Some Key Concepts and History for Semantically Interoperable Web Rules

Prof. Benjamin Grosof

MIT Sloan School of Management

Information Technologies group

http://ebusiness.mit.edu/bgrosof

Invited Presentation (45-min.) at

Kickoff Face-to-Face Meeting of

W3C Rule Interchange Format Working Group,

Burlingame, CA, USA, Dec. 8-9, 2005,

http://www.w3.org/2005/rules

Quickie Bio of Presenter Benjamin Grosof

- 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:
    - e-contracting/supply-chain;   finance; trust; …
  - Overall knowledge representation, e-commerce, intelligent agents

- Co-Founder, Rule Markup Language Initiative – the leading emerging standards body in semantic web rules (http://www.ruleml.org)
- Area Editor, Semantic Web Services Initiative – which coordinates world-wide SWS research and early standards (http://www.swsi.org)

Resources for More Info

- On author's website (http://ebusiness.mit.edu/bgrosof), see especially:
  1. ISWC-2005 Rules Tutorial slideset (half-day conference tutorial, 200+ detailed slides)
     - …/#ISWC2005RulesTutorial
     - "Semantic Web Rules with Ontologies, and their E-Service Applications"
       – A. Core technology: knowledge representation languages, theory, and techniques; standards design
       – B. Tools: rule inference engines, translators between rule systems/languages, ontology integration
       – C. Applications in E-Services: semantic mediation and ontology translation, e-contracting, trust policies, financial reporting
         - Business value analysis, market roadmapping
  2. Production Logic Programs paper & info
     - …/#ProductionLogicPrograms (or just …/#PLP)

Also:

– Recent talks (including this one soon), not just papers

– SweetRules toolset (http://sweetrules.projects.semwebcentral.org)

Preface

- This is a kind of hybrid of a conventional invited tutorial-y talk and a glossary
  – Glossary style was requested by WG chairs
  – I adapted previous existing tutorial-y material, but not exactly as a glossary
  – I've never tried before to do a glossary kind of thing as a talk; here goes…

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents
- (BUNCH OF EXTRA OPTIONAL SLIDES)

Talk Mode: the MIT Firehose

Shortened from a 90-minute talk
$\Rightarrow$ Some skimmed

Flavors/Families of Rules Commercially Most Important today in E-Business

- E.g., in OO app's, DB's, workflows.
- "CCI" = Currently Commercially (most) Important

1. Relational databases (RDBMS), SQL: Views, queries, facts are all rules.
   - XQuery, SPARQL emerging. SQL99 even has recursive rules.
2. Production rules (OPS5 and CLIPS heritage): e.g.,
   - Fair Isaac, ILOG, Haley, etc.
3. Event-Condition-Action rules (loose family similar to PR), cf.:
   - business process automation / workflow tools.
   - active databases; publish-subscribe
4. Prolog. *"logic programs" as a full programming language.*
   - *"Pure" Prolog – declarative LP subset, has no cuts or external procedure calls, does backward inferencing in declarative LP*
5. *(Lesser: other knowledge-based systems, and things hard to classify or further from declarative such as some "business rule" systems.)*

Production Rules (PR): History

- Grand-daddy: OPS5 research system at CMU in ’70’s
- CLIPS *system*: open PR system done by US govt ~ a decade ago
- CLIPS *syntax*: used with tweaks by many current PR systems
- Have incremental/dynamic/updating capabilities
- PR reengineered in ’90’s to be fine-grain embedded in C++/Java etc. programming language, with access to those external objects
- OMG PRR standards effort since late 2003
- *Full PR systems often also have scripting and a kind of backward-inferencing capability; for semantically interoperable web rules that has not been the focus (at least initially).*

Event-Condition-Action Rules (ECA): History

- Loose family; no consensus/standard detailed formulation/abstraction
- Fairly similar to Production Rules: forward, Conditions, Actions
- Plus there's the "Event" notion (see next slide)
- More focus than PR on incremental inferencing and specialized optimizations around "Events"

- Many database systems have ECA capabilites, e.g., for transactional triggers or pub-sub.
- In '90's became used widely for loose coupled business process automation / workflow / integration / orchestration

Events in ECA Rules

- "Event" is a kind of premise info update, <u>and</u> a kind of control trigger for incremental inferencing
  - This conflates declarative and procedural aspects! *Challenge ;-)*
  - Often not precisely described, for given ECA system/language
- "Event" part of a rule body is a kind of condition, and control "port"
- Often there's "complex event processing" with specialized treatment
  - E.g., event sublanguage and special processors for generating/testing events
- *History* of event updates/info-states is often important

- Other flavors of rules can also do events and incremental inferencing, to varying degrees

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents

History of Standards Efforts in Interoperable Knowledge Representation: SQL and KIF

- SQL – a wild success ☺
- Knowledge Interchange Format (KIF) for First Order Logic (FOL) (early ’90’s)
  - ASCII syntax
  - Also included some tweaks, e.g., quotation/reification, lists
  - Considered nonmon seriously, but punted – not clear how to do it with full FOL
  - Designed/purposed for researchers, initially
  - ANSI standards draft, then pretty much languished
  - A few commercial efforts supported, e.g., IBM Agent Building Environment ’96
  - Didn’t take off. Little industry demand/pull for FOL.
  - Successor is ISO Common Logic standards effort
    - Also slow-moving. Cooperating fitfully with RuleML and SWSL.
  - FOL RuleML captures most of the expressiveness.

Standards History II: ISO Prolog

- ISO Prolog
  - An attempt to standardize full Prolog, a while ago
  - Good list of names for builtins and datatypes, for example.
  - My personal take on it:
    - Useful, but not a huge impact.
    - Prolog syntax had largely already been standardized indirectly via pedagogical practice and close ties to research community

KR Standards History III: RuleML

- RuleML (2000-) Rule Markup (& Modeling) Language
  - Pioneered standards design, and related prototyping, of true webized semantically interoperable web rules Declarative LP and extensions/restrictions. FOL and extensions/restrictions.
  - Achieved wide influence and mindshare in the R&D cognoscenti
  - Has engaged/driven a number of rules standards efforts/bodies since, including in OMG, W3C, Oasis, ISO, JSR; and SWSL, SWRL, WSML/WRL, CommonLogic
  - Grew out of Business Rules Markup Language, markup language supported by IBM CommonRules, piloted in the IBM-led EECOMS supply chain collaboration industry-government consortium in late 90's ($29M project).
  - Largely nucleated the semantic web rules research community
    - RuleML-2005 and prev. workshop series: an annual full conference, colocated with ISWC
  - *You'll hear more later today in another presentation*

KR Standards History IV: SWRL

- SWRL (2002-2004 mainly) “Semantic Web Rules Lang.”
  - Purposed to do rules on top of OWL esp. for extended ontologies
  - Chose not to tackle non-FOL-expressible aspects of rules
  - Restricted (function-free Horn) RuleML rules, on top of / merged with OWL-DL
  - All predicates are OWL classes or properties
  - RDF facts, as usual in OWL
  - FOL semantics
  - XML-Schema datatypes and builtin operations, too
  - A little-studied medium-large subset of FOL expressiveness. Undecidable computationally.
  - Inferencing techniques can rely on FOL or LP (for subset) engines
  - *You’ll hear more later today in another presentation*

KR Standards History V: SWSL

- SWSL (2003-) "Semantic Web Services Language"
  - By SWS Initiative: coordinates research and early standards design world-wide
  - Purposed for representing and reasoning with semantic descriptions of web services
  - Requirements analysis: want both expressive declarative LP and FOL
    - Including default/nonmon negation, courteous priorities, frame syntax, Hilog meta-logical expressiveness, datatypes, skolemization, equality
  - Cluster of SWS tasks around policy knowledge → use LP
    - Advertising, trust, contracting, mediation, monitoring
    - Another cluster around FOL for validation, composition, anal.
  - Picked RuleML, then collaborated with RuleML to extend its features, esp. drawn from F-Logic, and webized corresponding presentation syntax
  - *You'll hear more about it later today in another presentation*

KR Standards History VI: WRL

- WRL (2004-) “Web Rule Language”
  - From Web Services Mediation Ontology effort
  - Purpose: rule-based language for representing ontologies
  - Based mainly on F-Logic LP. Largely similar to RuleML and SWSL. (Overlap in participants.)
  - Subset defined corresponding to Description LP
  - Concerned with integration with OWL-DL, esp. via Description LP, and with rest of services work in WSMO.
  - Aims to be nicely packaged and practical
  - Collaborating with RuleML

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, "Semantic", Syntax, "Ontology"
- Declarative LP KR Concepts, History
- Indirect Inferencing
- "Policy", "Semantic Mediation" Tasks, Knowledge
- *Context:* "Semantic Web" and constituents

Some (More) KR Terminology

- Procedural attachment, a.k.a., “aproc” (way fewer syllables ☺) :
  - a procedure (e.g., Java method), <u>external</u> to the inferencing procedure itself, that is associated with a belief expression in the rule language and is called to obtain additional info or perform side-effectful actions
- Translation: mapping from one rule representation/syntax/system to *another*
- “Transformation”: mapping within the *same* rule representation/syntax/system, e.g., to <u>reduce</u>/eliminate an expressive feature (PROPOSED meaning)
- “logical function” = “constructor”; contrast “function” in CLIPS
- definite rule = rule with a non-empty head
- “Datalog” = function-free definite Horn
  - sometimes = just function-free;
  - sometimes = without definiteness
- Authoring of rulebase/KB: specifying/creating rules/knowledge

Some KR Terminology II

- Default = a rule/premise that is true by default, that goes through if consistent/unconflicted with other info within the scope of a particular rulebase/KB in which it appears
- Priority among rules/defaults: declarative vs. control; partial vs. total; inferrability
- Default negation, a.k.a. negation-as-failure (NAF, "naf"), a.k.a. weak negation. A key construct in LP. Always scoped within the particular rulebase in which it appears. Well founded semantics is well behaved and tractable. Alternative "stable" semantics differs in some cases, and can blow up.
- Strong negation, a.k.a. limited classical negation ("neg"), in LP.
- "Closed world": loose vs. precise.
  - Loose: scope the knowledge base, while doing nonmonotonic reasoning/entailment/inferencing
  - Precise: have/apply neg p(?xt) as a low-priority, catch-case default, for every predicate p, where ?xt is a tuple of variables of appropriate arity

Some KR Terminology III

- KB Merging: combine premises and/or conclusions from multiple KB’s, to form a new KB
- “Inclusion” merging: define a KB as including other(s).
  - E.g., in RuleML (e.g., SweetRules supports). Similar in OWL.
- “Tight” description of expressiveness of a particular KB:
  - Precise description of what expressive features are vs. are not present in the particular KB
  - Useful for interoperability – helps minimize burdens and maximize reuse
  - Implies it helps to have a finer “lattice” of expressive feature/restriction combinations

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents

Key Open Source Tools for Semantic Rules on Web, I: Flora-2

- Flora-2: F-Logic LP
  - Frame syntax, Hilog, skolemization, datatyping, more
  - built on top of XSB Prolog

Key Open Source Tools for Semantic Rules on Web, II: SweetRules

- SweetRules V2: tool set platform
  - Supports expressively powerful RuleML-based interoperability and inferencing
  - and also SWRL. Basis: declarative Logic Programs KR at heart, + some FOL
  - Description LP technique for merging restricted OWL-DL into LP
  - Courteous LP prioritized conflict handling
  - Unrestricted (scoped) default negation
  - Production LP / Situated LP procedural attachments for actions and tests/queries cf. PR
    - has generalization to permit <u>un</u>bound such queries
  - Indirect inferencing: translate, infer in another rule system, translate back
  - 1st interoperability between Production Rules and declarative LP
  - Based on Production LP KR approach, overall
  - Translation/inferencing in Jess, XSB, OWL/RDF, CommonRules, Jena, KIF, more
  - Inclusion merging of heterogeneous rulebases and ontology knowledge bases
  - Dozens of translators, pluggable and automatically composed
  - Supports WSDL actions – a true rule-based semantic web service system

Key Open Source Tools III: cwm

- cwm: Notation 3 rules lora-2: F-Logic LP
  - Nice webized syntax and capabilities for RDF
  - Not well studied semantically
  - But clearly: subset is declarative LP
  - Interesting KB merging
  - Implemented in Python

Key Open Source Tools IV: Jena-2

- Jena-2: Rete rule engine within OWL/RDF toolkit
  - Has forward Rete-based engine. Can view as baby PR.
  - Handles RDF data natively.

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents

"Semantic"

- "Semantic" in "semantic rules" and "semantic web" means:
  - 1. Knowledge-based
  - … and …
  - 2. Having meaning independent of algorithm and implementation
  - I.e., equipped with an interoperable conceptual abstraction
  - … based on declarative knowledge representation (KR)
  - (vs. procedural, dependent on inferencing control strategy, inferencing engine)

Concept of Knowledge Representation (KR)

- A knowledge representation S is defined as a triple (LP, LC, |=), 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)
  - |= is the <u>entailment</u> relation.
    - Conc(P,S) stands for the set of conclusions that are entailed in KR S by a set of premises P
    - We assume here that |= is a functional relation.

- Heritage of KR concept: AI, DB areas of comp sci; earlier: logic from math, phil.; programming languages foundations

Example of Entailment: Mortality

- In First-Order Logic (FOL) KR:
  - Let P be the premises:
  - $\forall$?X. human(?X) $\Rightarrow$ mortal(?X).
  - human(Socrates).
  -
  - In FOL, P entails (among others) the conclusion:
    - mortal(Socrates).
  - Notation:
    - "$\forall$" means "for all".
    - "?" Prefixes a logical variable.

Example of Entailment: Sunday Stroll

- In Bayesian Probability KR:
  - Let P be the premises:
    - prob(rainySunday) = 0.4.
    - prob(funSunday | rainySunday) = 0.3.
    - prob(funSunday | ¬rainySunday) = 0.9.
  -
  - In this KR, P entails (among others) the conclusion:
    - prob(funSunday) = 0.66.

Example of Entailment: Discounting

- In the Courteous Logic Programs KR (e.g., RuleML):
  Let P be the premises:
  - {loyald} discount(?cust, RamadaHotel, 10percent) $\leftarrow$ memberOf(?cust, AAA).
  - {seniord} discount(?cust, RamadaHotel, 25percent) $\leftarrow$ age(?cust, ?x) and greaterThan(?x, 64).
  - overrides(seniord, loyald).
  - $\perp \leftarrow$ discount(?c, ?h, ?y) and discount(?c, ?h, ?z) | (?y $\neq$?z).
  - memberOf(Faisal, AAA).
  - age(Faisal, 72).

  - In this KR, P entails (among others) the conclusion:
    discount(Faisal, RamadaHotel, 25percent).

Example of Discounting, cont.'d

In the more general Production Logic Programs KR:

Suppose one adds the rule:

– @emailCouponAd(?cust, RamadaHotel, ?x)
    ← discount(?cust, RamadaHotel, ?x).

Then P entails the action (i.e., sanctions a call to an attached procedure):

@emailCouponAd(Faisal, RamadaHotel, 25percent).

Here, "@" prefix indicates an attached procedure.

Model Theory As Basis for Semantics

- Normally entailment/Conc can be defined in terms of models
  - E.g., First Order Logic (FOL)
  - E.g., Declarative Logic Programs
- A model assigns a truth value to each expression in LC, while satisfying an overall accordance with the premises.
- The set of possible truth values depends on the KR
  - {true, false} for FOL; {t,f,u} for LP; {real in [0,1]} for Bayesian probability.
- How accordance (with the premises) is defined depends on the KR
- The number of models of a given set of premises may be 0, 1, or many, depending on the KR.
  - 0 is a kind of inconsistency or incoherence
  - If many, entailment is defined in terms of that set, usually as what's true in *all* models.

Inferencing

- Inferencing is the process or procedure of computing conclusions
  - Can be abstract / high level algorithm, e.g., non-deterministic, non-terminating
- Soundly = if a conclusion is inferred, it is also entailed
- Completely = if a conclusions is entailed, it is also inferred

- Proof theory means characterization of inferencing procedures in terms of how they correspond to the model theory. There may be multiple such abstract procedures.
  - E.g., resolution for FOL
  - E.g., PLP-resolution for LP

- Inferencing control strategy means what order steps of attempted proof are tried, e.g., in what order rule bodies/conditions are tested.

"Ontology" More General than OWL

- "ontology" in general sense = definitional knowledge [sense from AI and philosophy]
  - Could be in any KR, e.g., FOL, LP, or probabilistic

- Important kinds of ontologies:
  - Taxonomies: vocabulary and basic class hierarchy
  - Description Logic
  - Object oriented with default inheritance, e.g., C++/Java/C# class-hierarchy frameworks with overriding or cancellation of inheritance
  - Database schemas
  - XML schemas
  - UML aspects
  - Axiomatizations in FOL of time, space, processes

Kinds of Rule Syntax

- Concrete syntax: (i.e., encoding as data)
  - (In) Markup. Esp. today:
    - In XML
    - In RDF
  - Presentation syntax in ASCII/Unicode
- Abstract syntax: via EBNF-y grammar or data model

- E.g., RuleML has all the above
  - "Rule Markup *and Modeling* Language"

- Also:
  - "Meta-model" cf. OMG (e.g., RuleML has draft in progress)
  - Graphical, instead of Textual (e.g., in rule editor tools)

Slide also by Harold Boley (NRC)

RuleML Example: Markup and Tree

Markup Syntax

'The **discount** for a *customer* buying a *product* is **5.0 percent** if the *customer* is **premium** and the *product* is **regular**.',

discount(?customer,?product,“5.0 percent“) ← premium(?customer) /\ regular(?product);

```
<imp>
  <head>
    <atom>
     <opr><rel>discount</rel></opr>
     <tup><var>customer</var>
          <var>product</var>
          <ind>5.0 percent</ind></tup>
    </atom>
  </head>
  <body>
    <and>
      <atom>
        <opr><rel>premium</rel></opr>
        <tup><var>customer</var></tup>
      </atom>
      <atom>
        <opr><rel>regular</rel></opr>
        <tup><var>product</var></tup>
      </atom>
    </and>
  </body>
</imp>
```

```
imp
 head
 ----- atom
        -opr -- rel ......... discount
        -------> var ......... customer
        -------> var ......... product
        -------> ind ......... 5.0 percent

 body
 ----- and
        -atom
           -opr -- rel ......... premium
           -------> var ......... customer
        -atom
           -opr -- rel ......... regular
           -------> var ......... product
```

tup is an ordered tuple.

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents

History of Declarative LP

- Developed as a theoretical abstraction of RDBMS and pure Prolog

- Negation well understood by ’94 (not well understood before ’84)

- A number of expressive extensions in the last two decades
- Lots of algorithmic insights, expertise, good implementations
  - E.g., XSB; Flora-2, SweetRules on top of XSB

- For function-free case, similar tractability to RDBMS/SQL. (vs. FOL intractable.)

- Cannot do “reasoning by cases”, i.e., draw disjunctive conclusions and then chain on them in branched fashion. This is essential to the attractive computational complexity. E.g., not good for general constraint solving *a la* complex scheduling.

- Conclusions are essentially (reducible to) ground literals.

Declarative Logic Programs KR

- Basic case: Definite Horn function-free equality-free LP. Tractable. Same as core SQL but with no limitation to backward direction of inferencing.
- A number of extensions -- and restrictions …
  - thus an extensible family forming a lattice.
  - “LP” can mean the family or a member of it
  - “foo LP” can mean a sublattice or member of it
- Datatypes
- Logical functions. Loses tractability.
- Default negation (scoped), a.k.a. Negation-As-Failure

Declarative Logic Programs KR II

- Courteous LP: prioritized conflict handling; strong negation too. Defeasible Logic similar.
  - Guarantees consistent set of conclusions; wrt mutex's (mutual exclusion integrity constraints)
  - Reducible to (default) negation
- Production LP (generalizes Situated LP): procedural attachments for side-effectful actions and tests/queries.
  - Enables interoperability with Production Rules and similar ECA.
  - Built-ins as simple case: for arithmetic, string, etc. comparisons/operations
- Frame syntax cf. F-Logic
- Hilog cf. F-Logic: "meta-linguistic"/"reflection" capability

Declarative Logic Programs KR III

- Lloyd-Topor enhanced logical connectives and quantifiers
  - Reducible to (default) negation
- Skolemization, e.g., for existentials, RDF blank-nodes
- Reification: kind of quotation: belief formula becomes logical term
- Explicit equality: in heads of rules; with background special axioms
- Integrity constraints: reporting vs. inconsistency-generating

- A few other things too

- See SWSL report for a fairly good overview (it omits procedural attachments, however!)

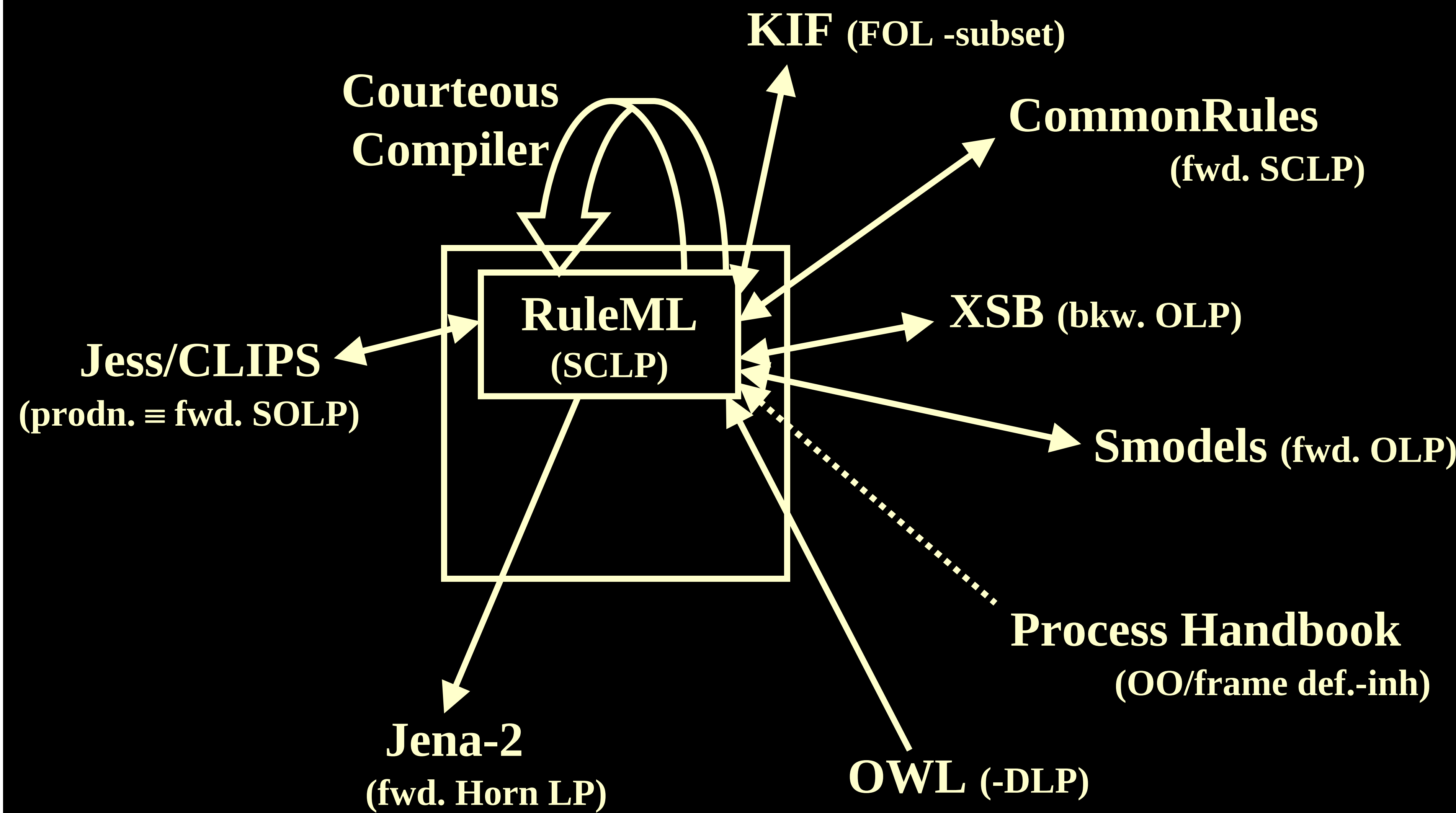
Outline

- Commercially Important Rule System Families

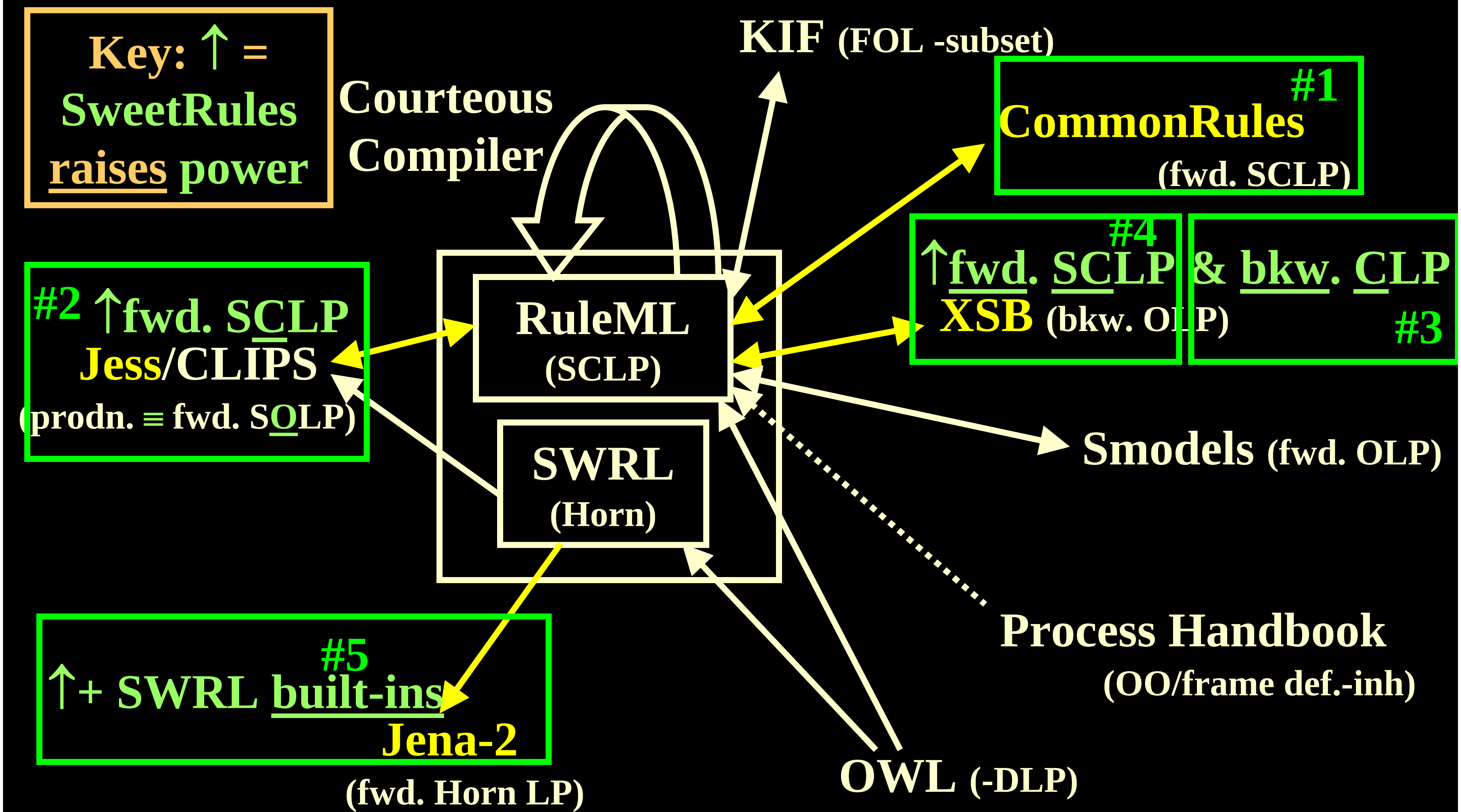
Indirect Inferencing

- Indirect inferencing from rule lang/system S1:
    1. translate to another rule system S2
    2. run inferencing in that system's engine
    3. translate back the conclusions to S1
  - Can use composite translators: e.g., S1 to S2 to S3
- E.g., in SweetRules:
  - from RuleML to XSB or Jess then back to RuleML

SweetRules V2.0   Translators Graph



SweetRules V2.0 *New Inferencing Engines*



Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, "Semantic", Syntax, "Ontology"
- Declarative LP KR Concepts, History
- Indirect Inferencing
- "Policy", "Semantic Mediation" Tasks, Knowledge
- *Context:* "Semantic Web" and constituents

"Policy" in Broad Sense -- *Knowledge and Tasks*

- Specification prescriptively of what is to be done
- Nice match representationally to rules, e.g., prioritized defaults
- Frequently involves "permitted/allowed", "prohibited/disallowed"
- E.g., in authorization, contracts, monitoring

"Semantic Mediation"

- Translate knowledge in a semantics-preserving way that handles differences in:
  - KR language, system, or syntax
  - Ontology
  - Context
- Nice match to rules: for specification of mapping ontology/context

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents

Semantic Web: concept, approach, pieces

- Shared semantics when interchange data ∴ knowledge
- Knowledge Representation (cf. AI, DB) as approach to semantics
  - Standardize KR syntax, with KR theory/techniques as backing
- Web-exposed Databases: SQL; XQuery (XML-data DB's)
  - Challenge: share DB schemas via meta-data
  - RDF: "Resource Description Framework" W3C standard
    - Meta-data low-level mechanics: unordered directed graphs (vs. ordered trees)
    - RDF-Schema extension: simple class/property hierarchy, domains/ranges
- Ontology = formally defined vocabulary & class hierarchy
  - OWL: "Ontologies Working Language" W3C standard
    - Subsumes RDF-Schema and Entity-Relationship models
    - Based on Description Logic (DL) KR ~subset of First-Order Logic (FOL))
- Rules = if-then logical implications, facts ~subsumes SQL DB's
  - RuleML: "Rule Markup Language" emerging standard
    - Based on Logic Programs (LP) KR ~extension of Horn FOL
    - Also provide FOL KR

"Semantic Web"

- = Semantic knowledge/inferencing/processing on the web
  - Roughly: knowledge representation on the web
  - More imprecisely: "knowledge-based" web
  - "On the web" is a matter of degrees of web-izing:
    - web-accessible; in markup; using URI predicate names; using web services for procedural attachments; …

Outline

- *Terminology/Concepts and History all along the way…*
- Commercially Important Rule System Families
- Knowledge Representation (KR) Standards Efforts
- More KR Terminology
- Key Open Source Tools
- Concept of KR, “Semantic”, Syntax, “Ontology”
- Declarative LP KR Concepts, History
- Indirect Inferencing
- “Policy”, “Semantic Mediation” Tasks, Knowledge
- *Context:* “Semantic Web” and constituents
- (BUNCH OF EXTRA OPTIONAL SLIDES)

OPTIONAL SLIDES I FOLLOW

Semantic Rules: Use Cases from our research

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

Production Logic Programs: A New Fundamental Rule KR Approach

In 2005:

- Production extension of LP:
  - actions and tests appear directly within rules (procedural attachments)
  - Generalizes Situated LP a bit, and reformulates it more familiarly
- Theory & algorithms achieving semantic interoperability of
  {core Production Rules} ← → declarative LP
  - Handles negation correctly, by stratifying PR agenda control strategy
  - 1st declarative semantics for Production Rules

- Combines with all the other features: Courteous, …
- → "Production LP" as umbrella LP KR approach

Semantic Rules News

News recently:

- Fundamental theory and technique breakthroughs, e.g.:
  - Declarative logic programs (LP) basis for interoperability, then webized → RuleML standards design (2001-)
  - Courteous LP prioritized defaults, robust modular merging
  - Description LP ontology integration
  - Production LP interoperability+semantics for production rules, declarative procedural attachments for actions and queries
  - SweetRules V2 open source toolset platform (2004-)
- Large US, EU research projects (DAML, WSMO) focus on rules (DARPA Agent Markup Language; Web Service Mediation Ontology)

Semantic Rules News (cont.'d)

News recently:

- W3C forms Rule Interchange Format WG, full standards effort, after holding a Workshop (2005)
- OMG forms standards efforts on production rules, rule management
- Semantic Web Services Framework design (2005) focuses on rules
- Rule-based Policy area heats up in web services, semantic web, incl. at Oasis. Oasis forms Semantic Execution Env. standards effort (2005).
- Semantic web rules workshop series becomes full research conference (RuleML-2005) colocated with ISWC
- Gartner etc. reports on rules sector

Semantic Rules: Differences from Rules in the 1980's / Expert Systems Era

- Get the KR right (knowledge representation)
  - More mature research understanding
  - Semantics independent of algorithm/implementation
  - Cleaner; avoid general programming/scripting language capabilities
  - Highly scaleable performance; better algorithms; choice from interoperability
  - Highly modular wrt updating; use prioritization
  - → Highly dynamic, scaleable rulebase authoring: distributed, integration, partnering
- Leverage Web, esp. XML
  - Interoperable syntax
  - Merge knowledge bases
- Embeddable
  - Into mainstream software development environments (Java, C++, C#); not its own programming language/system (cf. Prolog)
- Knowledge Sharing: intra- or inter- enterprise
- Broader set of Applications

KR: What's the Game? Desiderata

- Expressiveness: what can be said
  - useful, natural, complex enough
- Syntax: encoding data format -- e.g., in XML
  - easy enough to edit and communicate, by computers and by humans
- Semantics: principles of sanctioned inference, independent of reasoning algorithms:
  - clear, useful, natural, and understandable enough
- Computational Tractability (esp. worst-case): scale up in a manner qualitatively similar to relational databases: computation cycles go up as a polynomial function of input size
- Reasoning algorithms (compute the entailed conclusions):
  - sound (correct), complete, efficient, clear, and simple enough to engineer

SWS and Rules   Summary

*** SWS Tasks Form 2 Distinct Clusters,*

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

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

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

OPTIONAL SLIDES II FOLLOW

New Fundamental Rule KR Theory I

that enables Key Technical Requirements for SWS

- 1. Courteous Logic Programs: [Grosof]

  KR to combine rules from many sources, with:

  – Prioritized conflict handling to enable consistency, modularity; scaleably

  – Interoperable syntax and semantics

- 2. Situated Logic Programs: [Grosof]

  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

New Fundamental Rule KR Theory II

that enables Key Technical Requirements for SWS

- 3. Reference Ontologies from Rules Via URI Names [Grosof]

- 4. Description Logic Programs: [Grosof, Horrocks, Volz, & Decker]

  KR to combine LP (RuleML) rules on top of DL (OWL) ontologies,

  with:

  – Power in inferencing (including for consistency)
  – Scaleability of inferencing
  – Approach: Analyze and exploit the Intersection of DL and LP (within FOL)

New Fundamental Rule KR Theory III

that enables Key Technical Requirements for SWS

- 5. Courteous Inheritance: [Grosof & Bernstein]
  - OO default inheritance as Courteous LP
  - Used to Leverage Process Handbook, & other Legacy OO Knowledge, to create SW service ontologies
- 6. Production Rules as LP: [Grosof]
  - OPS5-heritage production rules as Situated Courteous LP
  - Find and fix fundamental weakness in chaining through negation in Rete-based inferencing
  - Unify commercially most important and fast-growing rule families
- 7. Hypermonotonic Reasoning: [Grosof (in-progress)]
  - Unify Nonmon LP KR with FOL KR
  - Nonmon LP as sound & incomplete wrt FOL

New Fundamental Rule KR Theory

that enables Key Technical Requirements for SWS

In 1985-94:

- Prolog interoperable with relational DB; LP extends core-SQL [many]
- Richer logical connectives, quantifiers [Lloyd & Topor]
- "Well Founded" Semantics for Negation-As-Failure [Van Gelder et al; Przmusinski]
- Hilog quasi-higher order expressiveness, meta-syntax flexibility [Kifer et al.]
- Frame syntax cf. F-Logic [Kifer *et al.*]

In 1995-2004:

- Courteous LP: prioritized conflict handling [Grosof]
  - Robust, tractable, modular merging & updating
- Situated LP: hook rules up to services [Grosof]
- Description LP: combine Description Logic ontologies [Grosof *et al.*]
- Courteous Inheritance: combine OO default ontologies [Grosof *et al.*]
- Production Rules as LP: interoperate [Grosof *et al.*]
  - Declarative LP as interoperable core between commercial families [Grosof *et al.*]
- Hypermonotonic Reasoning: combine with FOL [Grosof (in-progress)]

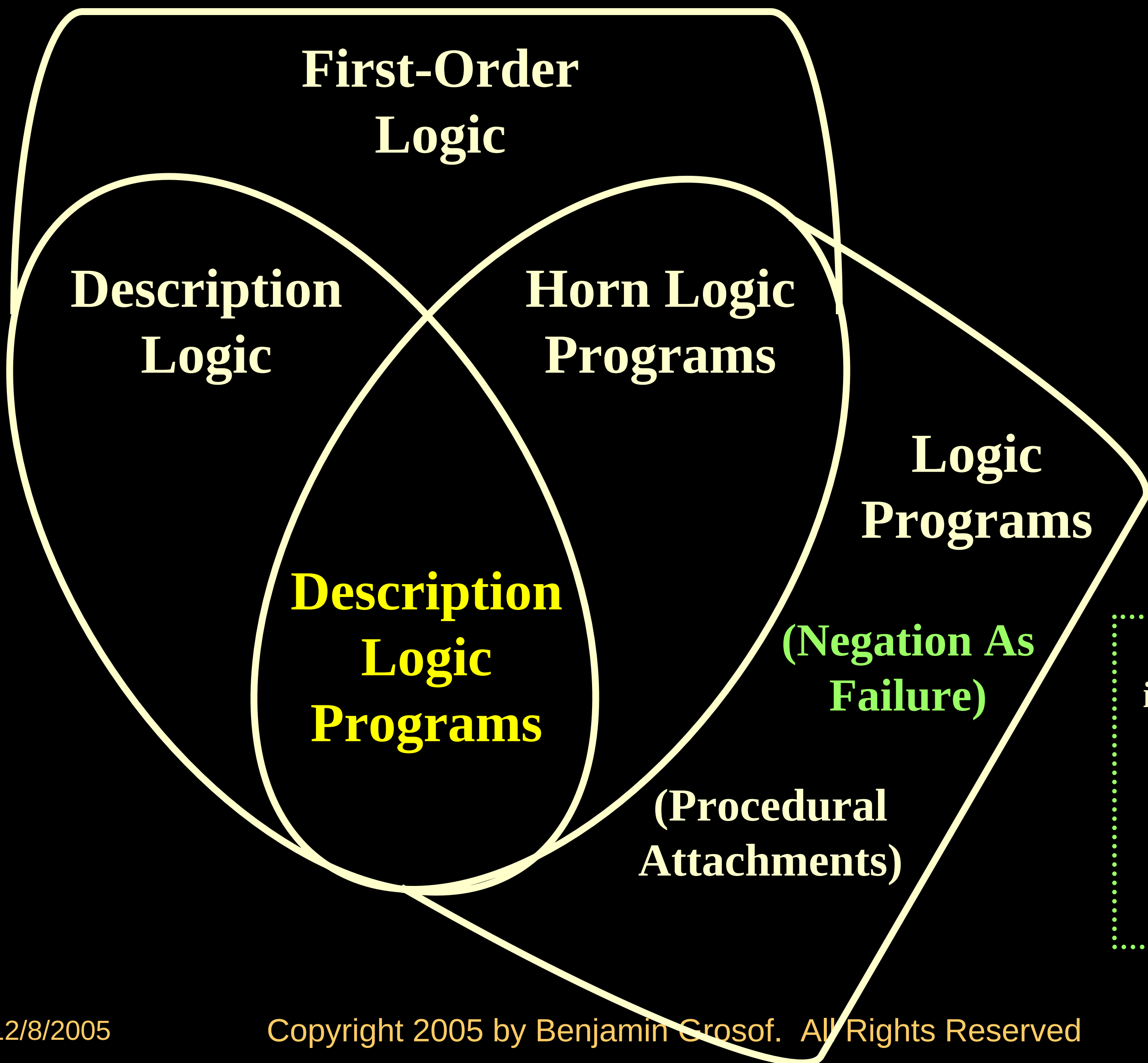
Production Logic Programs:
A New Fundamental Rule KR Approach

In 2005:

- Production extension of LP:
  - actions and tests appear directly within rules (procedural attachments)
  - Generalizes Situated LP a bit, and reformulates it more familiarly
- Theory & algorithms achieving semantic interoperability of
  {core Production Rules} ← → declarative LP
  - Handles negation correctly, by stratifying PR agenda control strategy
  - 1st declarative semantics for Production Rules

- Combines with all the other features: Courteous, …
- → "Production LP" as umbrella LP KR approach

Venn Diagram: Expressive Overlaps among KR's



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

Criteria for Interchangeable Semantic Rule Representation

1

- *High-level:* Agents reach common understanding; contract is easily modifiable, communicatable, executable.

2

- Inter-operate: heterogeneous commercially important rule systems.
- Expressive power, convenience, natural-ness.
- ... but: computational <u>tractability</u>.
- <u>Modularity</u> and locality in revision.

3

- <u>Declarative</u> semantics.
- Logical non-monotonicity: default rules, negation-as-failure.
  - essential feature in commercially important rule systems.
- Prioritized conflict handling.
- Ease of parsing.
- Integration into Web-world software engineering.
- Procedural attachments.

OLP

Courteous

XML

Situated

New Analysis: Key Technical Requirements for SWS

- 1. Combine rules with ontologies, from many web sources, with:
  - Rules on top of ontologies
  - Interoperability of heterogeneous rule and ontology systems
  - Power in inferencing
  - Consistency wrt inferencing
  - Scaleability of inferencing

- 2. Hook rules (with ontologies) up to web services
  - Ex. web services: enterprise applications, databases
  - Rules use services, e.g., to query, message, act with side-effects
  - Rules constitute services executably, e.g., workflow-y business processes
  - Rules describe services non-executably, e.g., for discovery, deal negotiation
  - On top of web service process models, coherently despite evolving messiness

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

- 1. "Death. Taxes. Integration." - They're always with us.

- 2. "Business processes require communication between organizations / applications." - Data and programs cross org./app. boundaries, both intra- and inter- enterprise.

- 3. "It's the *automated knowledge* economy, stupid!" - The world is moving towards a knowledge economy. And it's moving towards deeper and broader automation of business processes. The first step is automating the use of <u>structured</u> knowledge.
  - Theme: *reuse* of knowledge across multiple tasks/app's/org's

The Semantic Web

- The 1st generation, the Internet, enabled disparate machines to exchange data.
- The 2nd generation, the World Wide Web, enabled new applications on top of the growing Internet, making enormous amounts of information available, in human-readable form, and allowing a revolution in new applications, environments, and B2C e-commerce.

- The next generation of the net is an "agent-enabled" resource (the "Semantic Web") which makes a huge amount of information available in machine-readable form creating a revolution in new applications, environments, and B2B/EAI e-commerce.

…by enabling "agent" communication at a Web-wide scale.

  - "Agent" = knowledge-based application

Next Generation Web

Semantic Web Services

Semantic Web techniques

Automated Knowledge Bases

Rules (RuleML)

Ontologies (OWL)

Databases (SQL, XQuery, RDF)

Web Services techniques

API's on Web

(WSDL, SOAP)

Two interwoven aspects:
Program: Web Services
Data: Semantic Web

XML

First Generation Web

OPTIONAL SLIDES III FOLLOW

Overview of RuleML Today I

- RuleML Initiative (2000--) http://www.ruleml.org
  - Dozens of institutions (~35), researchers; esp. in US+Canada, EU
- Mission priorities:
  1. Enable semantic exchange of rules/facts between most commercially important rule systems
     - Production rules, relational databases, Prolog, Event-Condition-Action rules
  2. Synergize with RDF, OWL (& other relevant web standards as arrive)
  3. Enable rule-based semantic web services, e.g., policies
- Standards specification: current version V0.8+
  - 1st version 2001; basic now fairly stable
- Logical Knowledge Representation at core of semantics
  - Declarative Logic Programs (LP) & First Order Logic (FOL) … Webized
  - Firm foundations in decades of R&D theory, algorithms, implementations

Overview of RuleML Today II

- A number of tools (~60 engines, translators, editors), demo applications. E.g., SweetRules open source platform.
- Very influential & lots of mindshare in cutting edge R&D community. 20,000+ Google Hits (as of Mar. 2004)
- Annual International Scientific Workshop since 2002
- Cooperating closely with the leading umbrella Web standards organizations and SW research efforts:
  - OMG – providing markup and semantics for production rules meta-model.
  - Discussions well underway to launch Oasis, W3C standards working groups.
  - Encouraged (and funded in part) by DARPA
  - Collaborating with Semantic Web Services Initiative (SWSL), Web Services Mediation Language (WSML) & REWERSE in EU

Overview of RuleML Today III

- Logic Programs is a Fully Declarative KR (not simply Prolog!)
  - Well-established logic with model theory
  - Available algorithms, implementations
  - Close connection to relational DB's
    - core SQL is Datalog Horn LP
- Abstract graph syntax
  - 1st encoded in XML…
  - … then RDF … also a presentation syntax for human editing
- Expressive Extensions incrementally, esp. already:
  - Non-monotonicity: Negation as failure; Courteous priorities
  - Procedural Attachments: Situated actions/effecting, tests/sensing
  - Hilog, frame syntax,
  - *In-progress:*
    - reification cf. F-Logic Programs, SWSL
    - Events cf. Event-Condition-Action

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

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

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

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

SWS Adoption Roadmap: Some Strategy Considerations

- "Death. Taxes. Integration."
- Expect see beginning in a lot of B2B interoperability or heterogeneous-info-integration intensive (e.g., finance, travel)
  - Actually, probably 1st intra-enterprise, e.g., <u>EAI</u>
- Reduce costs of <u>communication</u> in procurement, operations, customer service, supply chain ordering and logistics
- <u>Agility</u>/speed/flexibility in business processes, supply chains
- "Killer app" target known for 30 years: do better job of EDI

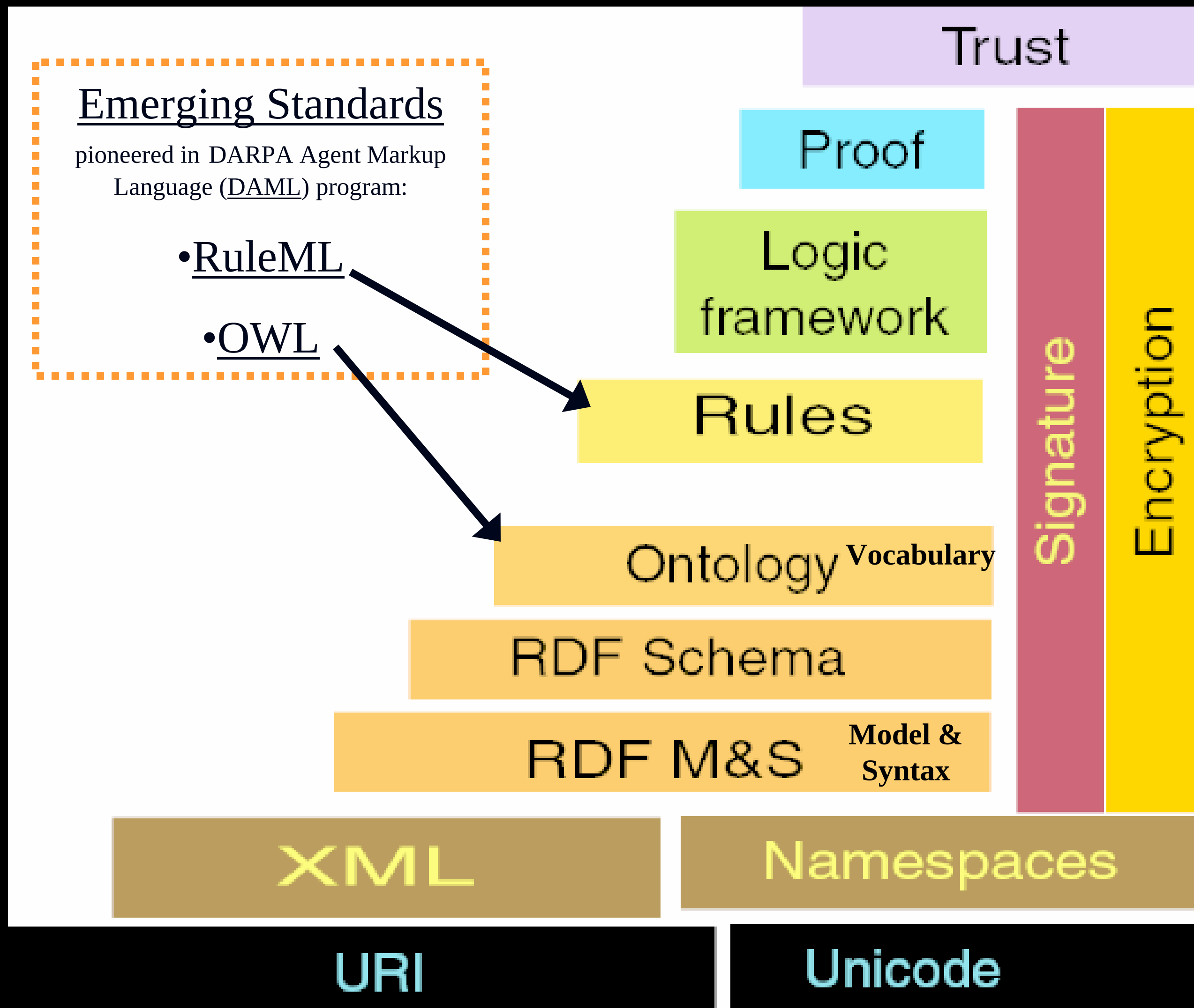
SweetRules Overview

- Concept and Architecture: Open Source Tools Platform for SW Rules and RuleML. http://sweetrules.projects.semwebcentral.org (2004-)
    - Multi-institutional collaboration led by MIT Sloan, with 12+ other co.'s / univ.'s
- Capabilities:
  - Translation and interoperability between heterogeneous rule systems (forward- and backward-chaining) and their rule languages/representations of the most commercially important flavors (relational database / Prolog and production rules / event-condition-action)
  - 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, OO default inheritance
  - Focus on kinds of rule systems that are commercially important
    - E.g., Jess production rules, XSB Prolog, IBM Common Rules, HP Jena, …
  - Highly scaleable performance by piggybacking on mature commercial implementations (e.g., Jess, XSB)
  - Automatically composes translators, inference engines

SweetRules V2.0 Fundamental KR

- Fundamental KR: Situated Courteous Logic Programs (SCLP) KR = Knowledge Representation
  - 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

(Older) W3C Semantic Web "Stack": Standardization Steps



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

Changes in W3C SW Stack Diagram

- 2002: added rules
  - Based on RuleML and DAML rules research
- 2005: added DLP as common core layer under both Rules and OWL
  - Based on Description LP research, SweetRules, and W3C Workshop discussion consensus

SW Stack: Acronym Expansion

- W3C = World Wide Web Consortium: umbrella standards body
- XML-S: XML Schema, i.e., basic XML spec
- RDF: Resource Description Framework:
  - W3C Working Group
  - Labelled directed graph syntax
  - Good for building knowledge representation on top of: simpler, more powerful than basic XML
  - M&S = Model and Syntax
  - RDF Schema = extension: simple class hierarchies
- Ontology = formally defined vocabulary & class hierarchy, generalizes Entity-Relationship models
  - OWL = W3C Web Ontologies Working Language
  - … based closely on DAML+OIL

SW Overall Dependencies

- The W3C "stack" picture is a rough simplification.

- Rules do <u>not</u> require RDF
  - Can just use XML or even an ASCII "presentation syntax"

- Ontologies do <u>not</u> require RDF <u>nor</u> OWL
  - There are other techniques; OWL lacks some features
    - OWL <u>does</u> require RDF

- Customers and major vendors will be still digesting <u>XML</u> data management in next 2-5 years
  - … before moving on to heavy <u>RDF</u> usage

Web Service -- definition

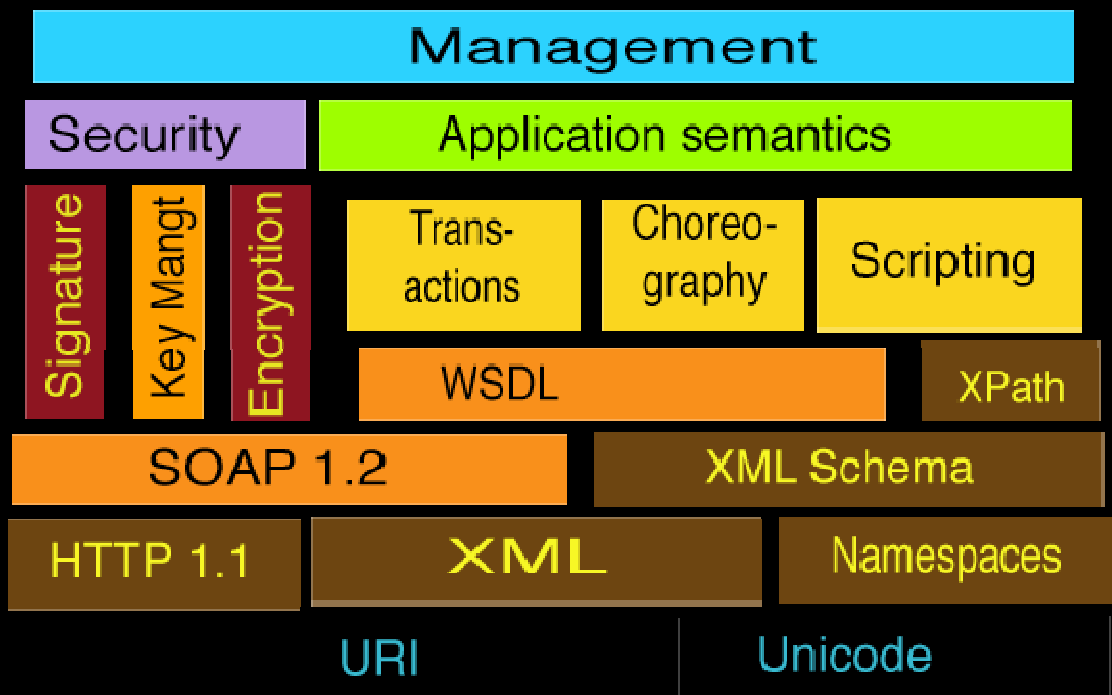
- *(For purposes of this talk:)*

- A procedure/method that is invoked through a Web protocol interface, typically with XML inputs and outputs

Semantic Web Services

- Convergence of Semantic Web and Web Services
- Consensus definition and conceptualization still forming
- Semantic (Web Services):
  - Knowledge-based service descriptions, deals
    - Discovery/search, invocation, negotiation, selection, composition, execution, monitoring, verification
    - <u>Advantage: **reuse** of knowledge across app's, these tasks</u>
  - Integrated knowledge
- (Semantic Web) Services: e.g., infrastructural
  - Knowledge/info/DB integration
  - Inferencing and translation

Web Services Stack outline

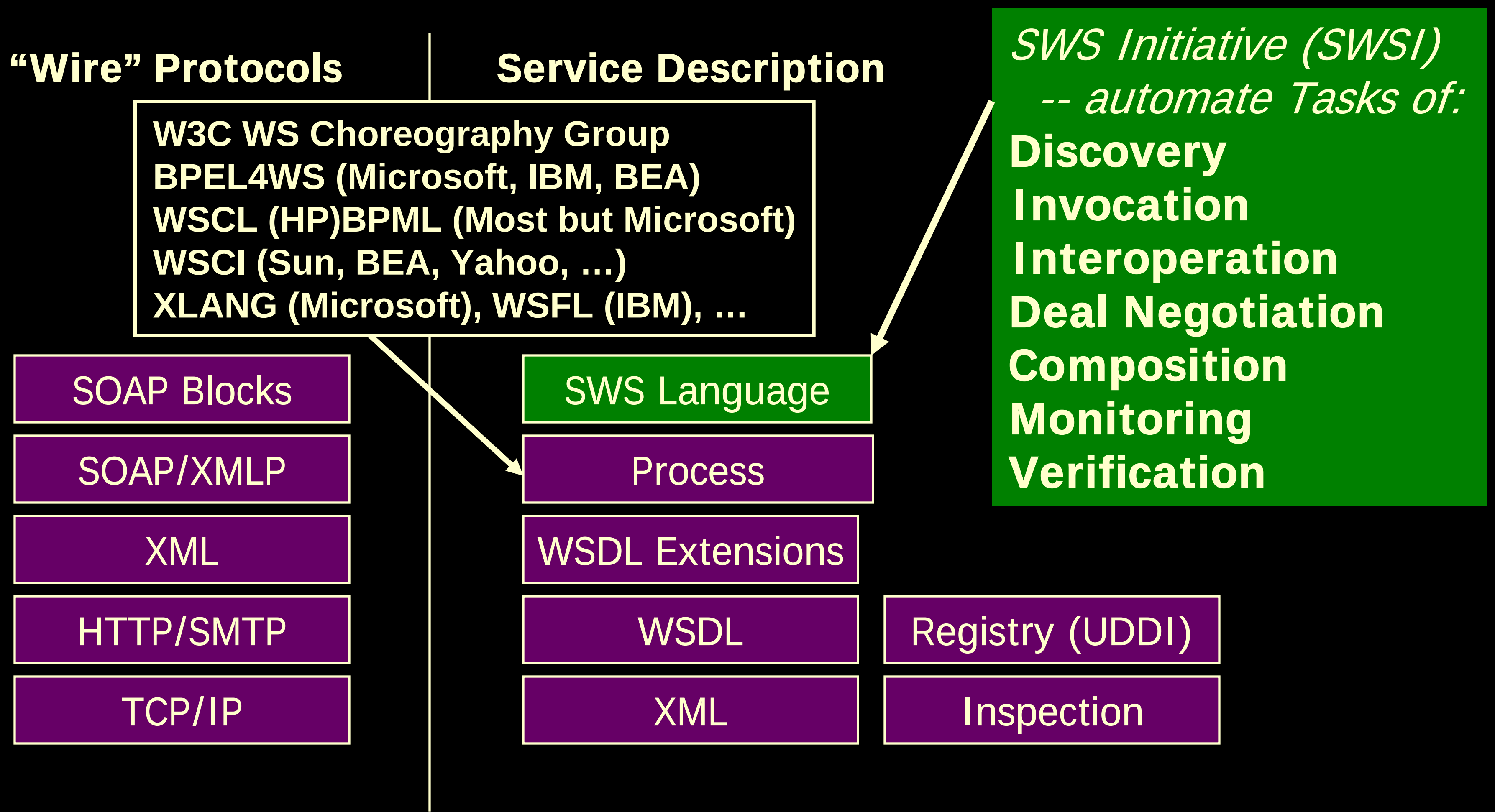


NOTES:

WSDL is a Modular Interface spec
SOAP is Messaging and Runtime
Also:
- UDDI is for Discovery
- BPEL4WS, WSCI, … are for transactions
- Routing, concurrency, …

Diagram courtesy Tim Berners-Lee: http://www.w3.org/2004/Talks/0309-ws-sw-tbl/slide6-0.html

SWS Language effort, on top of Current WS Standards Stack



[Slide authors: Benjamin Grosof (MIT Sloan), Sheila McIlraith (Stanford) , David Martin (SRI International), James Snell (IBM)]

WS Stack: some Acronym Expansion

- SOAP = simple protocol for XML messaging
- WSDL = protocol for basic invocation of Web Services, their input and output types in XML
- Choreography = higher-level application interaction protocols in terms of sequences of exchanged message types, contingent branching
  - There's now a W3C Working Group
- "Agreement" here = agreement between invoker and provider of the service, described at knowledge level
- *Overall: in 2001-2002 lots of proprietary jockeying and de-facto mode testing/pressuring of the open-consortial standards bodies (e.g., of W3C) "riding the tiger". Then more via W3C, Oasis starting in 2003.*

SWS Tasks at higher layers of WS stack

Automation of:

- Web service <u>discovery</u>
  Find me a shipping service that will transport frozen vegetables from San Francisco to Tuktoyuktuk.

- Web service <u>invocation</u>
  Buy me "Harry Potter and the Philosopher's Stone" at <u>www.amazon.com</u>

- Web service <u>deals</u>, i.e., contracts, and their <u>negotiation</u>
  Propose a price with shipping details for used Dell laptops to Sue Smith.

- Web service <u>selection</u>, <u>composition</u> and <u>interoperation</u>
  Make the travel arrangements for my WWW11 conference.

[Modification of slide also by Sheila McIlraith (Stanford) and David Martin (SRI International)]

OPTIONAL SLIDES IV FOLLOW

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.
- Known advantages of rules vs. general code
  - separable business logic, more reusable across app.'s, life cycle
  - good for loose coupling cf. workflow
  - good for representing contingent behavior of services/processes.
  - 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.

Rule-based Semantic (Web) Services

- Rules/LP in appropriate combination with DL as KR, for RSWS
  - DL good for <u>categorizing</u>: a service overall, its inputs, its outputs
- Rules to describe <u>service process models</u>
  - rules good for representing:
    - <u>preconditions</u> and <u>postconditions</u>, their contingent relationships
    - <u>contingent</u> behavior/features of the service more generally,
      - e.g., exceptions/problems
  - familiarity and naturalness of rules to software/knowledge engineers
- Rules to specify <u>deals about services</u>: cf. e-contracting.

Rule-based Semantic Web Services

- Rules often good to executably specify service process models
  - e.g., business process automation using procedural attachments to perform side-effectful/state-changing actions ("effectors" triggered by drawing of conclusions)
  - e.g., rules obtain info via procedural attachments ("sensors" test rule conditions)
  - e.g., rules for knowledge translation or inferencing
  - e.g., info services exposing relational DBs

- Infrastructural: rule system functionality as services:
  - e.g., inferencing, translation

Some Semantic Web Advantages for Biz

- Builds upon XML's much greater capabilities (vs. HTML*) for structured detailed descriptions that can be processed automatically.
  - Eases application development effort for assimilation of data in inter-enterprise interchange
- Knowledge-Based E-Markets -- where Agents Communicate

(Agent = knowledge-based application)

  - ∴ potential to revolutionize *interactivity* in Web marketplaces: B2B, …
- Reuse same knowledge for multiple purposes/tasks/app's
  - Exploit declarative KR; Schemas

- * new version of HTML itself is now just a special case of XML