Rules in the Semantic Web Services Language:

An Overview for Standardization Directions

by Benjamin Grosof, Michael Kifer,** and David Martin****

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

***SUNY Stonybrook, http://www.cs.sunysb.edu/~kifer*

****SRI International, http://www.ai.sri.com/people/martin*

Presented at W3C Workshop on Rules for Interoperability
(held Nov. 30 – Dec. 2),
Apr. 27, 2005, Washington, DC, USA
http://www.w3.org/2004/12/rules-ws

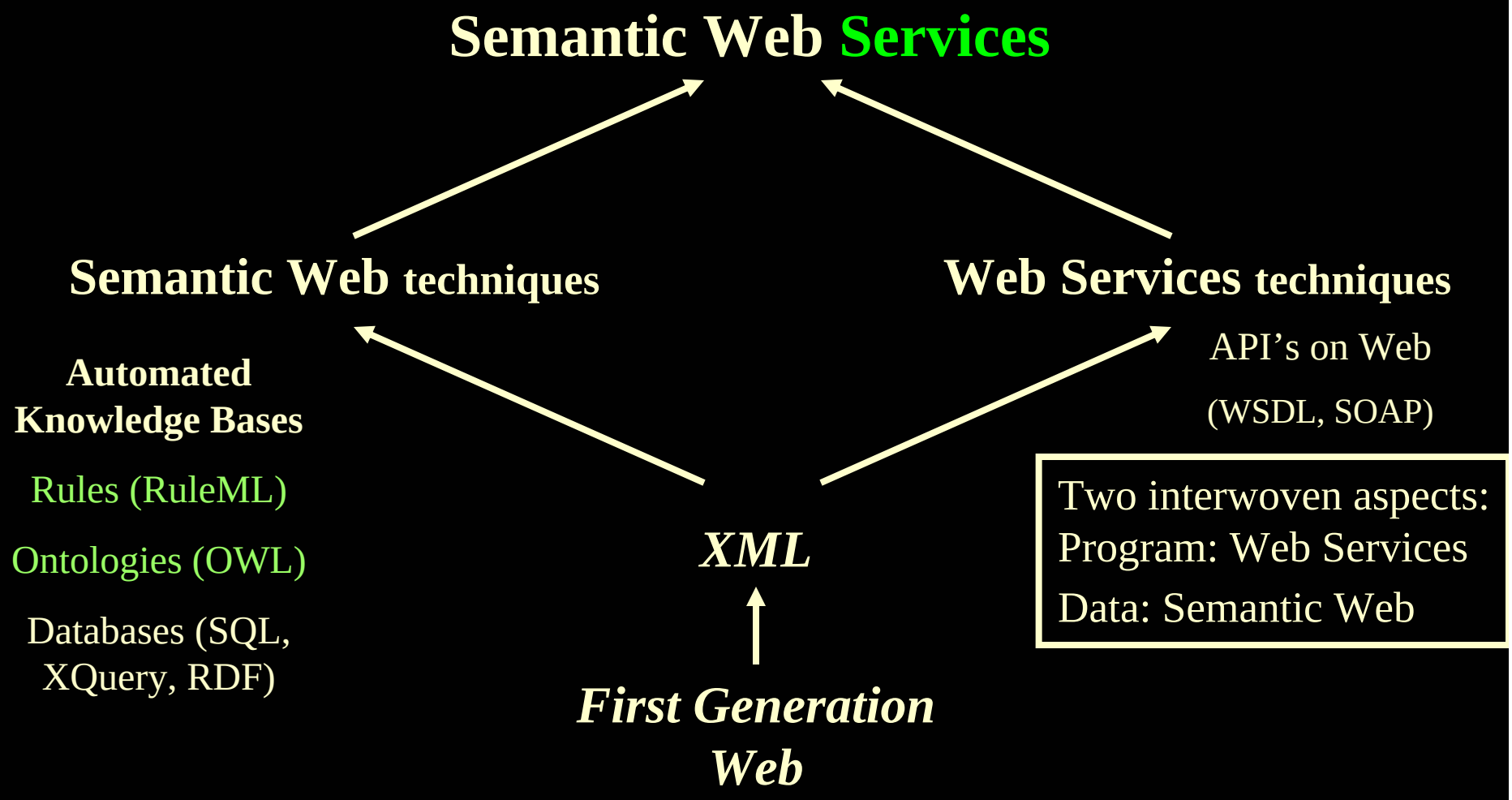
Talk Mode: the Firehose

Short time: ⇒ Some slides skimmed

Outline

- Intro:
  - What is Semantic Web Services
  - Semantic Web Services Initiative & its Language effort
- SWSL Strategy and Design Overall
- SWSL Applications Requirements and Scenarios
- Rule Language Design
  - Feature Layers:
    - Basic: Datalog Horn + NAF + webizing
    - Additional: Courteous, Frame Syntax, …
  - Integrating Ontologies
- Roots of SWSL Rules
  - Implementation techniques & platforms: SweetRules, Flora
  - Business Uses and Value of Semantic, Web Rules

Next Generation Web



SWSL Effort

- Semantic Web Services Initiative (SWSI) was founded in late 2002, to coordinate research and early standards work world-wide in area of Semantic Web Services (SWS)
  - Incl. DAML, WSMO
- Language Committee, Architecture Committee
  - Several dozen active participants
- Industrial Partners too: about 40 companies
- Design Report Apr. 2005: (incl. Requirements)
  - Language
  - Ontology (Core)
  - Application Scenarios

http://www.swsi.org

http://www.daml.org/services/swsl/report

- quick tour

This document, and the documents linked below, are DRAFTs, and will undergo further development before they become final.

Semantic Web Services Language and Ontology Proposals

DRAFT

This is the technical report of the Semantic Web Services Language (SWSL) Committee of the Semantic Web Services Initiative (SWSI). This report consists of the following four top-level documents, with five related appendices.

- Semantic Web Services Language and Ontology Overview
- The Semantic Web Services Language
- The Semantic Web Services Ontology
- Application Scenarios

Appendices (of the Overview document):

- Semantic Web Services Requirements

Appendices (of the Ontology document):

- PSL in SWSL-FOL and SWSL-Rules
- Axiomatization of the FLOWS Process Model
- Axiomatization of the Process Model in SWSL-Rules
- Reference Grammars

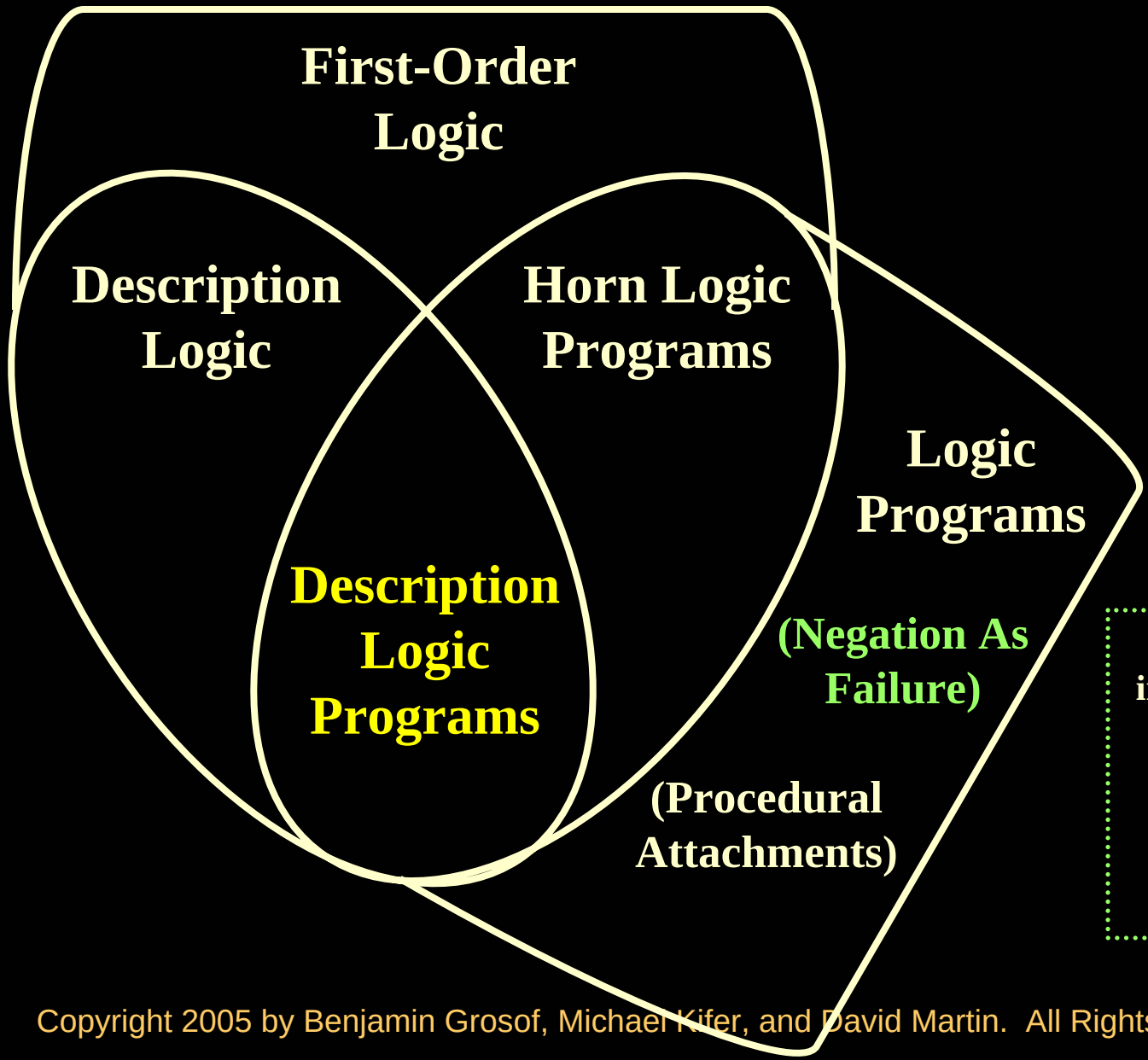
SWSL Goals and Strategy I

- Develop approach to specifying
  semantic descriptions of services – i.e., processes
  - Language for rules and ontologies, including for policies and process models
  - Application scenarios
  - Requirements analysis underpinning these
- Extend out from OWL-S approach
  - Provide particular core service ontologies
- Complement and draw upon WSMO

SWSL Strategy II

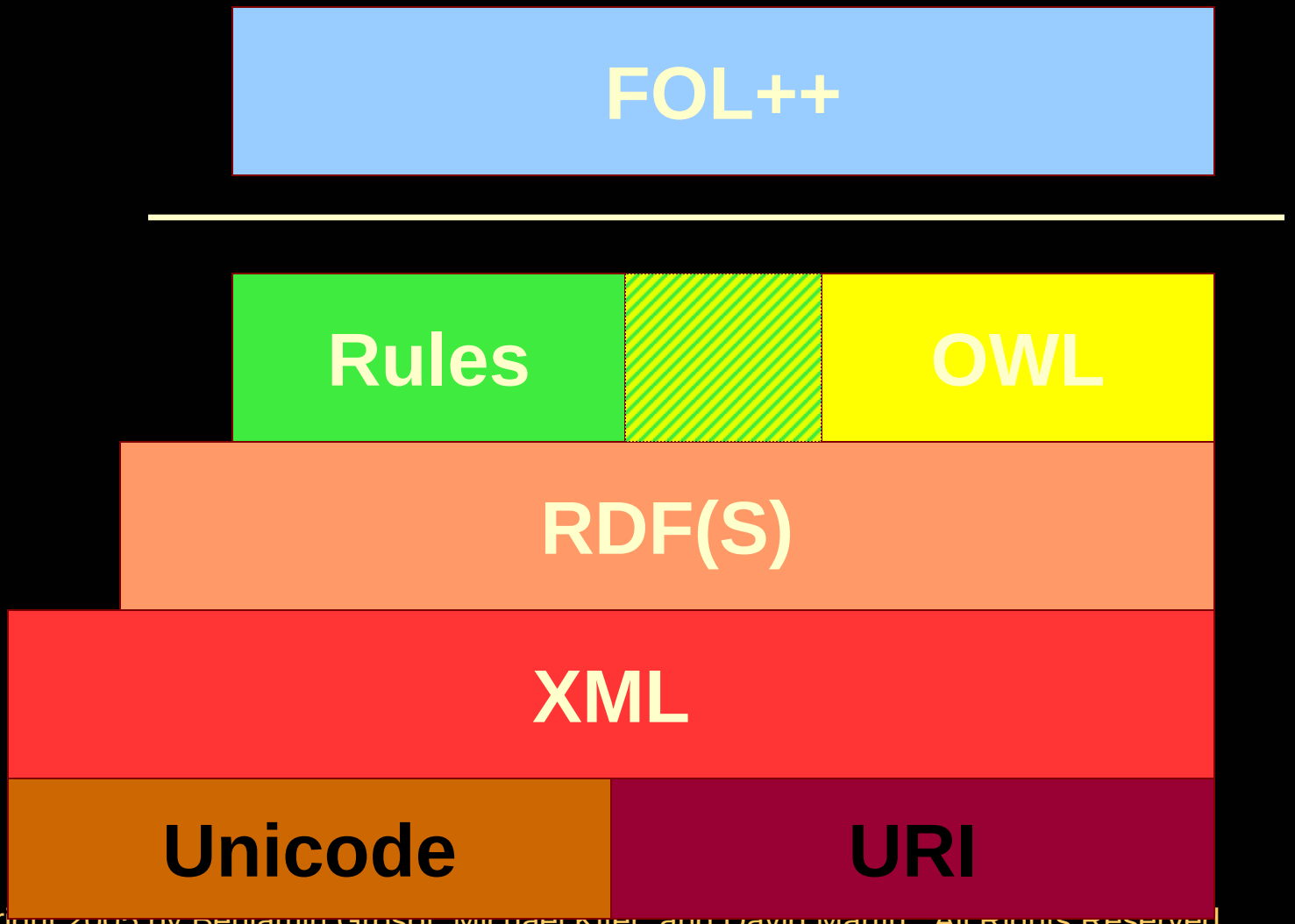
- Overall:
  - take advantage of more expressive languages
    - Use well-understood logical knowledge representation (KR), esp. rules + ontologies
  - extend the conceptual model of services
- Leverage broad availability of LP-based Rule languages, environments, tools, etc.
- For ontologies:
  - Allow full FOL expressiveness
    - for process modeling ontologies, verification/analysis reasoning
  - Maintain connections with the world of OWL
  - For providing core service ontologies:
    - Build on mature conceptual models
      - PSL, Dublin core, W3C architecture
- Layers of expressiveness

Venn Diagram: Expressive Overlaps among KR's



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

The Web Rule Language in its Context: RuleML, WSMO, SWSL View



Technical Requirements for SWSL-Rules

- 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 Web 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 *immediately* important: triggering of side-effectful actions (cf. Situated LP effecting or Transaction Logic)

SWSL Rule Language Design Overall

- Language design uses, and becomes, RuleML (LP)
  - Adopted RuleML for markup syntax
  - Extended RuleML, in collaboration with RuleML Initiative
  - Developed new presentation syntax, ditto
    - To create and communicate examples to drive SWSI design
- Language has two parts; these share: many features, most syntax
  - "SWSL-Rules" = the LP KR.
  - "SWSL-FOL" = the FOL KR.

Outline

- Intro:
  - What is Semantic Web Services
  - Semantic Web Services Initiative & its Language effort
- SWSL Strategy and Design Overall
- SWSL Applications Requirements and Scenarios
- Rule Language Design
  - Feature Layers:
    - Basic: Datalog Horn + NAF + webizing
    - Additional: Courteous, Frame Syntax, …
  - Integrating Ontologies
- Roots of SWSL Rules
  - Implementation techniques & platforms: SweetRules, Flora
  - Business Uses and Value of Semantic, Web Rules

SWSL Applications Requirements Analysis: 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)

Application Scenarios in SWSL Report

- The Amazon E-commerce Service
  - Amazon & partners heavily uses web services today
- Policy Rules for E-Commerce
- Using Defaults in Domain-Specific Service Ontologies
  - Sell Product, its specializations
- Service Discovery with SWSL-Rules
  - …
- Additional referenced:
  - SweetDeal e-contracting in SCM, auctions, etailing,
  - …

Zoom-in: Policy Rules for E-Commerce Application Scenarios in SWSL Report

- Several Examples:
  - price discounting
  - refunds supply chain ordering lead time
  - creditworthiness
  - credit card transaction authorization
- Include B2C, B2B, retailing, supply-chain, several industry domains
  - books, appliances, manufacturing, …
- Each of these policies is useful for not just one but several different kinds of tasks within an application realm.
  - E.g., refund rules ⇒ advertising
    ⇒ contract negotiation/formation
    ⇒ monitoring / exception-handling, process execution

Zoom-in: Refund Policy Rules

```
/* refund policy rules */

{unconditionalGuarantee}
   refund(?Return, percent(90))
         :- return(?Return) and delay(?Return, ?D) and
               lessThanOrEqual(?D, days(30)).

{defectiveGuarantee}
   refund(?Return, percent100)
         :- return(?Return) and reason(?Return, defective) and
              delay(?Return, ?D) and lessThanOrEqual(?D, years(1)).

overrides(defectiveGuarantee, unconditionalGuarantee).

!- refund(?Refund, percent(90)) and refund(?Refund, percent(100)).
```

Background: Courteous Feature of LP

- Prioritized conflict handling
- Fully declarative semantics
- Guaranteed consistency of set of conclusions
  - Can escalate unresolved conflicts
- Aids modular merging, updating, authoring
- Enables robust knowledge merging over Web, between organizations (EAI, B2B, …)
- Preserves tractability of LP (similar scaleability to relational databases)
- Implementable via modular add-on ("courteous compiler") to any rule system supporting NAF
  - E.g., Jess production rules, XSB Prolog
  - Reference implementation in open source [SweetRules]

Outline

- Intro:
  - What is Semantic Web Services
  - Semantic Web Services Initiative & its Language effort
- SWSL Strategy and Design Overall
- SWSL Applications Requirements and Scenarios
- Rule Language Design
  - Feature Layers:
    - Basic: Datalog Horn + NAF + webizing
    - Additional: Courteous, Frame Syntax, …
  - Integrating Ontologies
- Roots of SWSL Rules
  - Implementation techniques & platforms: SweetRules, Flora
  - Business Uses and Value of Semantic, Web Rules

Current SWSL Rule Language Layers

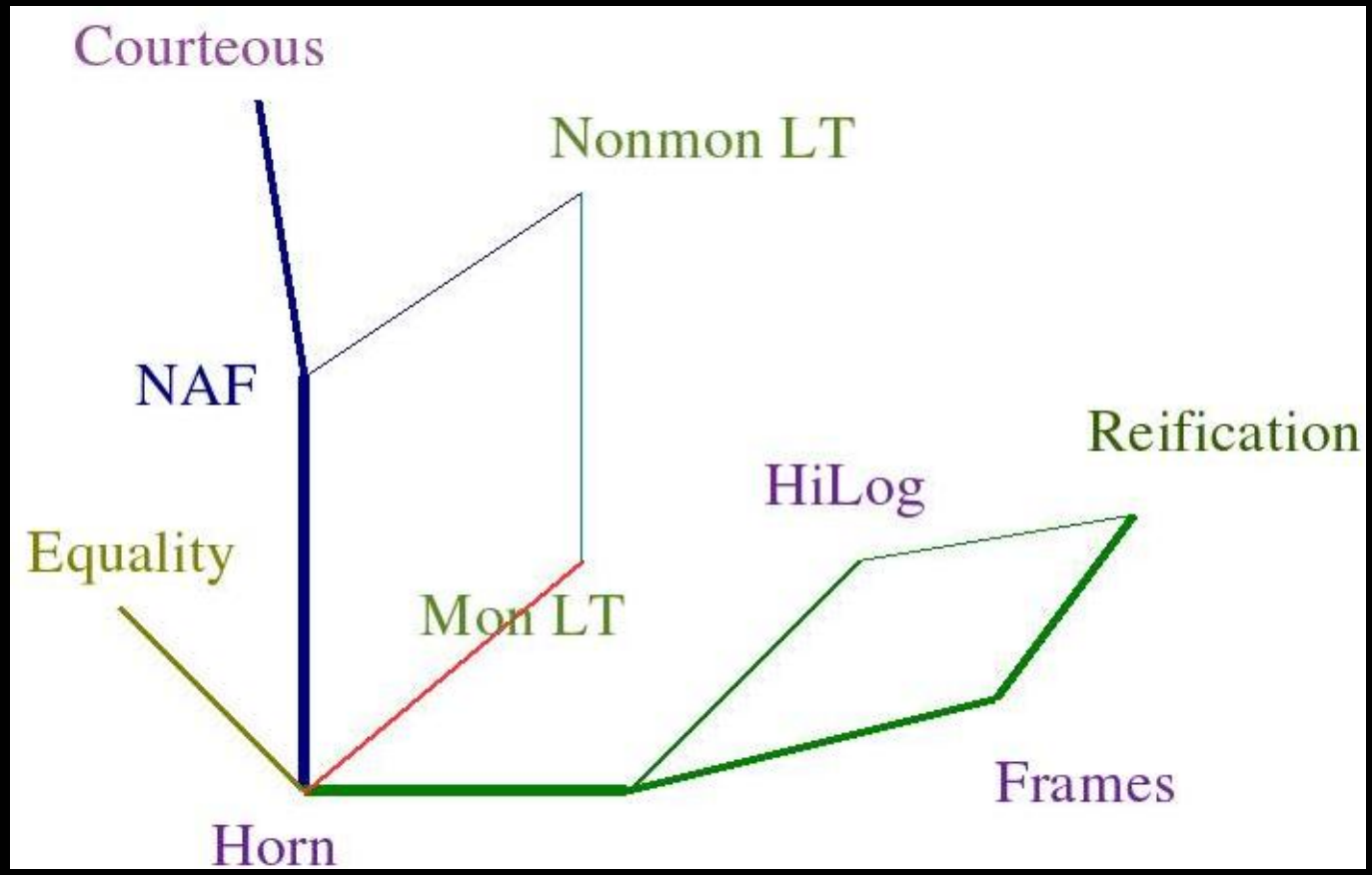
- Layers of expressiveness in SWSL-Rules
  - Basic:
    - Datalog Horn + NAF + webized names
  - Additional features:
    - Courteous, Frame/slotted syntax, Datatyping, Signature declarations, logical functions & skolemization, Integrity constraints, Lloyd-Topor, Hilog, basic reification, basic user equality
    - .. The same list as you saw for RuleML; except pushed off for future is …

Future SWSL Rule Language Layers

- Procedural attachments for actions/effecting and tests/sensing, cf.
  - Production Rules, Event-Condition-Action Rules

  $\Rightarrow$ as KR:
  - Situated LP, Production LP [Grosof et al.]
  - Transaction Logic [Kifer et al.]
- Also misc. other: e.g., aggregation, modules, …

SWSL-Rules Layering Diagram



Integrating Ontologies with Rules

- Kinds of Ontologies:
  - Description Logic, OWL
  - FOL, e.g., PSL (NIST Process Specification Lang.)
  - OO with default inheritance, e.g.,
    - Process Handbook (5000 business processes)
    - C++, Java, C#
- Approaches:
  1. Description Logic Programs
  2. Use Nonmon LP, preferably Courteous
     - Represent inheritance
     - Hypermonotonic mapping of FOL into Courteous LP
       - Uses new theory. Employed to create part of SWSL Rules core Ontology

Outline

- Intro:
  - What is Semantic Web Services
  - Semantic Web Services Initiative & its Language effort
- SWSL Strategy and Design Overall
- SWSL Applications Requirements and Scenarios
- Rule Language Design
  - Feature Layers:
    - Basic: Datalog Horn + NAF + webizing
    - Additional: Courteous, Frame Syntax, …
  - Integrating Ontologies
- Roots of SWSL Rules
  - Implementation techniques & platforms: SweetRules, Flora
  - Business Uses and Value of Semantic, Web Rules

ROOTS OF SWSL RULES

Implementability of this vision?

- That's a lot of features
- What's available now?
- Does this really get us interoperability among commercially important flavors of rules?

Example Implementations
supporting Rich Rules features cf. SWSL/RuleML

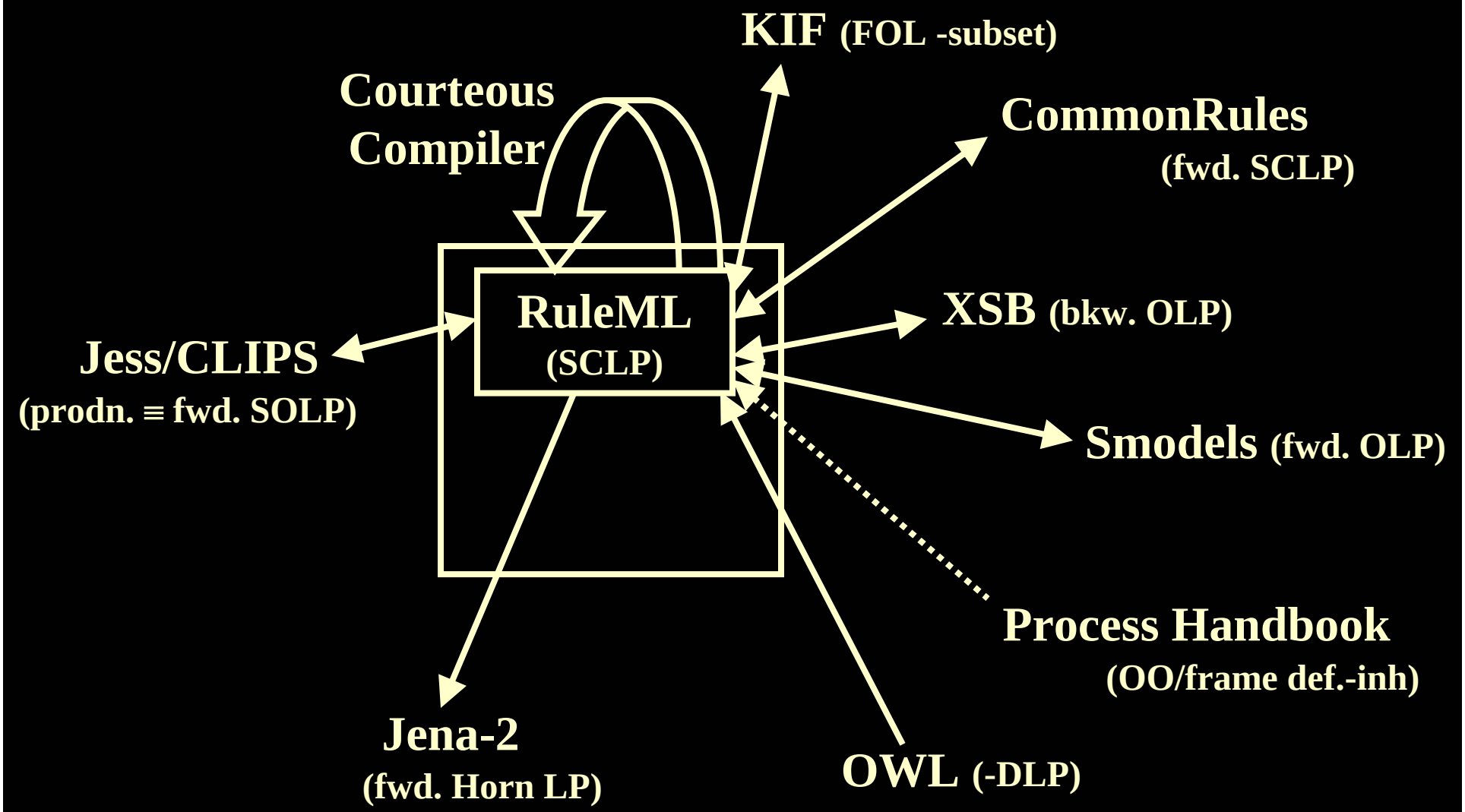
- Example open source implementations:
  - Reference implementation of techniques, e.g., features, translation
  - Joint the party to get transitive interoperability
- SweetRules open source
  http://sweetrules.projects.semwebcentral.org
  - RuleML centric
  - Whole platform incl. translation, inferencing, etc.; pluggable
    - Multi-institutional: MIT (lead), UMBC, BBN, Stanford, U. Zurich; cooperation from IBM, HP, U. Karlsruhe, SUNY Stonybrook, Sandia, …

Example Implementation Platforms, continued

- Courteous prioritized conflict handling, incl. Compiler
- Combination with OWL (Description LP subset)
- Also: Situated LP, Production Rules interoperability
  - WSDL Web Services, or Java methods, as actions or tests
  - Correct NAF (cf. LP) in production rules
- Running semantic interoperability among heterogeneous commercially important languages/systems: XSB Prolog, Jess production rules, IBM CommonRules, HP Jena, OWL, KIF, Smodels, MIT Process Handbook

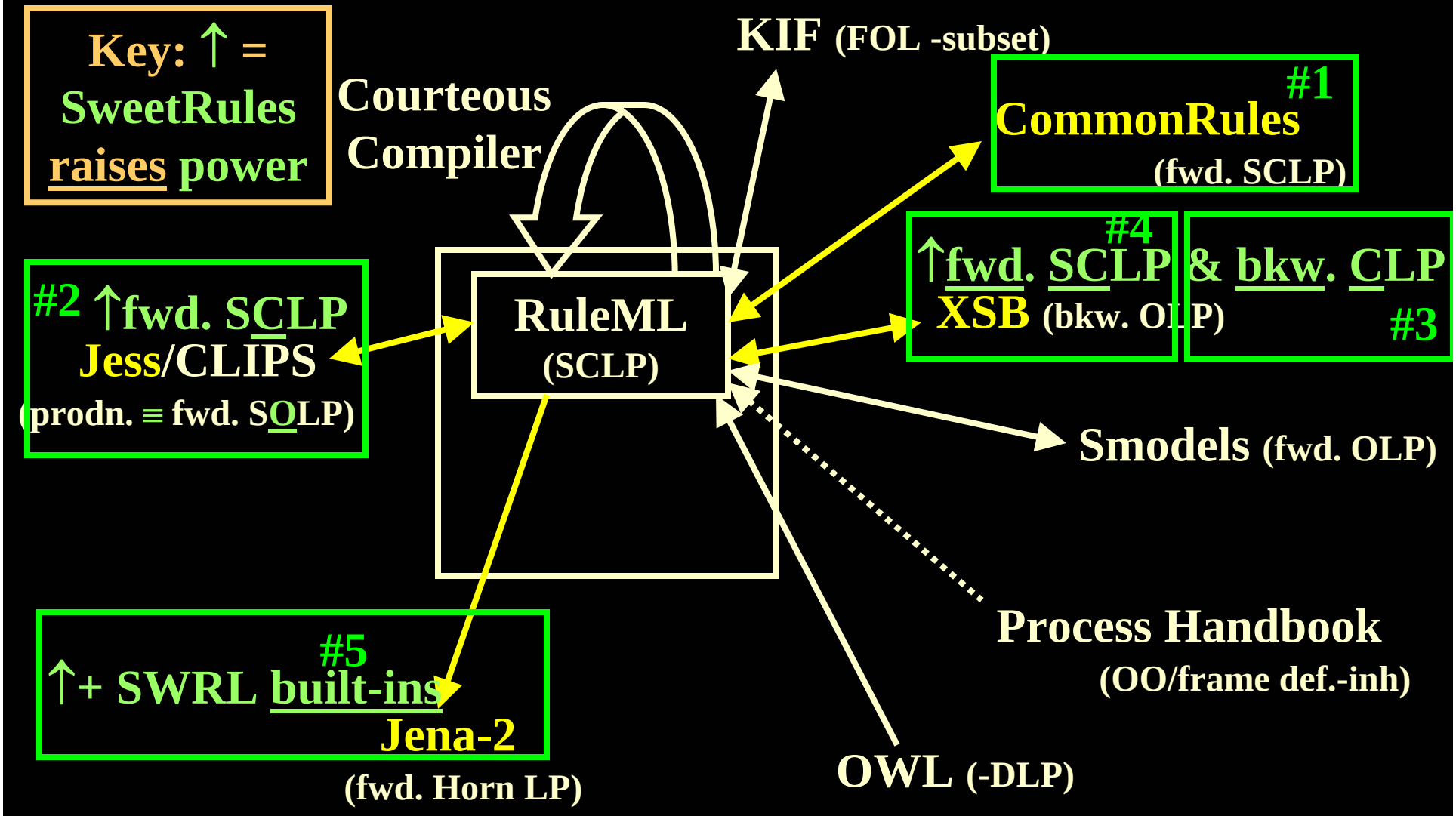
- Flora open source http://flora.sourceforge.net on top of XSB
  - Most of the other SWSL-Rules features: …
  - Frame syntax
  - Hilog, reification
  - … More …. also Transaction Logic, …

SweetRules V2.0 Translators Graph



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

SweetRules V2.0 New Inferencing Engines



(SCLP = Situated Courteous Logic Programs)

Development effort is not giant

for semantically interoperable web rules via RuleML

SweetRules experience:

- Only 2 person-years of effort to build
- ~25,000 lines of Java
  - Integrates systems with millions of lines of code
  - Avoids reengineering engine guts
- Loosely coupled, pluggable
- … yet semantically rigorous

$\Rightarrow$ deep interoperability

SweetRules & MIT RuleML Use Cases

- 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

Some Answers to: "Why do Semantic , Web Rules 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 structured knowledge.
  - Theme: *reuse* of knowledge across multiple tasks/app's/org's

Strategic Business Foci in our SW Research

- Knowledge-based Services Engineering: intra- and inter- enterprise
- Target "killer app" known for 30 years: do better job of EDI
- Challenges:
  - Ease of development, deployment ↑
  - Reuse of knowledge ↑
  - ⇒ life cycle costs ↓ , agility ↑
- Starting with: Policies
  - Using recent theory breakthroughs in semantic rules
  - E.g., for end-to-end contracting and authorization (incl. security)
- Starting with: EAI as well as B2B

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., EAI
- Reduce costs of communication in procurement, operations, customer service, supply chain ordering and logistics
- Agility/speed/flexibility in business processes, supply chains
- "Killer app" target known for 30 years: do better job of EDI

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

- Get the KR right
  - More mature research understanding; incl. a number of theory advances
  - Semantics independent of algorithm/implementation
  - Cleaner; avoid general programming/scripting language capabilities
  - Highly scaleable; high performance; better algorithms
  - Highly modular wrt updating; use prioritization
- 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

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)]

Objectives for Integrating Distributed SW Rules and Ontologies, Motivating SweetRules and its underlying theory+standards

BEFORE

AFTER

Contradictory conflict is globally contagious, invalidates all results.

⇒

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

Knowledge integration tackling the 5 D's (diversity, distributedness, disagreement, dynamism, & delay) is labor-intensive, slow, costly.

⇒

Knowledge integration is highly automated, faster, cheaper.

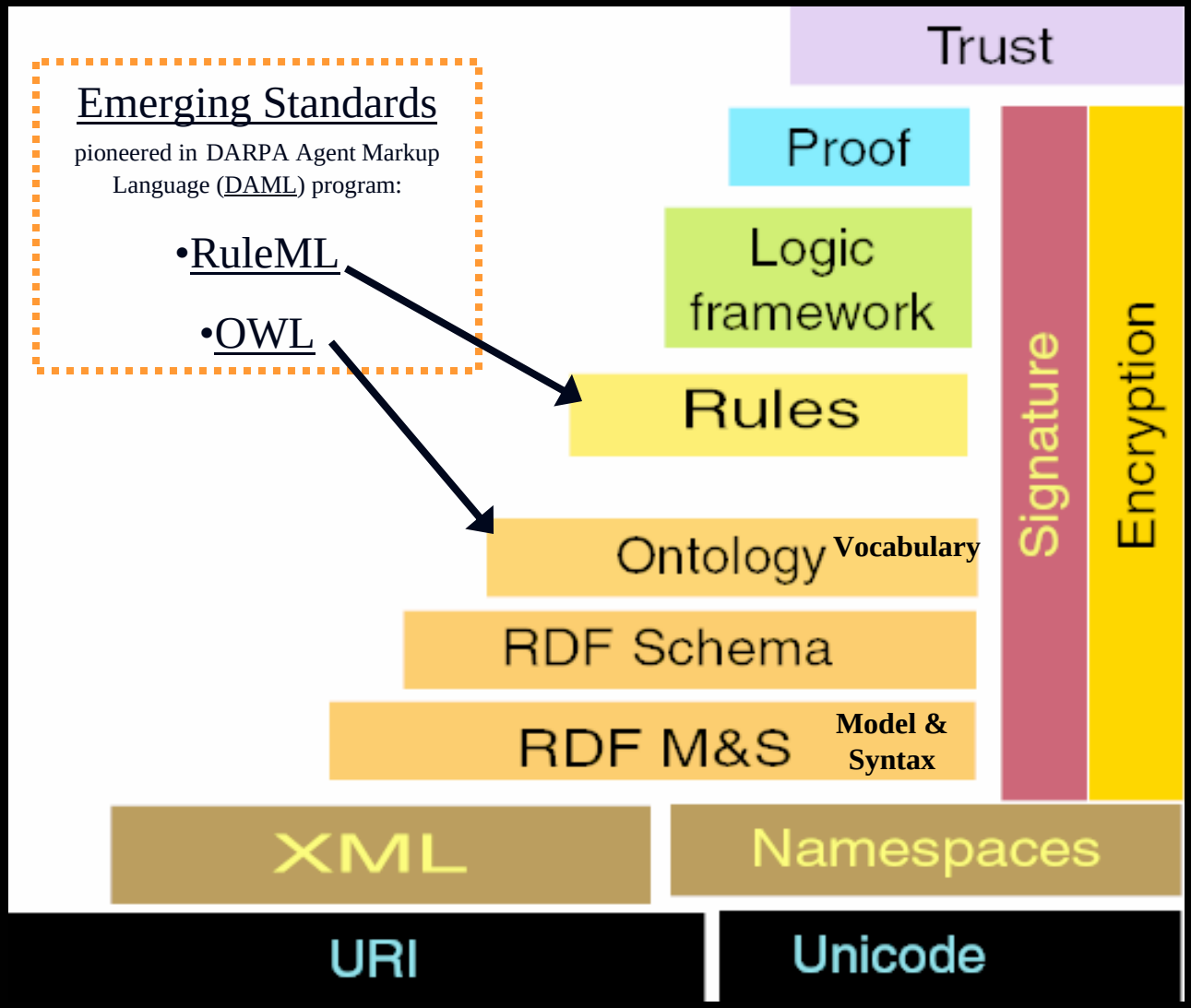
OPTIONAL SLIDES FOLLOW

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

W3C Semantic Web "Stack": Standardization Steps



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

SW Overall Dependencies

- The W3C “stack” picture is a rough simplification.
- Rules do not require RDF
  - Can just use XML or even an ASCII “presentation syntax”
- Ontologies do not require RDF nor OWL
  - There are other techniques; OWL lacks some features
    - OWL does require RDF
- Customers and major vendors will be still digesting XML data management in next 2-5 years
  - … before moving on to heavy RDF usage

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 entailment 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.

Example of Entailment: Mortality

- In First-Order Logic (FOL) KR:
  - Let P be the premises:
  - ∀?X. human(?X) ⇒ mortal(?X).
  - human(Socrates).
  -
  - In FOL, P entails (among others) the conclusion:
    - mortal(Socrates).
  - Notation:
    - "∀" 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)
    ← memberOf(?cust, AAA).
  - {seniord} discount(?cust, RamadaHotel, 25percent)
    ← age(?cust, ?x) and greaterThan(?x, 64).
  - overrides(seniord, loyald).
  - ⊥ ← discount(?c, ?y) and discount(?c, ?z) | (?y ≠ ?z).
  - memberOf(Faisal, AAA).
  - age(Faisal, 72).

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

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

Outline of some Points

- Intro: What is Semantic Web for Services
  - Knowledge Representation in XML; Agents; with Web Services
- Why it Matters for Business
  - Knowledge-based Services Engineering
  - Examples of Policies for Contracting and Authorization
    - Pricing, Comparison Shopping, Ordering Lead Time, …
- Semantic Rules: Technology and Standardization
  - RuleML, Theory Advances, SweetRules Open Source Platform
- Roadmapping Business Value and Market Evolution
  - Cheaper, Faster, Better; EAI and B2B; Early Verticals

Rationale: Why SWSL Effort Chose Nonmon LP in RuleML As Basis

- 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

OPTIONAL SLIDES END