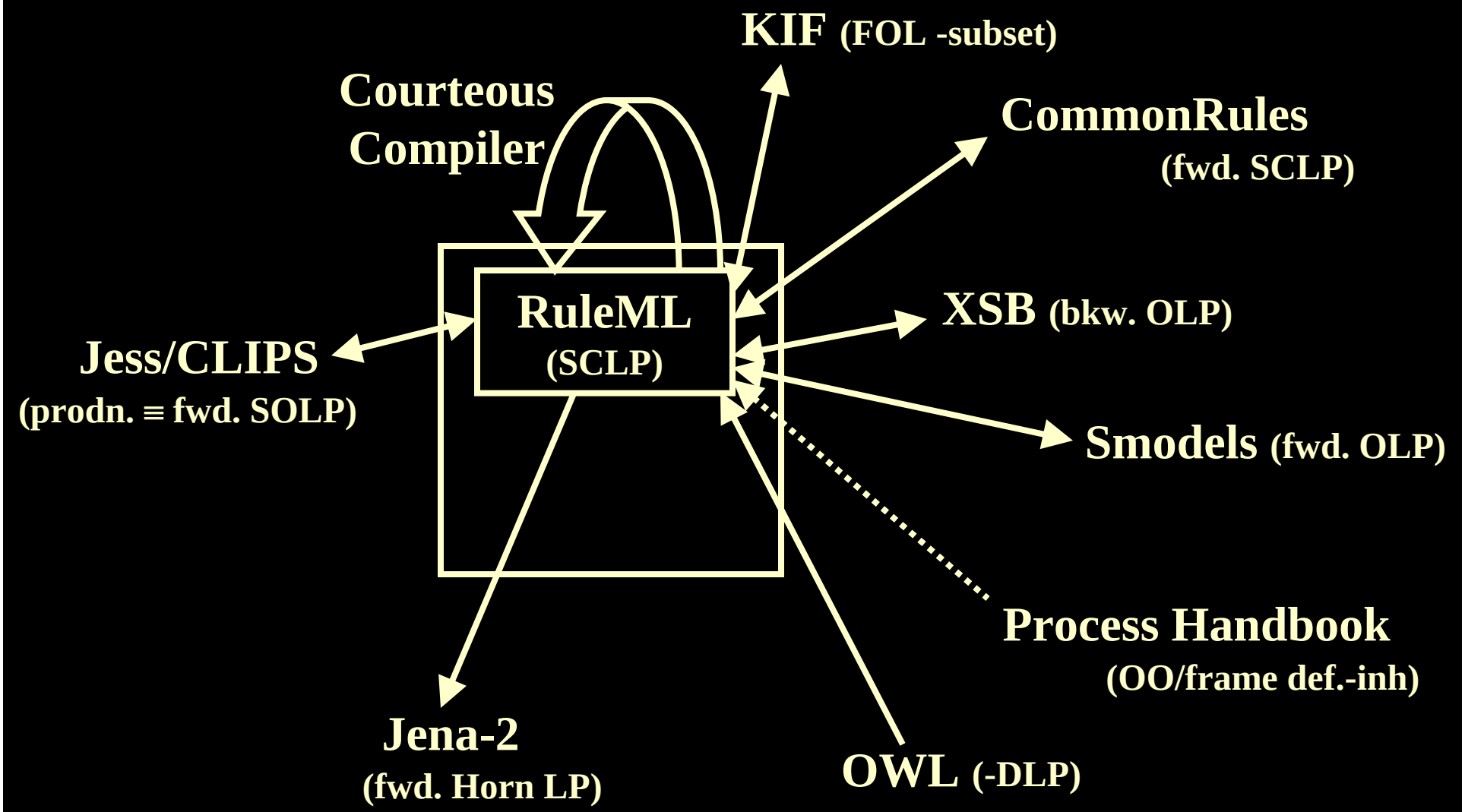
- Concept and Architecture: Tools suite for Rules and RuleML
  - Translation and interoperability between heterogeneous rule systems (forward- and backward-chaining) and their rule languages/representations
  - Inferencing including via translation between rule systems
  - Authoring, Analysis, and testing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

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

SweetRules V2.0+ Fundamental KR Today

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

SweetRules V2.1 Translators Graph

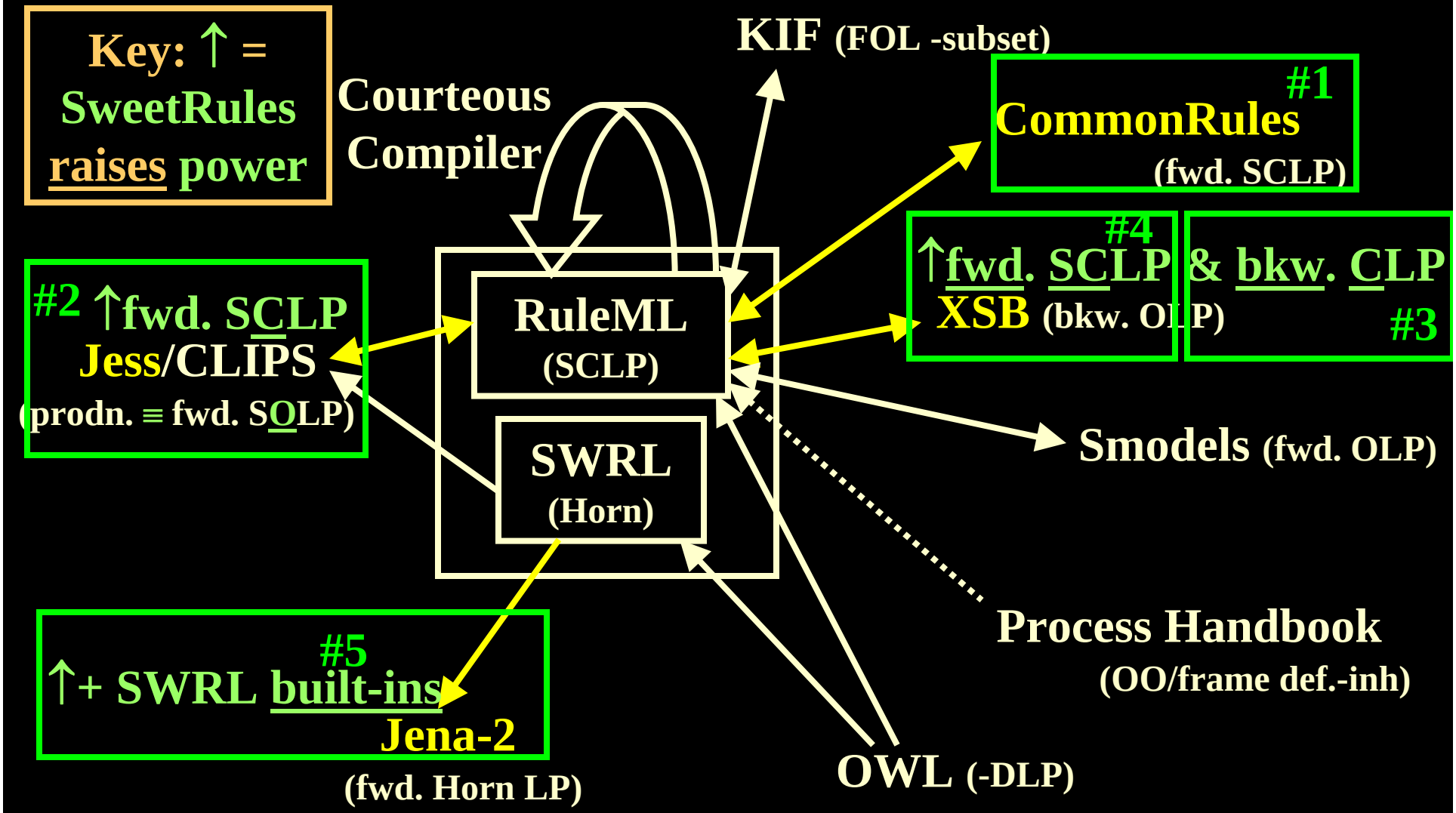


(SCLP = Situated Courteous Logic Programs. OLP = Ordinary LP (plain NAF))

SweetRules Inferencing Capabilities Today: Overview

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

SweetRules V2.0+ *New Inferencing Engines*



SweetRules Capabilities Today Cont.'d

- Authoring and Testing front-end: *currently less mature, more partial*
  - Command-line UI
  - Protégé OWL Plug-in Enhancement
    - SWRL Rule Editor (separate component from SweetRules)
- Analyzers incl. Validators: *currently less mature, more partial*
  - e.g., DiffFacts for incremental reasoning

Novel Capabilities I

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

Additional Firsts in Implementation

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

Novel KB Merging of Rules + Ontologies

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

SweetRules Application Scenarios

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

Business Value of RuleML Rules for Policies, e.g., Authorization/Security

- Interoperability, flexibility and re-use benefits
  - Reduced Vendor Lock-in
- Easier Integration: with rest of business policies and applications, business partners, mergers & acquisitions
  - Enterprise integration, B2B
- Reduced system development, maintenance, & training costs
- Better/faster/cheaper policy development & administration
  - Easier to understand and modify by humans
- Quality and Transparency of implementation in enforcement
  - Provable guarantees of behavior of implementation
- Improved visibility and assurance in enterprise policy implementation ⇒ better compliance, senior governance
- Rich, expressive policy management language allows better conflict handling in policy-driven decisions: prioritization & negation mechanisms
- ⇒ Agility, change management ↑

SweetRules Goals & Site

- 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
- See http://sweetrules.projects.semwebcentral.org
  - Open-source code; extensive documentation; tutorial material

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, ...