Version of 4/21/09

Rules on the Web

WWW-2009 Conference Tutorial

Half-day (3 hours + break), 21 April 2009,
18th International World Wide Web Conference, Madrid, Spain

*by Benjamin Grosof *(presenter),*
*Mike Dean**, and Michael Kifer****

** Vulcan Inc. http://www.mit.edu/~bgrosof*
*** BBN Technologies http://www.daml.org/people/mdean*
**** Stony Brook University http://www.cs.sunysb.edu/~kifer*

Acknowledgements: Thanks for outline help from: Raphael Volz, Innovation Consulting GmbH http://raphaelvolz.de

WELCOME! to the WWW-2009 Tutorial

“Rules on the Web”

by Benjamin Grosof (presenter),
Mike Dean, and Michael Kifer

INSTRUCTIONS! All participants, please:

- Download the final-version tutorial slideset (updated since the preliminary web-posted version) at http://www.mit.edu/~bgrosof/#WWW2009RulesTutorial

- Sign in on the participants list (hard copy sheet) with your name, organization, email;
optionally also add your interests, homepage URL

Top-Level Outline of Tutorial

A. Introduction, Overview, and Uses

B. Concepts and Foundations

C. Conclusions and Directions

– References and Resources, as appendix

Background Assumed:

- basic knowledge of first-order logic, relational databases, XML

Quickie Bio of Presenter Benjamin Grosof

- Senior Research Program Manager, Vulcan, Inc. (Paul G. Allen's co.)
  - Leads Project Halo Advanced Research
  - Also advises Venture Capital arm (leading investor in the space)
- Principal, Benjamin Grosof & Associates (consulting)

- Pioneer/inventor of semantic rules for web and enterprises. Basis for:
  - Main web industry standards
    - W3C Rule Interchange Format (RIF) & Web Ontology Language (OWL) Rules
  - Oracle's semantic rules in flagship database software suite
  - Business applications pilots and strategy roadmaps
    - e-commerce; trust; finance; mobile; biomed; etc.

- 7 years as MIT professor in Info. Tech., at Sloan Mgmt. School
- 12 years at IBM T.J. Watson Research; 2 years at startups
- Ph.D. Stanford Computer Sci. (Artificial Intelligence); B.A. Harvard
- Led 2 released products at IBM, and major open source DARPA system
- 50+ scientific publications, 2 patents

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

Outline of Part B. Concepts & Foundations

1. Overview of Logical Knowledge Representations
   - First Order Logic (FOL), Logic Programs (LP)
2. Horn Case
3. Nonmonotonicity, Defaults, Negation, Priorities
   - Semantics for Default Negation
   - Courteous LP, Argumentation Theories
4. More Connectives and Quantifiers: Lloyd-Topor, Skolemization
5. Additional Features: Datatypes, Integrity Constraints, Equality, Aggregation
6. Procedural Attachments to Actions, Queries, Built-ins, and Events
   - Production/Situated LP, Production Rules
7. HiLog, Higher-Order Syntax, Reification, Meta-Reasoning
8. F-Logic, Frame Syntax, Object Oriented Style
9. Combining / Relating LP and FOL, e.g., Rule-based Ontologies
   - Rules for Description Logic: Description LP, DL-Safe
   - Weakened FOL in LP, Hypermonotonic Mapping
10. Recap
    - Outstanding Research Issues

Note:

Most of 2nd half of PART B.

must be omitted

due to time limitations.

We'll just discuss the outline for that.

Outline of Part C. Conclusions & Directions

1. More about Tools
2. Conclusions
3. Directions for Future research

4. References and Resources

PART A. SLIDES FOLLOW

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

Learning Goals for Tutorial

1. Overview of current state of logical KR theory, applications, languages, standards, tools/systems, market

2. Relationship to Web and Semantic Tech, overall

3. Introduction to the research issues

“Semantic” Technology

- “Semantic” in “semantic web” and “semantic rules” means:
  - 1. Knowledge-based

    … and …
  - 2. Having meaning independent of algorithm and implementation
  - Equipped with an interoperable conceptual abstraction

    … based on declarative knowledge representation (KR)

    = Shared principles of what inferences are sanctioned from a given set of premises

What are Rules on the Web

- Convergence of three streams is well along the way
  1. Using Web for interchange of rules, even pre-Web legacy kinds
     - XML syntax for rules. Transcend organizational silos.
  2. Rules working in Web context, using:
     - Web data, schemas, ontologies; Web services, queries, databases
  3. Rules using semantic knowledge representation (KR)
     - Semantics are required for effective sharing of knowledge and tools
- Web as scope for rule-based structured knowledge
  - Enrich the Web as a knowledge platform – public and intranets
  - Collaborative knowledge acquisition (KA), e.g., Wiki's
  - Web-located knowledge bases (KB's) and KR services
- ⇒ Semantic rules on the Web
  - Standardization is a key activity currently

Emerging Next Generation Web

hazy still: ***Semantic Web Services***

Semantic Web techniques

Automated Knowledge Bases

Rules (RuleML, RIF)

Ontologies (OWL, RDFS)

Databases (SQL, SPARQL)

Web Services techniques

API's on Web

(WSDL, SOAP)

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

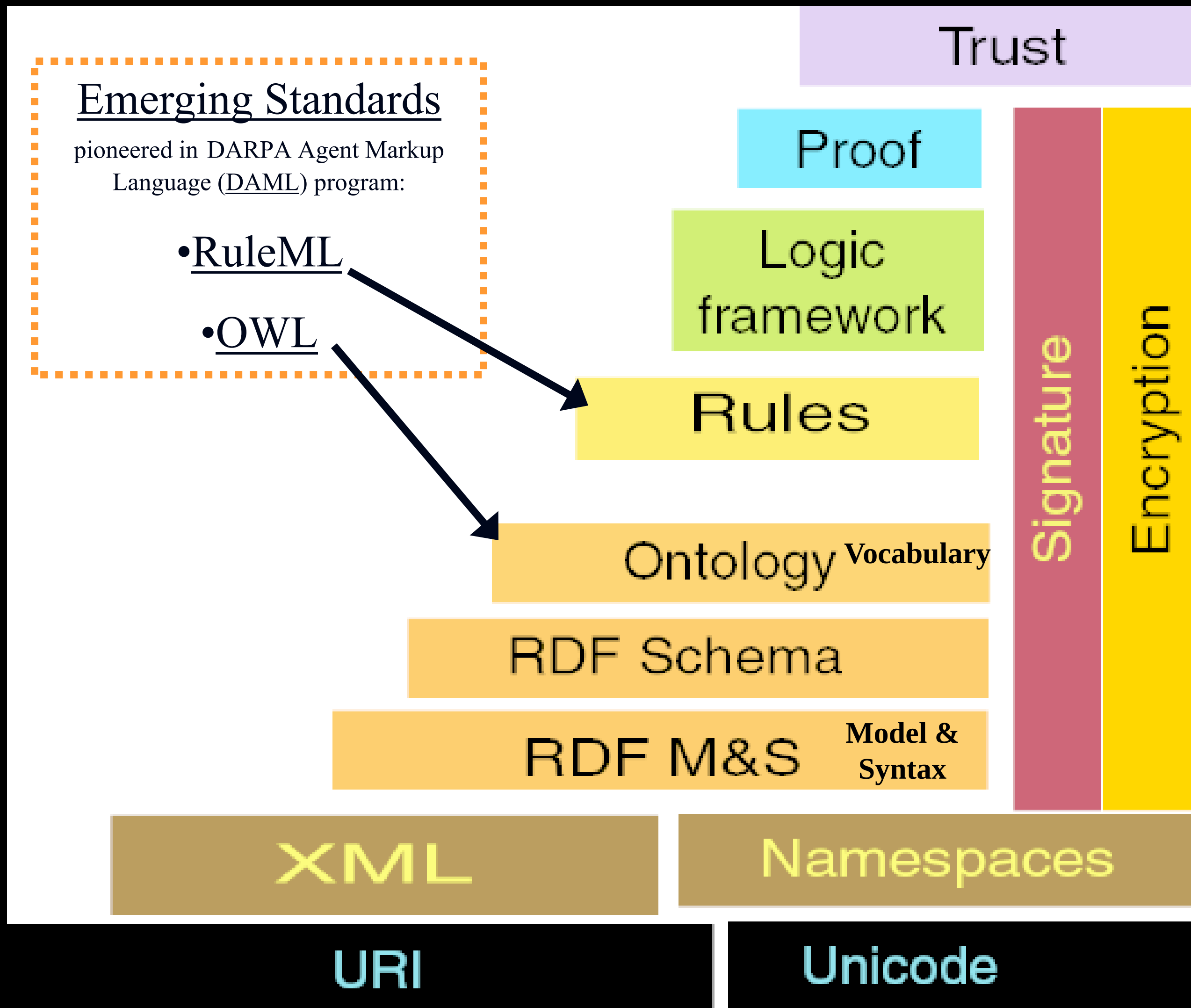
XML

First Generation Web

Semantic Web: concept, approach, pieces

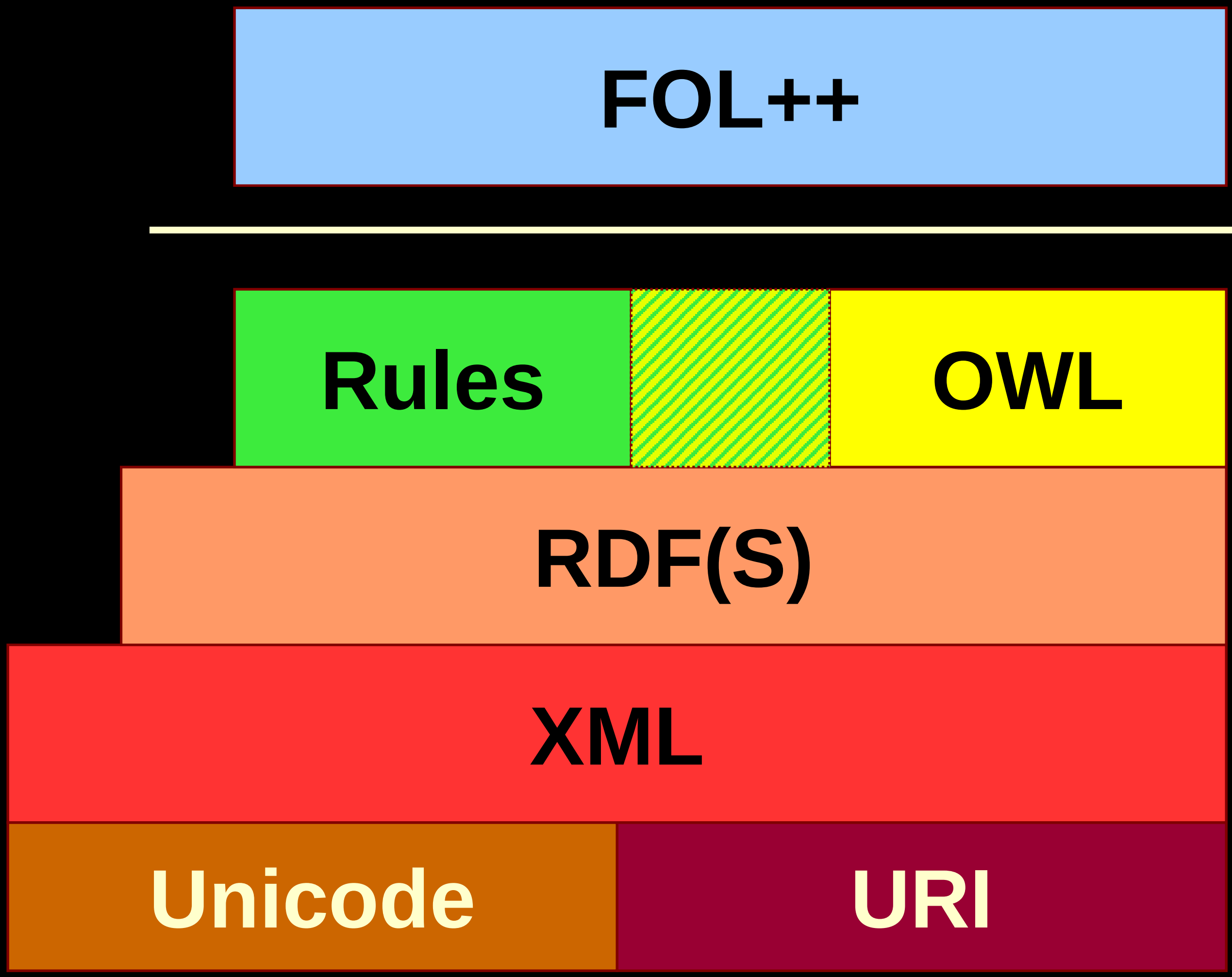
- Shared semantics when interchange data ∴ knowledge
- Knowledge Representation (cf. AI, DB) as approach to semantics
  - Standardize KR syntax, with KR theory/techniques as backing
- Web-exposed Databases: relational and XML/RDF data/queries
  - Challenge: share database schemas via meta-data
  - RDF = "Resource Description Framework" W3C* standard
- Ontology = formally defined vocabulary
  - OWL: "Web Ontology Language" W3C standard
    - Taxonomic class/property hierarchy, domains/ranges
  - Ex.: Lions are a subcategory within felines
  - Ex.: Every health care visit has a required copayment amount
- Rules = if-then logical implications, facts ~subsumes relational DB's
  - RIF: "Rule Interchange Format" W3C standard (late draft)
    - Based on Logic Programs (LP) Knowledge Representation
    - Based on RuleML (Rule Markup & Modeling Language) standards design
  - Ex.: Any student who's abused printing privileges is prohibited from using color printers
  - Ex.: AAA members get a weekend discount of 20% on suites, at hotel chain X
  - Ex.: During the mitosis phase of an animal cell's lifecycle, all DNA is replicated

W3C Semantic Web "Stack": Standardization Steps [2002]

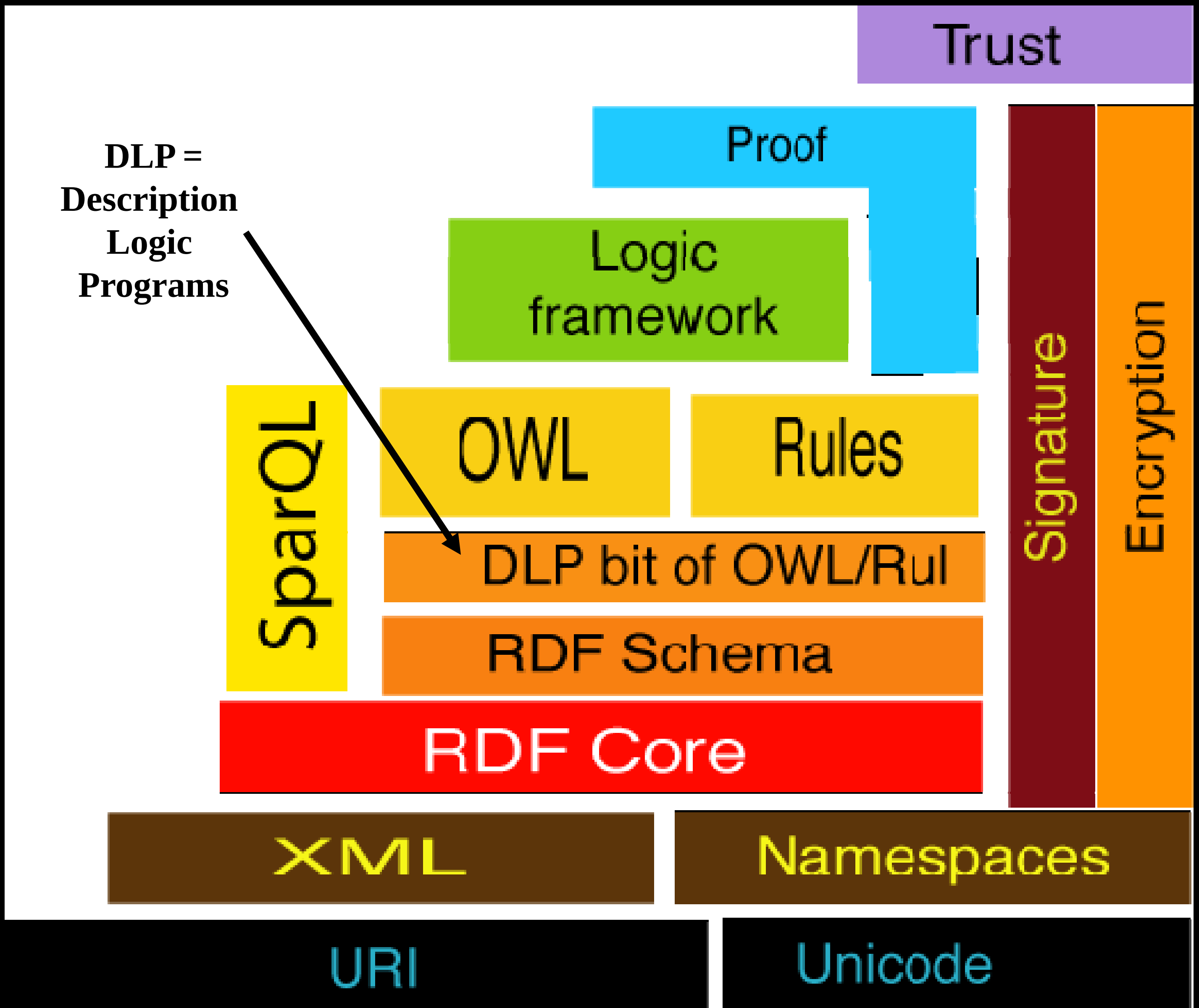


[Diagram http://www.w3.org/DesignIssues/diagrams/sw-stack-2002.png is courtesy Tim Berners-Lee]

The Web Rule Language in its Context [by RuleML & SWSI & WSMO 04-2005]



08-2005 W3C Semantic Web "Stack": Standardization Steps



Flavors of Rules Commercially Most Important today in E-Business

- E.g., in OO app's, DB's, workflows.

- Relational databases, SQL: Views, queries, facts are all rules.
  - SQL99 even has recursive rules.
- Production rules (OPS5 heritage): e.g.,
  - Jess, ILOG, Blaze, Haley: rule-based Java/C++ objects.
- Event-Condition-Action rules (loose family), cf.:
  - business process automation / workflow tools.
  - active databases; publish-subscribe.
- Prolog. *"logic programs" as a full programming language.*
- *(Lesser: other knowledge-based systems.)*

Commercial Applications of Rules today in E-Business

- There are many. An established area since the 1980's.
  - Expert systems, policy management, workflow, systems management, financial & insurance, e-commerce, trust, personal messaging, defense intelligence, …
  - Far more applications to date than of Description Logic.

- Advantages in systems specification, maintenance, integration.

- Market momentum: moderately fast growing
  - Fast in early-mid 1980's.
  - Slow late 1980's-mid-1990's.
  - Picked up again in late 1990's. (Embeddable methodologies.)
  - Accelerating in 2000's.

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; → matchmaking.
  - represent sales help, customer help, procurement, authorization/trust, brokering, workflow.
  - 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.

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

Why Standardizing Rules Now

- Rules as a form of KR (knowledge representation) are especially useful:
  - relatively mature from basic research viewpoint
  - good for prescriptive specifications (vs. descriptive)
    - a restricted programming mechanism
  - integrate well into commercially mainstream software engineering, e.g., OO and DB
    - easily embeddable; familiar
    - vendors interested already: Webizing, app. dev. tools
- ⇒⇒ Identified as part of mission of the W3C Semantic Web Activity, in approx. 2001

Overview of Key Languages & Standards

1. Database Queries are Rules
   - SQL, SPARQL, XQuery
2. Rule Markup/Modeling Language (RuleML) and related standards designs
   - Web Services modeling: SWSL, WSML
   - FOL: SWRL, Common Logic
3. W3C Rule Interchange Format (RIF)
   - Basic (no defaults or actions yet) + Framework for extension
4. SILK: Hyper Logic Programs – advanced expressiveness
5. Rules in, and for, W3C OWL (and RDFS) ontologies
6. OMG Production Rule Representation (PRR)
7. OMG Semantics of Business Vocabulary and Business Rules (SBVR)
8. JSR94 Rule Management API's

Overview of Key Tools

1. Rule systems designed to work with RDF/OWL
   - Commercial-world: Jena; Oracle; others
   - Research-world: SweetRules; cwm; others
2. Prolog and Production Rule systems
   - XSB; Jess; others
3. Advanced Expressiveness
   - Flora-2 and SILK; IBM CommonRules
4. Rules in Semantic Wikis
   - Semantic MediaWiki
5. Some Available Large Rule Bases
   - OpenCyc, Process Handbook, OpenMind

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

Example: E-Commerce Pricing Offer from SupplierCo to Buyer

…

@usualPrice price(per_unit, ?PO, \$60) $\leftarrow$

purchaseOrder(?PO, supplierCo, ?AnyBuyer) $\wedge$

quantity_ordered(?PO, ?Q) $\wedge$ (?Q $\geq$ 5) $\wedge$ (?Q $\leq$ 1000) $\wedge$

shipping_date(?PO, ?D) $\wedge$ (?D $\geq$ 24Apr00) $\wedge$ (?D $\leq$ 12May00).

@volumeDiscount price(per_unit, ?PO, \$51) $\leftarrow$

purchaseOrder(?PO, supplierCo, ?AnyBuyer) $\wedge$

quantity_ordered(?PO, ?Q) $\wedge$ (?Q $\geq$ 100) $\wedge$ (?Q $\leq$ 1000) $\wedge$

shipping_date(?PO, ?D) $\wedge$ (?D $\geq$ 28Apr00) $\wedge$ (?D $\leq$ 12May00) .

overrides(volumeDiscount , usualPrice) .

$\perp \leftarrow$ price(per_unit, ?PO, ?X) $\wedge$ price(per_unit, ?PO, ?Y) | (?X $\neq$?Y).

...

- Above rules are represented in Logic Programs KR, using esp. the Courteous defaults feature
  - Notation: @ prefixes a rule label.

Pricing Example --

XML Encoding of Rules in RuleML

```
<rulebase>
  <imp>
    <_rlab>usualPrice</_rlab>
    <_head>
      <cslit>
        <_opr><rel>price</rel></_opr>
         <ind>per_unit</ind>
         <var>PO</var>
         <ind>$60</ind>
     </cslit>
   </_head>
   <_body>     …  (see next page, if included)    </_body>
  </imp>
…
</rulebase>
```

NB: This uses an older version of RuleML markup syntax

EECOMS Example of Conflicting Rules: Ordering Lead Time

- Vendor's rules that prescribe how buyer must place or modify an order:
- A) 14 days ahead if the buyer is a qualified customer.
- B) 30 days ahead if the ordered item is a minor part.
- C) 2 days ahead if the ordered item's item-type is backlogged at the vendor, the order is a modification to reduce the quantity of the item, and the buyer is a qualified customer.
- D) 45 days ahead if the buyer is a walk-in customer.

- Suppose more than one of the above applies to the current order? **Conflict!**
- Helpful Approach: **precedence** between the rules.
  - E.g., D is a catch-case: A > D , B > D , C > D
- Often only *partial* order of precedence is justified.
  - E.g., C > A , but no precedence wrt B vs. A, nor wrt C vs. B.

Ordering Lead Time Example in LP with Courteous Defaults

- @prefCust orderModifNotice(?Order,14days)
  $\leftarrow$ preferredCustomerOf(?Buyer,SupplierCo) $\wedge$
  purchaseOrder(?Order,?Buyer,SellerCo) .
- @smallStuff orderModifNotice(?Order,30days)
  $\leftarrow$ minorPart(?Buyer,?Seller,?Order) $\wedge$
  purchaseOrder(?Order,?Buyer,SupplierCo) .
- @reduceTight orderModifNotice(?Order,2days)
  $\leftarrow$ preferredCustomerOf(?Buyer,SupplierCo) $\wedge$
  orderModifType(?Order,reduce) $\wedge$
  orderItemIsInBacklog(?Order) $\wedge$
  purchaseOrder(?Order,?Buyer,SupplierCo) .
- overrides(reduceTight , prefCust) .
- $\perp \leftarrow$ orderModifNotice(?Order,?X) $\wedge$ orderModifNotice(?Order,?Y) | X $\neq$?Y.

- NB: Rule D, and prioritization about it, were omitted above for sake of brevity.

EECOMS Supply Chain: Early Commercial Implementation & Piloting

- EECOMS agile supply chain collaboration industry consortium including Boeing, Baan, TRW, Vitria, IBM, universities, small companies
  - $29Million 1998-2000; 50% funded by NIST ATP
  - <u>application piloted</u> IBM CommonRules and early approaches which led to SweetDeal, RuleML, and SweetRules
    - contracting & negotiation; authorization & trust

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

Challenge: Capturing Semantics around Policies

- Deep challenge is to capture the semantics of data and processes, so that can:
  - Represent, monitor, and enforce policies – e.g., trust and contracts
  - Map between definitions of policy entities, e.g., in financial reporting
  - Integrate policy-relevant information powerfully

Policies for Compliance and Trust Mgmt.: Role for Semantic Web Rules

- Trust Policies usually well represented as rules
  - Enforcement of policies via rule inferencing engine
  - E.g., Role-based Access Control
    - This is the most frequent kind of trust policy in practical deployment today.
  - W3C P3P privacy standard, Oasis XACML XML access control emerging standard, …
- Ditto for Many Business Policies beyond trust arena, too
  - "Gray" areas about whether a policy is about trust vs. not: compliance, regulation, risk management, contracts, governance, pricing, CRM, SCM, etc.
  - Often, authorization/trust policy is really a part of overall contract or business policy, at application-level. Unlike authentication.
  - Valuable to reuse policy infrastructure

Trust Policies and Compliance in US Financial Industry Today

- Ubiquitous high-stakes Regulatory Compliance requirements
  - Sarbanes Oxley, SEC (also in medical domain: HIPAA), etc.
- Internal company policies about access, confidentiality, transactions
  - For security, risk management, business processes, governance
- Complexities guiding who can do what on certain business data
- Often implemented using rule techniques

- Often misunderstood or poorly implemented leading to vulnerabilities
- Typically embedded redundantly in legacy silo applications, requiring high maintenance
- Policy/Rule engines lack interoperability

Example Financial Authorization Rules

Classification	Application	Rule
Merchant	**Purchase Approval**	If credit card has fraud reported on it, or is over limit, do not approve.
Mutual Funds	**Rep trading**	*Blue Sky*: State restrictions for rep's customers.
Mortgage Company	**Credit Application**	TRW upon receiving credit application must have a way of securely identifying the request.
Brokerage	**Margin trading**	Must compute current balances and margin rules before allowing trade.
Insurance	**File Claims**	Policy States and Policy type must match for claims to be processed.
Bank	**Online Banking**	User can look at own account.
All	**House holding**	For purposes of silo (e.g., statements or discounts), aggregate accounts of all family members.

Slide also by Chitravanu Neogy

Advantages of Standardized SW Rules

- Easier Integration: with rest of business policies and applications, business partners, mergers & acquisitions
- Familiarity, training
- Easier to understand and modify by humans
- 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

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

- Reduced system dev./maint./training costs
- Better/faster/cheaper policy admin.
- Interoperability, flexibility and re-use benefits
- Greater visibility into enterprise policy implementation => better compliance
- Centralized ownership and improved governance by Senior Management
- Rich, expressive trust management language allows better conflict handling in policy-driven decisions

Some Answers to: "Why does SW/SWS 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

SW Early Adoption Candidates: High-Level View

- "Death. Taxes. Integration."
- Application/Info Integration:
  - Intra-enterprise
    - EAI, M&A; XML infrastructure trend
  - Inter-enterprise
    - E-Commerce: procurement, SCM
  - Combo
    - Business partners, extranet trend

SW Adoption Roadmap: Strategy Considerations

- Likely first uses in a lot of B2B interoperability or heterogeneous-info-integration intensive applications (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
  - increase speed, creates value, increases dynamism
  - macro effects create
    - stability sometimes (e.g., supply chain reactions due to lag; other negative feedbacks)
    - volatility sometimes (e.g., perhaps financial market swings)
  - increase flexibility, decrease lock-in
- Agility in business processes, supply chains

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

Ontology Translation Via Rules

- Use rules to represent mappings from data source to domain ontologies
  - Rules can be automatically or manually generated
  - Can support unit of measure conversion and structural transformation
- Example using SWRL
  - http://www.daml.org/2004/05/swrl-translation/Overview.html

Equational Ontological Conflicts in Financial Reporting

of customers = # of end_customers + # of distributors

Gross Profit = Net Sales – Cost of Goods

P/E Ratio = Price / Earnings(last 4 Qtr)

Price = Nominal Price + Shipping

# of customers = # of end_customers + # of prospective customers

Gross Profit = Net Sales – Cost of Goods – Depreciation

P/E Ratio = Price/ [Earnings(last 3 Qtr) + Earnings(next quarter)]

Price = Nominal Price + Shipping + Tax

" heterogeneity in the way data items are *calculated* from other data items *in terms of definitional equations*"

Slide also by Aykut Firat and Stuart Madnick

EOC in Primark Databases

Key Concepts

Top 25 US Co. by Net Sales (Disclosure)

Rank	Company	Net Sales (000's)		Date
1	General Motors Corp	168,828,600	?	12/31/95
2	Ford Motor Co	137,137,000		12/31/95
3	Exxon Corp	121,804,000	?	12/31/95
4	Wal Mart Stores Inc	93,627,000		01/31/96
5	AT&T	79,609,000		12/31/95
6	Mobil Corp	73,413,000	?	12/31/95
7	International Business M	71,904,000		12/31/95
8	General Electric Co	70,028		
...	...	...		

Top 25 International Co. by Net Sales (Worldscope)

Rank	Company	Net Sales (000's)	Date
1	Mitsubishi Corporation	165,848,468	03/31/96
2	General Motors Corp	163,861,100	12/31/95
...	...	...	...
8	Exxon Corp	107,893,000	12/31/95
...	...	...	...
16	International Business M	71,940,000	12/31/95
17	General Electric Co	69,948,000	12/31/95
20	Mobil Corp	64,767,000	12/31/95
...	...	...	...

Primark was a company that owned:

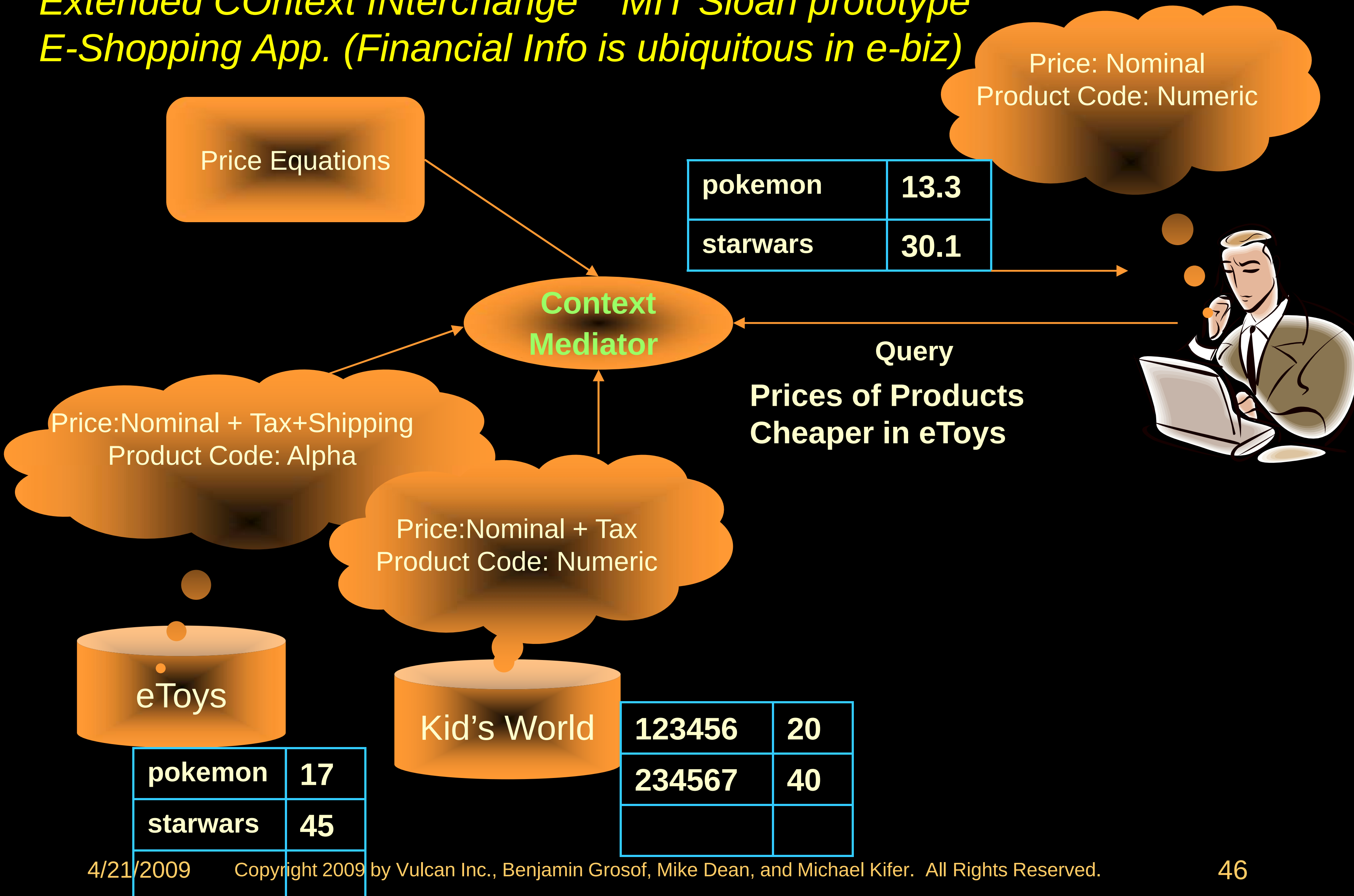
- Disclosure
- Worldscope
- DataStream

Information services

Solution Approach: ECOIN

Extended COntext INterchange   MIT Sloan prototype

E-Shopping App. (Financial Info is ubiquitous in e-biz)



pokemon	13.3
starwars	30.1

pokemon	17
starwars	45

123456	20
234567	40

Approach: ECOIN

Solution Methodology

•Context-based loosely-coupled integration

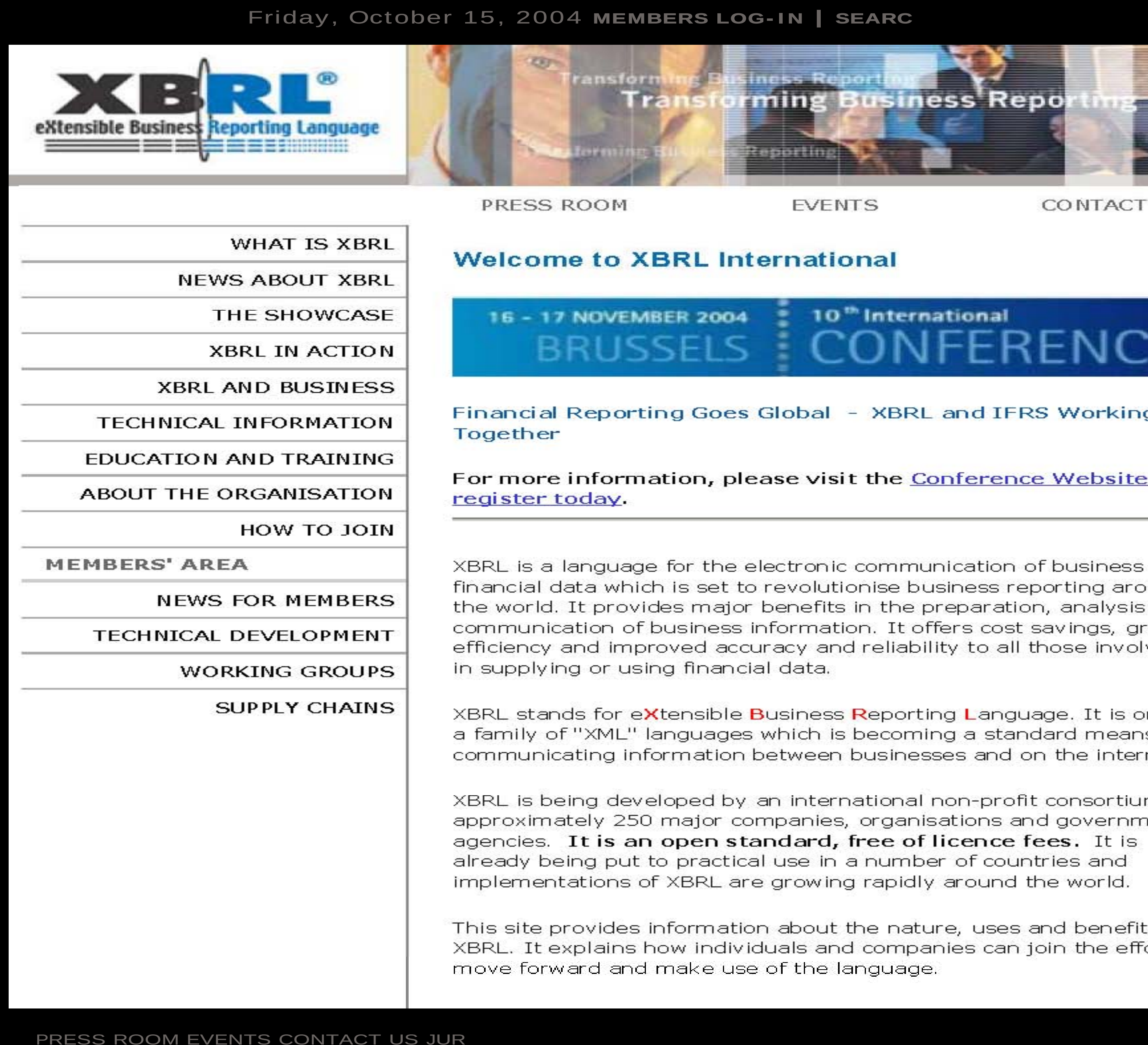
- •Extends the Context Interchange (COIN) framework developed at MIT

•Symbolic Equation Solving using Constraint Logic Programming

- •Integrates symbolic equation solving techniques with abductive logic programming

• *In-progress:* Utilizing RuleML and OWL in ECOIN

1. OWL formulation of COIN ontologies: see [Bhansali, Madnick, & Grosof ISWC-2004 poster]


Friday, October 15, 2004 MEMBERS LOG-IN | SEARC
XBRL®
eXtensible Business Reporting Language
Transforming Business Reporting
PRESS ROOM
EVENTS
CONTACT
WHAT IS XBRL
NEWS ABOUT XBRL
THE SHOWCASE
XBRL IN ACTION
XBRL AND BUSINESS
TECHNICAL INFORMATION
EDUCATION AND TRAINING
ABOUT THE ORGANISATION
HOW TO JOIN
MEMBERS' AREA
NEWS FOR MEMBERS
TECHNICAL DEVELOPMENT
WORKING GROUPS
SUPPLY CHAINS
Welcome to XBRL International
16 - 17 NOVEMBER 2004
BRUSSELS
10th International
CONFERENC
Financial Reporting Goes Global - XBRL and IFRS Working Together
For more information, please visit the Conference Website register today.
XBRL is a language for the electronic communication of business financial data which is set to revolutionise business reporting aro the world. It provides major benefits in the preparation, analysis communication of business information. It offers cost savings, gr efficiency and improved accuracy and reliability to all those involv in supplying or using financial data.
XBRL stands for eXtensible Business Reporting Language. It is o a family of "XML" languages which is becoming a standard means communicating information between businesses and on the intern
XBRL is being developed by an international non-profit consortiu approximately 250 major companies, organisations and governm agencies. It is an open standard, free of licence fees. It is already being put to practical use in a number of countries and implementations of XBRL are growing rapidly around the world.
This site provides information about the nature, uses and benefit XBRL. It explains how individuals and companies can join the eff move forward and make use of the language.
PRESS ROOM EVENTS CONTACT US JUR

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

Contracts in E-Commerce Lifecycle

- Discovery, advertising, matchmaking
  - Search, sourcing, qualification/credit checking
- Negotiation, bargaining, auctions, selection, forming agreements, committing
  - Hypothetical reasoning, what-if'ing, valuation
- Performance/execution of agreement
  - Delivery, payment, shipping, receiving, notification
- Problem Resolution, Monitoring
  - Exception handling

Approach: Rule-based Contracts for E-commerce

- Rules as way to specify (part of) business processes, policies, products: as (part of) contract terms.
- Complete or partial contract.
  - As default rules. Update, e.g., in negotiation.
- Rules provide high level of conceptual abstraction.
  - easier for non-programmers to understand, specify, dynamically modify & merge. E.g.,
  - by multiple authors, cross-enterprise, cross-application.
- Executable. Integrate with other rule-based business processes.

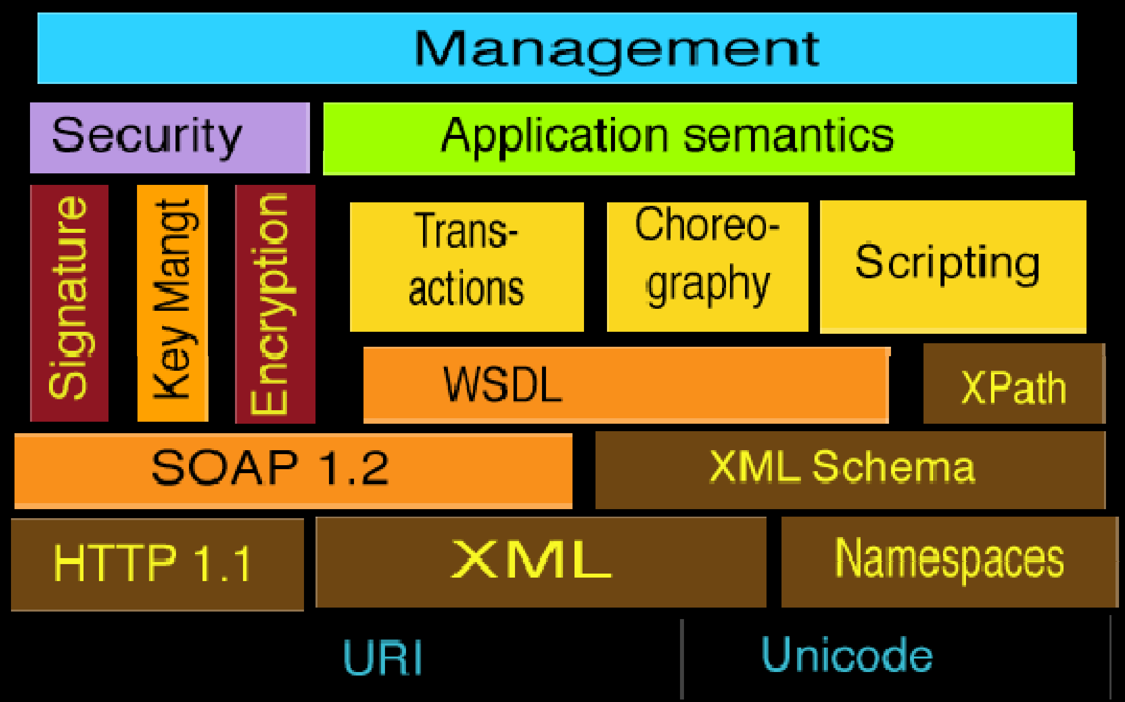
Semantic Web Services

- 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

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

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

"Wire" Protocols | **Service Description**

W3C WS Choreography Group
BPEL4WS (Microsoft, IBM, BEA)
WSCL (HP)BPML (Most but Microsoft)
WSCI (Sun, BEA, Yahoo, …)
XLANG (Microsoft), WSFL (IBM), …

SOAP Blocks
SOAP/XMLP
XML
HTTP/SMTP
TCP/IP

SWS Language
Process
WSDL Extensions
WSDL
XML

Registry (UDDI)
Inspection

SWS Initiative (SWSI)
-- automate Tasks of:
Discovery
Invocation
Interoperation
Deal Negotiation
Composition
Monitoring
Verification

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

Semantic Web Services Framework (SWSF)

- By Semantic Web Services Initiative (SWSI) http://www.swsi.org
  - Coordinates global research and early-phase standardization in SWS
  - http://www.swsi.org
  - Researchers from universities, companies, government
  - Industrial partners; DAML and WSMO backing
  - Collaborators: OWL-S, WSMO, RuleML, DAML

- Designed SWSF: http://www.daml.org/services/swsf/1.0/
  - Rules & FOL language (SWSL/RuleML)
  - Ontology for SWS (SWSO)
    - Drawn largely from OWL-S and PSL
  - Application Scenarios
  - *Also: 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)

Rule-based Semantic Web Services

- Rules/LP in appropriate combination with DL as KR, for RSWS
  - DL good for categorizing: a service overall, its inputs, its outputs

- Rules to describe service process models
  - rules good for representing:
    - preconditions and postconditions, their contingent relationships
    - contingent behavior/features of the service more generally,
      - e.g., exceptions/problems
  - familiarity and naturalness of rules to software/knowledge engineers

- Rules to specify deals about services: cf. e-contracting.

Outline of Part A. Intro & Uses

1. Overview of tutorial, and get acquainted
2. What are: Rules on the Web, Semantic Rules/Web/Tech
3. Uses and Kinds of rules
   - Commercial, web. Current, envisioned.
   - Requirements. Business value, IT lifecycle.
   - Strategic roadmapping of future adoption
4. Example Use Cases
   - E-commerce pricing, ordering
   - Trust, compliance, policies, financial services
   - Info integration, ontology mapping, business reporting.
   - E-contracts, e-commerce life cycle. Semantic Web Services.

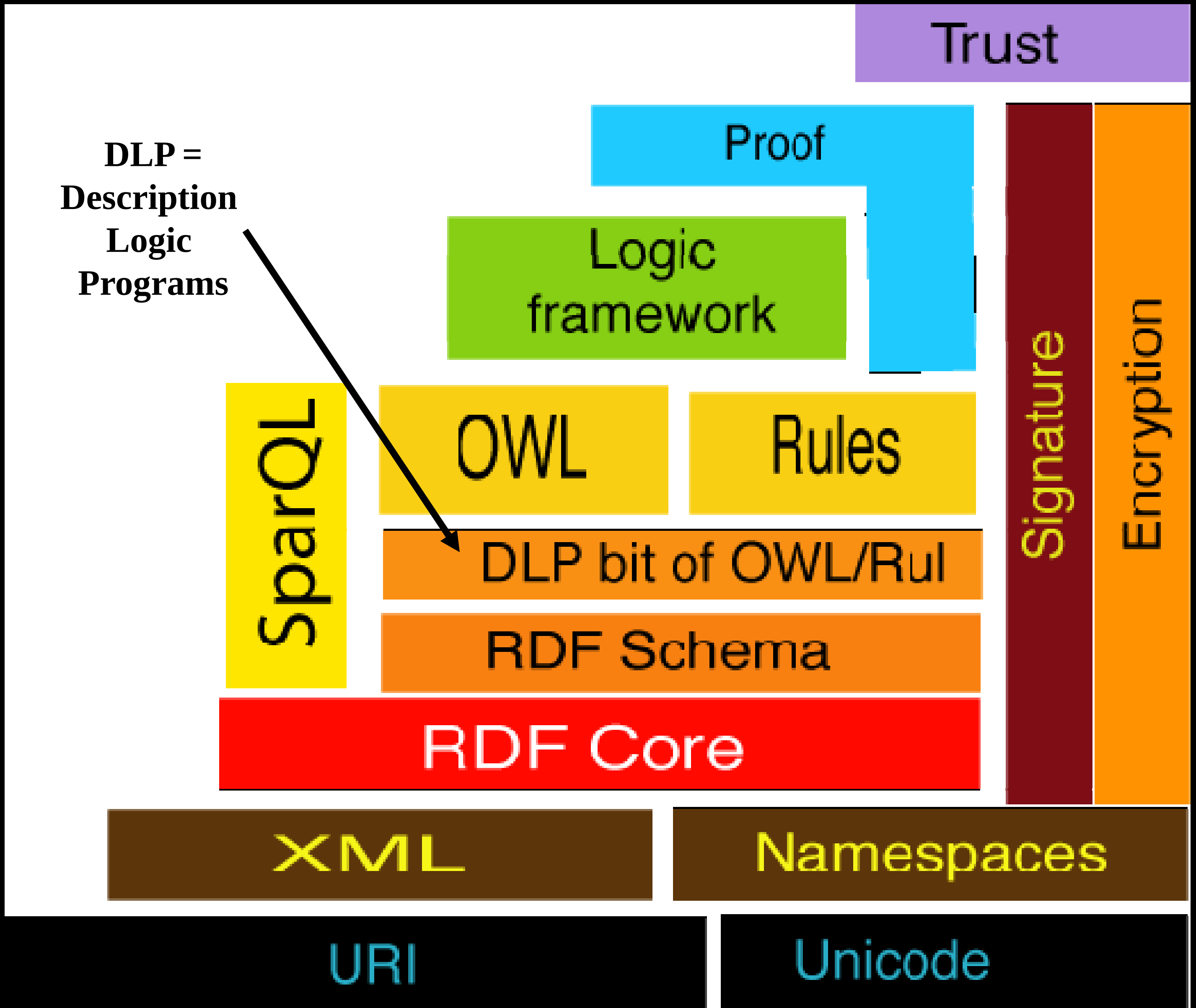
NB: (2.)-(4.) are interleaved. Some of (4.) is skimmed.

PART B. SLIDES FOLLOW

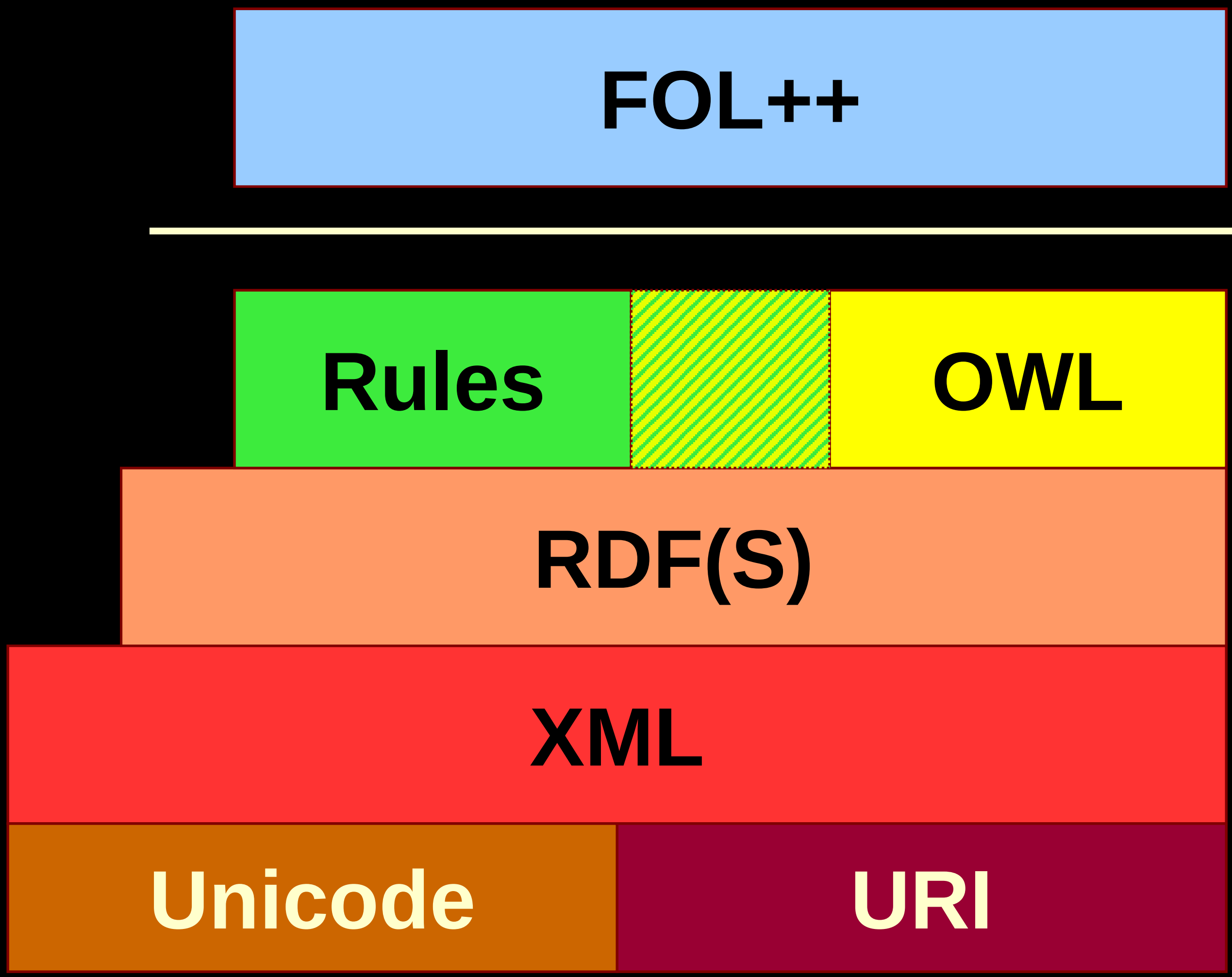
Outline of Part B. Concepts & Foundations

1. Overview of Logical Knowledge Representations
   - First Order Logic (FOL), Logic Programs (LP)
2. Horn Case
3. Nonmonotonicity, Defaults, Negation, Priorities
   - Semantics for Default Negation
   - Courteous LP, Argumentation Theories
4. More Connectives and Quantifiers: Lloyd-Topor, Skolemization
5. Additional Features: Datatypes, Integrity Constraints, Equality, Aggregation
6. Procedural Attachments to Actions, Queries, Built-ins, and Events
   - Production/Situated LP, Production Rules
7. HiLog, Higher-Order Syntax, Reification, Meta-Reasoning
8. F-Logic, Frame Syntax, Object Oriented Style
9. Combining / Relating LP and FOL, e.g., Rule-based Ontologies
   - Rules for Description Logic: Description LP, DL-Safe
   - Weakened FOL in LP, Hypermonotonic Mapping
10. Recap
    - Outstanding Research Issues

08-2005 W3C Semantic Web "Stack": Standardization Steps



The Web Rule Language in its Context
[by RuleML & SWSI & WSMO   04-2005]



Concept of KR

- A KR 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)
    - *Remark: In declarative logic programs KR, LC is a subset of LP*
  - |= 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 Conc is a functional relation.

- Typically, e.g., in FOL and LP, entailment is defined formally in terms of <u>models</u>, i.e., truth assignments that satisfy the premises and meet other criteria.

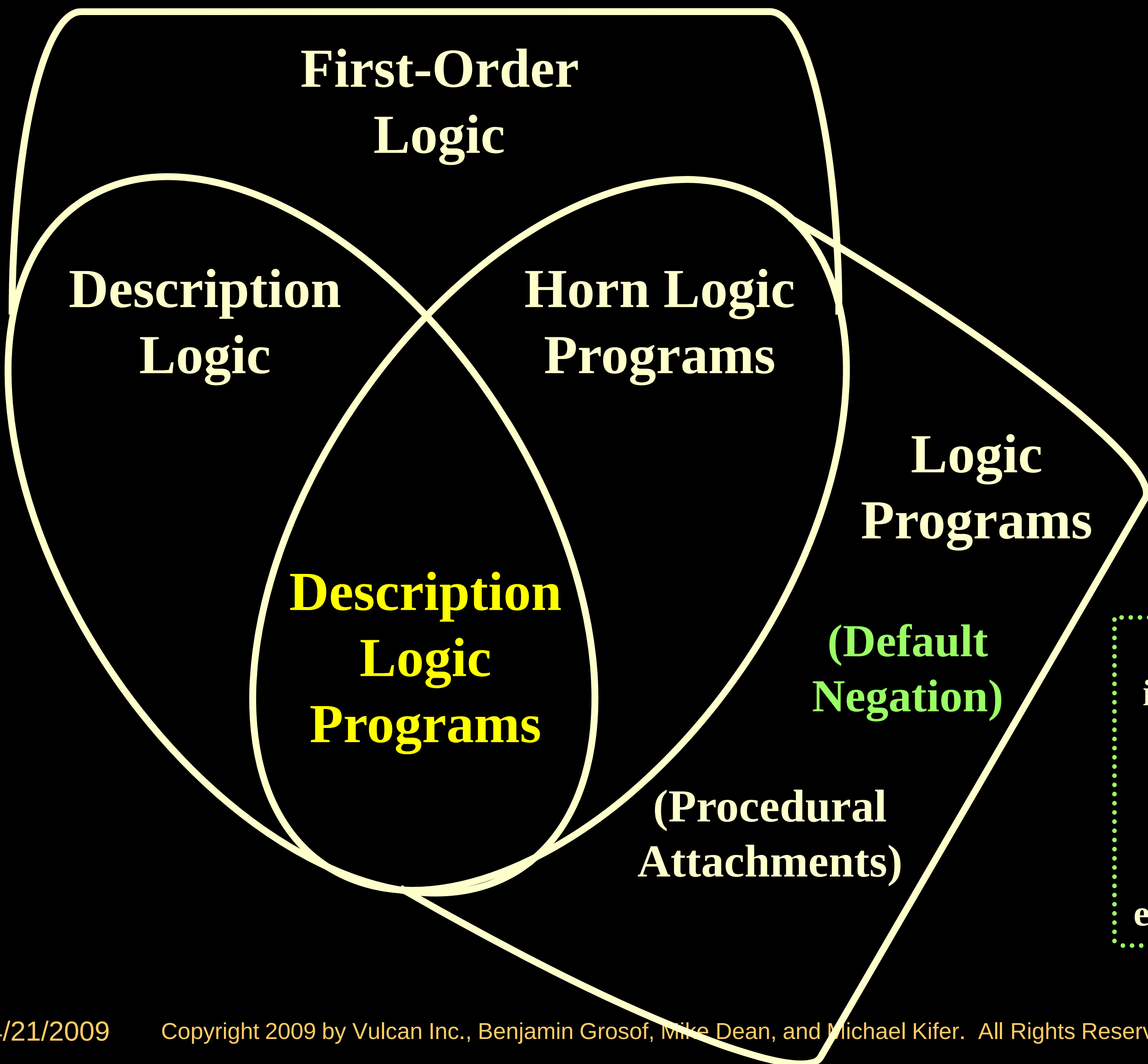
Knowledge Representation: What's the Game?

- Expressiveness: useful, natural, complex enough
- Reasoning algorithms
- Syntax: encoding data format -- here, in XML
- Semantics: principles of sanctioned inference, independent of reasoning algorithms
- 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

Overview of Logic Knowledge Representations (KR's) and Markup Standards

- First Order Logic (FOL). Also called "classical logic", as is HOL (below).
  - Standards efforts:
    - ISO Common Logic (CL); FOL RuleML
  - Restriction: Horn FOL
  - Restriction: Description Logic (DL) – overlaps with Horn
    - Standards: W3C OWL-DL; W3C RDF-Schema (expressive subset)
  - Extension: Higher Order Logic (HOL)
    - Hilog = higher order syntactically, but reducible to first order
- Logic Programs (LP)
  - (Here: in the *declarative* sense.)
  - Standard (late draft): W3C Rule Interchange Format (RIF)
  - Standard designs for additional expressiveness: RuleML / SWSL / SILK
  - Extension features: Hilog; also:
    - Nonmonotonicity: Negation, Defaults (cf. Courteous)
    - Procedural Attachments (aproc's) for external queries, events, actions
  - Restriction: Horn LP
  - Restriction: Description Logic Programs (DLP) – overlaps with DL

Venn Diagram: Expressive Overlaps among KR's



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

Description Logic: KR Expressiveness, in brief

- Restriction of First Order Logic (FOL)
  - Strongest restriction is on the patterns of variable appearances
    - Cannot represent most kinds of chaining among rules
  - No logical functions
- Allows:
  - Class predicates of arity 1
  - Property predicates of arity 2
  - Membership axioms: foo instance-of BarClass
  - Inclusion axioms between classes (possibly complex)
    - C1 subsumed-by C2
    - I.e., x instance-of C1 $\Rightarrow$ x instance-of C2
  - Complex class expressions, e.g.
    - Electrical device that has two speakers and a 120V or 220V power supply
  - Indirectly can represent n-ary predicates
- Good for representing:
  - Many kinds of ontological schemas, including taxonomies
  - Taxonomic/category subsumptions (with strict inheritance)
  - Some kinds of categorization/classification and configuration tasks

Summary of Computational Complexity of KR's

- *For task of inferencing, i.e., answering a given query.*
  - *Tractable = time is polynomial in n, worst-case; n = |premises|*
- First Order Logic (FOL)
  - Intractable for Propositional (co-NP-complete)
  - Undecidable in general case
  - Decidable but intractable for Description Logic
- Logic Programs (LP) with extensions for negation, defaults, Hilog, aproc's
  - Tractable for broad cases
    - $O(n^2)$ for Propositional with negation and defaults
    - Complexity qualitatively similar to Relational DB's
    - Truly Web-scaleable, therefore
  - Undecidable in general case (cause: infinite recursion thru functions)

More on Computational Complexity of LP

- $O(n)$ for propositional Horn. (Ditto in FOL.)
- $O(n \cdot m)$ for propositional with negation (well-founded), where m = # atoms ($m \leq n$)
  - Defaults add no increase in the complexity bound (reducible linearly to NAF)
- Typically-met restrictions:
  - Constant-bounded number of distinct variables per rule (VB restriction)
    - In DL form of DLP, VB ≡ constant-bounded number of distinct DL quantifiers (incl. min/max cardinality) in class descriptions per inclusion axiom
  - Time per aproc call is tractable (AT restriction)
- Most feature extensions can be added to LP without affecting tractability
- Key restriction to ensure tractability (or decidability) is to:
  - Avoid blow-up from <u>recursion through logical functions</u> (of arity > 0)
    - ⇒ Keep the relevant set of ground atoms tractable (or finite)
    - Here, recursion means dependency cycles among rules
  - E.g., function-free is a simple sufficient condition
    - Then # of ground atoms = $O(n^{v+1})$, where v is the bound in VB
  - *More research on detailed theory and algorithms is needed, however*

Requirements Analysis for Logical Functions

- Function-free is a commonly adopted restriction in practical LP/Web rules today
  - DB query languages: SQL, SPARQL, XQuery
  - RIF Basic Logic Dialect
  - Production rules, and similar Event-Condition-Action rules
  - OWL

- BUT functions are often needed for Web (and other) applications. Uses include:
  - Hilog and reification – higher-order syntax
    - For meta- reasoning, e.g., in knowledge exchange or introspection
      - Ontology mappings, provenance, KB translation/import, multi-agent belief, context
      - KR macros, modals, reasoning control, KB modularization, navigation in KA
      - The Web is all about meta- data!
  - Skolemization – to represent existential quantifiers
    - E.g., RDF blank nodes
  - Convenient naming abstraction, generally
    - steering_wheel(my_car)

Outline of Part B. Concepts & Foundations

1. Overview of Logical Knowledge Representations
   - First Order Logic (FOL), Logic Programs (LP); Complexity
2. Horn Case
3. Nonmonotonicity, Defaults, Negation, Priorities
   - Semantics for Default Negation
   - Courteous LP, Argumentation Theories
4. More Connectives and Quantifiers: Lloyd-Topor, Skolemization
5. Additional Features: Datatypes, Integrity Constraints, Equality, Aggregation
6. Procedural Attachments to Actions, Queries, Built-ins, and Events
   - Production/Situated LP, Production Rules
7. HiLog, Higher-Order Syntax, Reification, Meta-Reasoning
8. F-Logic, Frame Syntax, Object Oriented Style
9. Combining / Relating LP and FOL, e.g., Rule-based Ontologies
   - Rules for Description Logic: Description LP, DL-Safe
   - Weakened FOL in LP, Hypermonotonic Mapping
10. Recap
    - Outstanding Research Issues

Horn LP as Foundation Core KR

- Horn LP provides the foundation core KR and conceptual intuitions for Rules, both in:
  - pre- Semantic Web – including commercial rule systems
  - Semantic Web – including RIF and RuleML

Horn FOL

- The Horn subset of FOL is defined relative to <u>clausal</u> form of FOL
- A Horn clause is one in which there is at most one positive literal.

  It takes one of the two forms:

  1. $H \lor \neg B1 \lor \ldots \lor \neg Bm$.   A.k.a. a <u>definite</u> clause / <u>rule</u>
     - <u>Fact</u>   H .   is special case of rule (H ground, m=0)
  2. $\neg B1 \lor \ldots \lor \neg Bm$.   A.k.a. an <u>integrity constraint</u>

  where $m \geq 0$, H and Bi's are atoms.

  (An atom = pred(term_1,…,term_k) where pred has arity k.)
- A definite clause (1.) can be written equivalently as an <u>implication</u>:
  - Rule := $H \Leftarrow B1 \land \ldots \land Bm$.   where $m \geq 0$, H and Bi's are atoms

    head if body ;
- An integrity constraint (2.) can likewise be written as:
  - $\perp \Leftarrow B1 \land \ldots \land Bm$.   A.k.a. <u>empty-head</u> rule ($\perp$ is often omitted).

For refutation theorem-proving , represent a <u>negated goal</u> as (2.).

Horn LP Syntax and Semantics

- Horn LP syntax is similar to implication form of Horn FOL
  - The implication connective's semantics are a bit weaker however. We will write it as $\leftarrow$ instead of $\Leftarrow$.
- Declarative LP with model-theoretic semantics
  - Same for forward-direction ("derivation" / "bottom-up") and backward-direction ("query" / "top-down") inferencing
  - Model M(P) = a set of (concluded) ground atoms
    - Where P = the set of premise rules
- Semantics is defined via the least fixed point of an operator T_P. T_P outputs conclusions that are immediately derivable (through some rule in P) from an input set of intermediate conclusions I_j.
  - $I_{j+1} = T_P(I_j)$; $I_0 = \varnothing$ (empty set)
    - I_{j+1} = {all head atoms of rules whose bodies are satisfied by I_j}
  - M(P) = LeastFixedPoint(T_P) ; where LFP = the I_m such that $I_{m+1} = I_m$
  - Simple algorithm: `DO` {run each rule once} `UNTIL` {quiescence}

Example of Horn LP vs. Horn FOL

- Let P be:
  - DangerousTo(?x,?y) $\leftarrow$ PredatorAnimal(?x) $\wedge$ Human(?y).
  - PredatorAnimal(?x) $\leftarrow$ Lion(?x).
  - Lion(Simba).
  - Human(Joey).
- I1 = {Lion(Simba), Human(Joey)}
- I2 = {PredatorAnimal(Simba),Lion(Simba), Human(Joey)}
- I3 = {DangerousTo(Simba,Joey), PredatorAnimal(Simba),Lion(Simba), Human(Joey)}
- I4 = I3. Thus M(P) = I3.

- Let P' be the Horn <u>FOL</u> rulebase version of P above, where $\Leftarrow$ replaces $\leftarrow$.
- Then the ground atomic conclusions of P' are exactly those in M(P) above.
- P' also entails various non-ground-atom conclusions, including:

1. Non-unit derived clauses, e.g., DangerousTo(Simba,?y) $\Leftarrow$ Human(?y).
2. All tautologies of FOL, e.g., Human(?z) $\vee$ $\neg$Human(?z).
3. Combinations of (1.) and (2.), e.g., $\neg$Human(?y) $\Leftarrow$ $\neg$DangerousTo(Simba,?y).

Horn LP Compared to Horn FOL

- Fundamental Theorem connects Horn LP to Horn FOL:
  - M(P) = {all ground atoms entailed by P in Horn FOL}

- Horn FOL has additional non-ground-atom conclusions, notably:
  - non-unit derived clauses; tautologies
- Can thus view Horn LP as the f-weakening of Horn FOL.
  - "f-" here stands for "fact-form conclusions only"
  - A restriction on form of conclusions (not of premises).
- Horn LP – differences from Horn FOL:
  - Conclusions Conc(P) = essentially a set of ground atoms.
    - Can extend to permit more complex-form queries/conclusions.
  - Consider Herbrand models only, *in typical formulation and usage.*
    - P can then be replaced equivalently by {all ground instantiations of each rule in P}
    - But can extend to permit: extra unnamed individuals, beyond Herbrand universe
  - Rule has non-empty head, *in typical formulation and usage.*
    - Can extend to detect violation of integrity constraints

The "Spirit" of LP

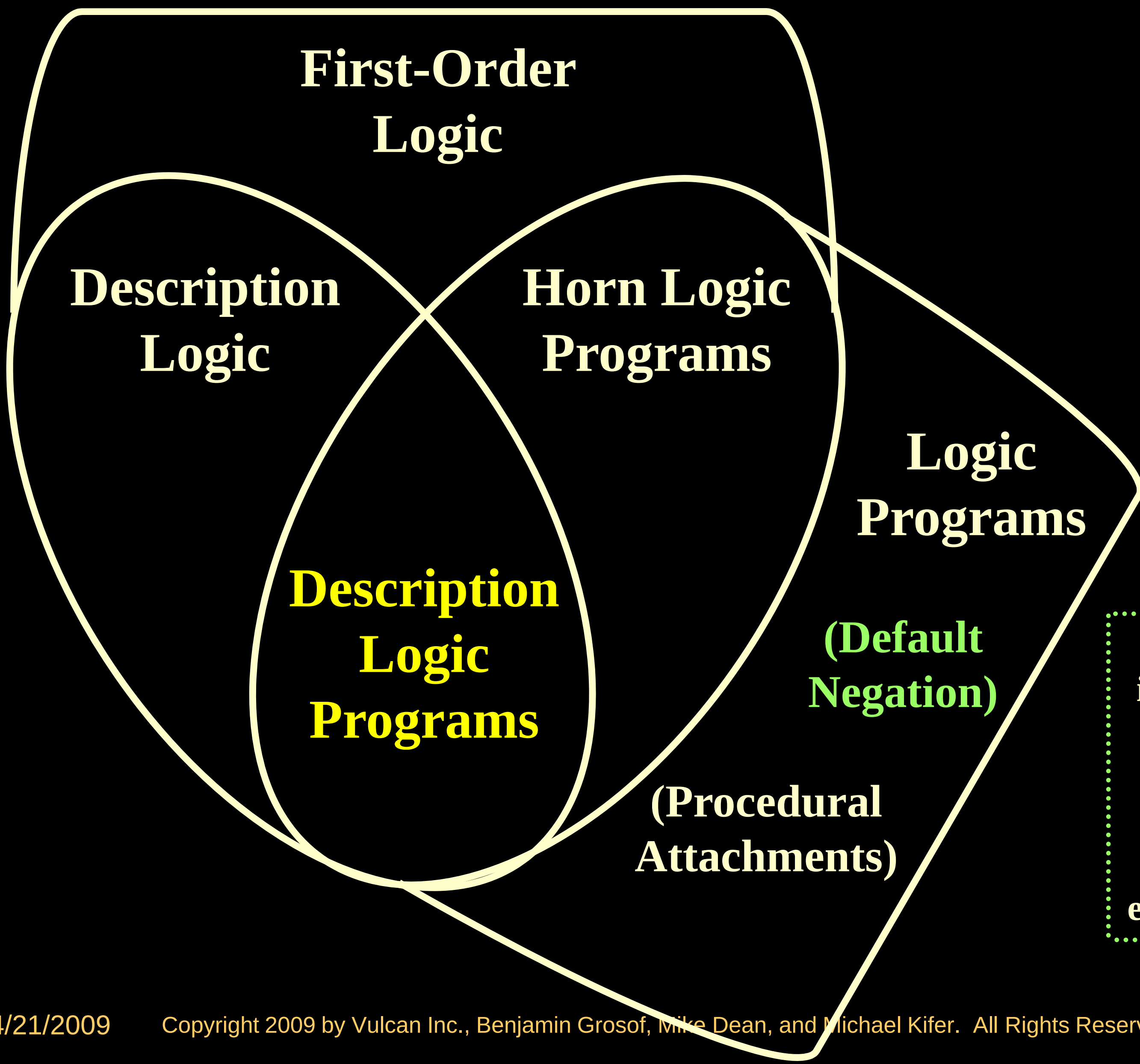
The following summarizes the "spirit" of how LP differs from FOL:

- "Avoid Disjunction"
  - Avoid disjunctions of positive literals as expressions
    - In premises, intermediate conclusions, final conclusions
  - Permitting such disjunctions creates exponential blowup
    - In propositional FOL: 3-SAT is NP-hard
    - In the leading proposed approaches that expressively add disjunction to LP with negation, e.g., propositional Answer Set Programs
  - No "reasoning by cases", therefore
- "Stay Grounded"
  - Avoid (irreducibly) non-ground conclusions

LP, unlike FOL, is straightforwardly extensible, therefore, to:

- Nonmonotonicity – defaults, incl. NAF
- Procedural attachments, esp. external actions

Venn Diagram: Expressive Overlaps among KR's



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

Outline of Part B. Concepts & Foundations

1. Overview of Logical Knowledge Representations
   - First Order Logic (FOL), Logic Programs (LP); Complexity
2. Horn Case
3. Nonmonotonicity, Defaults, Negation, Priorities
   - Semantics for Default Negation
   - Courteous LP, Argumentation Theories
4. More Connectives and Quantifiers: Lloyd-Topor, Skolemization
5. Additional Features: Datatypes, Integrity Constraints, Equality, Aggregation
6. Procedural Attachments to Actions, Queries, Built-ins, and Events
   - Production/Situated LP, Production Rules
7. HiLog, Higher-Order Syntax, Reification, Meta-Reasoning
8. F-Logic, Frame Syntax, Object Oriented Style
9. Combining / Relating LP and FOL, e.g., Rule-based Ontologies
   - Rules for Description Logic: Description LP, DL-Safe
   - Weakened FOL in LP, Hypermonotonic Mapping
10. Recap
    - Outstanding Research Issues

Concept of Logical Monotonicity

- A KR S is said to be <u>logically monotonic</u> when in it:

  $$P1 \subseteq P2 \quad \Rightarrow \quad Conc(P1,S) \subseteq Conc(P2,S)$$

- Where P1, P2 are each a set of premises in S
- I.e., whenever one adds to the set of premises, the set of conclusions non-strictly grows (one does not retract conclusions).

- *Monotonicity is good for pure mathematics.*
  - *"Proving a theorem means never having to say you're sorry."*

Nonmonotonicity – its Pragmatic Motivations

- Pragmatic reasoning is, in general, nonmonotonic
  - E.g., policies for taking actions, exception handling, legal argumentation, Bayesian/statistical/inductive, etc.
  - Monotonic is a special case – simpler in <u>some</u> regards
- Most commercially important rule systems/applications use nonmon
  - A basic expressive construct is ubiquitous there:
    - <u>Default Negation</u> a.k.a. Negation-As-Failure (NAF)
      - BUT with varying semantics – <u>often not fully declarative</u> cf. LP
        - Primarily due to historical hangovers and lack of familiarity with modern algorithms
  - Another expressive construct, almost as ubiquitous there, is:
    - <u>Priorities</u> between rules
- Such nonmonotonicity enables:
  - Modularity and locality in revision/updating/merging

Default Negation: Intro

- Default negation is the most common form of negation in commercially important rule and knowledge-based systems.
- Concept/Intuition for **~**q ; ~ stands for default negation
  - q is not derivable from the available premise info
  - fail to believe q
  - … but might also not believe q to be false
  - A.k.a. "*weak*" negation, or NAF.
- Contrast with: **¬**q ; ¬ stands for strong negation
  - q is believed to be false
  - A.k.a. *"classical"* negation

LP with Default Negation

- *Normal* LP, a.k.a. *Ordinary* LP (OLP)
  - Adds Default Negation to Horn LP
- Syntax: Rule generalized to permit negated body literals:
  - $H \leftarrow B_1 \wedge \ldots \wedge B_k \wedge \sim B_{k+1} \wedge \ldots \wedge \sim B_m$.

  where $m \geq 0$, H and Bi's are atoms

- Semantics has subtleties for the fully general case
  - Difficulty is interaction of negation with recursion
    - Recursion in LP means cyclic dependencies (thru the rules) of predicates/atoms
  - Lots of theory developed during 1984-1994
  - Well-understood theoretically since 1994

Semantics for LP with Default Negation

- For fully general case, there are two* alternative semantics
- Both agree for a broad restricted case: *stratified* OLP
- The *Well Founded* Semantics (WFS) is most popular
  - Tractable for the propositional case. Often linear, worst-case quadratic.
  - The main focus commercially. E.g., XSB, Ontobroker.
  - Employs a 3rd truth value u ("undefined"), when non-stratified
  - Definition uses iterated minimality: Horn-case then close-off; repeat til done.
- An earlier *Stable* Semantics is studied esp. by some theory researchers
  - Enables a limited kind of disjunction in conclusions, bit closer to FOL in spirit
  - ⇒ Intractable for propositional case
  - Does not employ the 3rd truth value
  - ⇒ Can be ill-defined, i.e., there may be no set of conclusions
  - When not ill-defined, it extends WFS with further disjunctive conclusions
  - Answer set programs extends it expressively to permit disjunction in premises

* There are also other older proposals of mainly historical interest

Basic Example of LP with NAF

- RB1: (NB: this example is purely fictional.)
  - price(Amazon,Sony5401,?day,?cust,49.99)
    $\leftarrow$ inUSA(?cust) $\wedge$ inMonth(?day,2004-10) $\wedge$ ~onSale(?day).
  - price(Amazon,Sony5401,?day,?cust,39.99)
    $\leftarrow$ inUSA(?cust) $\wedge$ inMonth(?day,2004-10) $\wedge$ onSale(?day).
  - inMonth(2004-10-12, 2004_10).
  - inMonth(2004-10-30, 2004-10).
  - inUSA(BarbaraJones).
  - inUSA(SalimBirza).
  - onSale(2004-10-30).
- RB1 entails: (among other conclusions)
  1. Price(Amazon,Sony5401,2004-10-12,BarbaraJones,49.99)
  2. Price(Amazon,Sony5401,2004-10-30,SalimBirza,39.99)
- RB2 = RB1 updated to add: onSale(2004-10-12).
- RB2 does NOT entail (1.). Instead (nonmonotonically) it entails:
  3. Price(Amazon,Sony5401,2004-10-12,BarbaraJones,39.99)

Brief Examples of Non-Stratified OLP

- RB3:
  - a.
  - $c \leftarrow a \wedge \sim b.$
  - $p \leftarrow \sim p.$
- Well Founded Semantics (WFS) for RB3 entails conclusions {a,c}.
  p is not entailed. p has "*undefined*" (u) truth value (in 3-valued logic).
- Stable Semantics for RB3: ill-defined; there *does not exist* a set of conclusions.
  (NOT: there is a set of conclusions that is empty.)

- RB4:
  - a.
  - $c \leftarrow a \wedge \sim b.$
  - $p \leftarrow \sim q.$
  - $q \leftarrow \sim p.$
- WFS for RB4 entails conclusions {a,c}. p,q have truth value u.
- Stable Semantics for RB4 results in two alternative conclusion sets: {a,c,p} and {a,c,q}. Note their intersection {a,c} is the same as the WFS conclusions.

Outline of Part B. Concepts & Foundations

1. Overview of Logical Knowledge Representations
   - First Order Logic (FOL), Logic Programs (LP); Complexity
2. Horn Case
3. Nonmonotonicity, Defaults, Negation, Priorities
   - Semantics for Default Negation
   - Courteous LP, Argumentation Theories
4. More Connectives and Quantifiers: Lloyd-Topor, Skolemization
5. Additional Features: Datatypes, Integrity Constraints, Equality, Aggregation
6. Procedural Attachments to Actions, Queries, Built-ins, and Events
   - Production/Situated LP, Production Rules
7. HiLog, Higher-Order Syntax, Reification, Meta-Reasoning
8. F-Logic, Frame Syntax, Object Oriented Style
9. Combining / Relating LP and FOL, e.g., Rule-based Ontologies
   - Rules for Description Logic: Description LP, DL-Safe
   - Weakened FOL in LP, Hypermonotonic Mapping
10. Recap
    - Outstanding Research Issues

Ubiquity of Priorities
in Commercially Important Rules -- and Ontologies

- Updating in relational databases
  - more recent fact *overrides* less recent fact
- Static rule ordering in Prolog
  - rule earlier in file *overrides* rule later in file
- Dynamic rule ordering in production rule systems (OPS5)
  - "meta-"rules can specify agenda of rule-firing sequence
- Event-Condition-Action rule systems rule ordering
  - often static or dynamic, in manner above
- Exceptions in default inheritance in object-oriented/frame systems
  - subclass's property value *overrides* superclass's property value, e.g., method redefinitions
- All lack Declarative KR Semantics

Semantical KR Approaches to Prioritized LP

The currently most important for Semantic Web are:

1. Courteous LP
   - KR extension to Ordinary LP
   - In RuleML, since 2001
   - Commercially implemented and applied
     - IBM CommonRules, since 1999
2. Defeasible Logic
   - Closely related to Courteous LP
     - Less general wrt typical patterns of prioritized conflict handling needed in e-business applications
     - In progress: theoretical unification with Courteous LP

Courteous LP: the What

- Updating/merging of rule sets: is crucial, often generates conflict.
- Courteous LP's feature prioritized handling of conflicts.
- Specify scope of conflict via a set of *pairwise* mutual exclusion constraints.
  - E.g., $\perp \leftarrow$ discount(?product,5%) $\wedge$ discount(?product,10%) .
  - E.g., $\perp \leftarrow$ loyalCustomer(?c,?s) $\wedge$ premiereCustomer(?c,?s) .
  - Permit classical-negation of atoms: $\neg$p means p has truth value *false*
    - implicitly, $\perp \leftarrow$ p $\wedge$ $\neg$p for every atom p.
- **Priorities** between rules: partially-ordered.
  - Represent priorities via reserved predicate that compares rule labels:
    - overrides(rule1,rule2) means rule1 is higher-priority than rule2.
    - Each rule optionally has a rule label whose form is a functional term.
    - overrides can be reasoned about, just like any other predicate.

Priorities are available and useful

- Priority information is naturally available and useful. E.g.,
  - recency: higher priority for more recent updates.
  - specificity: higher priority for more specific cases (e.g., exceptional cases, sub-cases, inheritance).
  - authority: higher priority for more authoritative sources (e.g., legal regulations, organizational imperatives).
  - reliability: higher priority for more reliable sources (e.g., security certificates, via-delegation, assumptions, observational data).
  - closed world: lowest priority for catch-cases.

- Many practical rule systems employ priorities of some kind, often implicit. E.g.,
  - rule sequencing in Prolog and production rules.
    - Courteous LP subsumes this as special case (totally-ordered priorities), plus enables: merging, more flexible & principled treatment.

Courteous LP: Advantages

- Facilitate updating and merging, modularity and locality in specification.
- Expressive: classical negation, mutual exclusions, partially-ordered prioritization, reasoning to infer prioritization.
- Guarantee consistent, unique set of conclusions.
  - **Mutual exclusion is enforced**. E.g., never conclude discount is both 5% and that it is 10%, nor conclude both p and $\neg$p.
- Scaleable & Efficient: low computational overhead beyond ordinary LP's.
  - Tractable given reasonable restrictions (VB Datalog):
    - extra cost is equivalent to increasing v to (v+2) in Ordinary LP, worst-case.
  - By contrast, more expressive prioritized rule representations (e.g., Prioritized Default Logic) add NP-hard overhead.
- Modular software engineering:
  - via courteous compiler: CLP $\rightarrow$ OLP.
    - A radical innovation. Add-on to variety of OLP rule systems. $O(n^3)$.

EEcoms Example of Conflicting Rules: Ordering Lead Time

- Vendor’s rules that prescribe how buyer must place or modify an order:
- A) 14 days ahead if the buyer is a qualified customer.
- B) 30 days ahead if the ordered item is a minor part.
- C) 2 days ahead if the ordered item’s item-type is backlogged at the vendor, the order is a modification to reduce the quantity of the item, and the buyer is a qualified customer.

- Suppose more than one of the above applies to the current order? **Conflict!**

- Helpful Approach: **precedence** between the rules. Often only *partial* order of precedence is justified. E.g., C > A.

Courteous LP's: Ordering Lead Time Example

- <leadTimeRule1> orderModificationNotice(?Order,14days)
- ← preferredCustomerOf(?Buyer,?Seller) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- <leadTimeRule2> orderModificationNotice(?Order,30days)
- ← minorPart(?Buyer,?Seller,?Order) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- <leadTimeRule3> orderModificationNotice(?Order,2days)
- ← preferredCustomerOf(?Buyer,?Seller) ∧
- orderModificationType(?Order,reduce) ∧
- orderItemIsInBacklog(?Order) ∧
- purchaseOrder(?Order,?Buyer,?Seller) .
- overrides(leadTimeRule3 , leadTimeRule1) .
- (⊥ ← orderModificationNotice(?Order,?X) ∧
- orderModificationNotice(?Order,?Y)) ← (?X ≠?Y) .

Courteous LP Semantics: *Prioritized argumentation in an opposition-locale.*

Conclusions from opposition-locales previous to this opposition-locale {p1,...,pk}

(Each pi is a ground classical literal. $k \geq 2$.)

↓

Run Rules for p1,...,pk

↓

Set of Candidates for p1,...,pk:
Team for p1, ..., Team for pk

↓

Prioritized Refutation

↓

Set of Unrefuted Candidates for p1,...,pk:
Team for p1, ..., Team for pk

↓

Skepticism

↓

Conclude Winning Side if any: at most one of {p1,...,pk}

Courteous feature: compileable, tractable

Courteous compiler

courteous representation
(Sit.) Courteous LP.

mutex* priorities

$\not\leftrightarrow$ $>$

$\equiv$ equivalent semantically

ordinary ("vanilla")
(Sit.) OLP representation

Sit. = Situated

* classical negation too

Tractable compilation:

O(n^3), often linear

Tractable inference: e.g., worst-case when no logical functions ("Datalog") & bounded v = |var's per rule| is equivalent to OLP with v $\rightarrow$ (v+2)

Preserves ontology.

Plus extra predicates for

- phases of prioritized argumentation (refutation, skepticism)
- classical negations

Argumentation Theories approach to Defaults in LP

- Recent results, reported in forthcoming paper "Logic Programs with Defaults and Argumentation Theories", by H. Wan, B. Grosof, M. Kifer, et al., at International Conf. on Logic Programming (ICLP) 2009. Available May 2009 on Web as preprint.
- Reformulates Courteous and other LP default approaches as a set of meta- style rules. Reducibility and well behavior results.
- Cleaner, more flexible and extensible semantics.
- Enables smooth and powerful integration of Courteous Defaults with Hilog and Frame syntax.
- Implemented in SILK V1 system, which extends Flora-2.

Outline of Part B. Concepts & Foundations

1. Overview of Logical Knowledge Representations
   - First Order Logic (FOL), Logic Programs (LP)
2. Horn Case
3. Nonmonotonicity, Defaults, Negation, Priorities
   - Semantics for Default Negation
   - Courteous LP, Argumentation Theories
4. More Connectives and Quantifiers: Lloyd-Topor, Skolemization
5. Additional Features: Datatypes, Integrity Constraints, Equality, Aggregation
6. Procedural Attachments to Actions, Queries, Built-ins, and Events
   - Production/Situated LP, Production Rules
7. HiLog, Higher-Order Syntax, Reification, Meta-Reasoning
8. F-Logic, Frame Syntax, Object Oriented Style
9. Combining / Relating LP and FOL, e.g., Rule-based Ontologies
   - Rules for Description Logic: Description LP, DL-Safe
   - Weakened FOL in LP, Hypermonotonic Mapping
10. Recap
    - Outstanding Research Issues

Hyper Logic Programs

- **SILK uses a new KR: Hyper Logic Programs (HLP)**
  - "Hyper" since it's Web (*hyper*text) centric, and it behaves *hyper*monotonically
- **It integrates several major LP extensions never previously combined:**
- **Higher-order and Frames and Skolemization, cf. F-Logic**
- **+ Defaults, cf. Courteous LP (and Defeasible Logic)**
  - **Newly generalized and modified approach**
- **+ Weakened Full Classical Logic, cf. Hypermonotonic mapping**
  - Greatly generalizes the approach of Description LP and OWL 2 RL
  - Unrestricted clauses, plus skolemization
  - Leverages Courteous feature
  - Give up disjunction / reasoning by cases, so is weakened
  - But behaves robustly in face of knowledge quality errors and conflictful merging

Basic Hypermonotonic Mapping from Clausal FOL to/from NAF-Free Courteous LP

- An FOL clause C:

  L1 or L2 or … or Lk

  is mapped to k <u>directed</u> clauses, one for each choice of head literal:

  L1 :- neg L2 and neg L3 and … and neg Lk

  L2 :- neg L1 and neg L3 and … and neg Lk

  …

  Lk :- neg L1 and neg L2 and … and neg Lk-1
- This is called the *omnidirectional ruleset* for C, a.k.a. the *omni rule(s)*
- Conversely, a naf-free Courteous LP rule is mapped to FOL as a material implication, thus clausal. (It's fairly easy to stick to naf-free.)
- A KR S *behaves hypermonotonically* == S is nonmonotonic and when its premises are viewed classically, then entailment in S is sound but incomplete w.r.t. classical
  - Incompleteness is desirable when there's conflict

Examples of Hypermonotonic mapping

- **/*** Car rental: A driver ?p is approved only if ?p has a validated rental application. */
  - /* FOL: */   forall ?p.   validated(?p)   <==   approved(?p).

  becomes the ff. omnidirectional ruleset in Hyper LP:
  - neg approved(?p) :- neg validated(?p) .
  - validated(?p)   :-   approved(?p).
- /* Scheduling: Joe's meeting will be at 3pm or 4pm or 5pm today. */
  - /* FOL source: */ mtg(3p) or mtg(4p) or mtg(5p).

  becomes the ff.
  - mtg(5p) :- neg mtg(3p) and neg mtg(4p).
  - mtg(4p) :- neg mtg(3p) and neg mtg(5p).
  - mtg(3p) :- neg mtg(4p) and neg mtg(5p).
- /* OWL beyond DLP: A and B are disjoint.   P on C has min cardinality 1. */
  - /* FOL */ forall ?x. neg (A(?x) and B(?x)).

  becomes the ff.
  - neg A(?x) :- B(?x).   /* Exploit neg .   */
  - neg B(?x) :- A(?x).
  - P(?x, _# (?x)) :- C(?x).   /* Exploit skolemization   feature.   */
  - neg C(?x) :- neg P(?x,_# (?x)).

Hypermon Mapping from FOL++ to LP

- Greatly generalizes the approach of Description LP and OWL 2 RL
- Leverages Courteous feature of Hyper LP
- Covers unrestricted FOL clauses, plus skolemization, thus full FOL
- Can further add Frames and Hilog (and deontic etc. modals, esp. using Hilog)
- Thus can cover full OWL/RDF and Common Logic, most of SBVR
- Give up disjunction / reasoning by cases, so is weakened
- But Courteous/Hyper LP handles conflict robustly
  - Whereas FOL is perfectly brittle semantically in face of contradictions from …
  - Quality problems/errors in the data and knowledge
  - Conflict when merging KBs

A VLKB with a million or billion axioms formed by merging from multiple Web sources, is unlikely to have zero KB/KA conflicts from

  - Human knowledge entry/editing, implicit context, updating cross

Example of Causal Process Reasoning in SILK

- **/*   Toxic discharge into a river causes fish die-off.   */**
- **/* Initial facts, and "mutex" constraint that fish count has a unique value */**
- **occupies(trout,Squamish).**
- **fishCount(s0,Squamish,trout,400).**
- **!- fishCount(?s,?r,?f,?C1) and fishCount(?s,?r,?f,?C2) | ?C1 != ?C2.**
- **/* Action/event description that specifies causal effect on next state */**
- **{tdf1} fishCount(?s+1,?r,?f,0) :- occurs(?s,toxicDischarge,?r) and occupies(?f,?r).**
- **/* Persistence ("frame") axiom */**
- **{pe1} fishCount(?s+1,?r,?f,?p) :- fishCount(?s,?r,?f,?p).**
- **/* Action effect axiom has higher priority than persistence axiom */**
- **{pr1} overrides(tdf1,pe1).**
- **/* An action instance occurs */**
- **{UhOh} occurs(s0+1,toxicDischarge,Squamish).**

- ***As desired:*** **|=   fishCount(s0+1,Squamish,trout,400) and**
  fishCount(s0+2,Squamish,trout,0).

PART C. SLIDES FOLLOW

Outline of Part C. Conclusions & Directions

1. More about Key Tools
2. Conclusions
3. Directions for Future research

4. References and Resources

More about Tools

1. *Rule systems designed to work with RDF/OWL*
   a. *Commercial-world: Jena; Oracle; other*

- Jena-2 SW suite has rule capabilities
  - Open source, popular, Java, basic Horn-ish
  - backward engine, forward engine (slow)
  - Works well with RDF(S). Suite includes OWL capabilities.
- Oracle has rule capabilities in semantic tech suite, as part of its flagship database platform
  - Basic Horn-ish, implements DLP+ subset of OWL, thus supports RDF(S)
  - Oracle also has other production-rule type products, including recently acquired Haley Ltd. – a leader in NL KA.
  - In development: support for W3C RIF Basic Logic Dialect
- Various others, e.g., Ontotext, Ontoprise Ontobroker, Versatile Information Systems

More about Tools

1. *Rule systems designed to work with RDF/OWL*
   b. *Research-world: SweetRules; cwm; others*

- SweetRules has translator from DLP subset of OWL to LP Rules in RuleML and SWRL. Open source, Java. Not maintained.
- Cwm implements N3 – rules + RDF. Semantically hazy in some regards, but overlaps a lot with LP. Open source, Python.
- Numerous others, e.g., combining SWRL with Jess
  - Protégé OWL Plug-in has SWRL+ rules; OWLJessKB

- SweetRules also includes a variety of capabilities for high expressiveness (e.g., prioritized defaults, action aproc's) and fully semantic interoperability between Jess production rules and XSB Prolog.

More about Tools

2. *Prolog and Production Rule systems*
   - XSB: semantic, Prolog, full WFS negation, fast, C with available Java front end (Interprolog)
   - Jess: production rules, popular, Java, free for non-commercial but not open source

   - Pointer generally: OpenRuleBench
     - WWW-2009 paper on rule systems benchmarking study, by S. Liang, M. Kifer, *et al.*
     - XSB, Flora-2, Ontobroker, YAP Prolog, DLV all did well

More about Tools

3. *Advanced Expressiveness*
   - Flora-2: open source, built in/on XSB Prolog, has Hilog, Frame, reification, skolemization features
   - SILK (in development): extends Flora-2 with Courteous defaults, aproc's, hypermonotonic translation, API's. Deltas in Java. Planned to be free for non-commercial, release in 2009 or 2010.
   - IBM CommonRules (1999-) supports Courteous Defaults and Action Aproc's. Cheap or free, Java.

4. *Rules in Semantic Wikis*
   - Semantic MediaWiki + (in development, planned fall 2009). Open source, PHP. Part of Vulcan/Ontoprise Halo extension to basic SMW.

More about Tools

5. *Some Available Large Rule Bases*

- OpenCyc / ResearchCyc
  - Open source / free for non-commercial
  - 3 Million axioms. Large 20 year effort.
  - Idiosyncratic semantically, but overlaps with LP. Translation to SILK is in development.

- Open Process Handbook
  - Open source. Semantic Wiki –ish.
  - 5,000 business processes, each with ~10 axioms
  - Lots of text and links too. 15 year effort.

- OpenMind – collaborative commonsense KB
  - Open source. ~1 Million axioms (roughly). Built by Web users.
  - Lacks declarative semantics.

Conclusions

1. Theme: Centrality to Web
   - More than most people realize, LP Rules are central to the Web, both current and future
   - Relational, XML, and RDF databases/querying is LP
   - Thriving commercial business rules market sector, based on production rules / event-condition-action rules, is moving to the Web, and translates largely to LP
   - Often used for ontologies: represent, implement, map
   - Semantic tech and semantic web is largely already LP-based

Conclusions, continued

2. Theme: Incremental Evolution
   - LP Rules, and Semantic Web overall, is incremental technologically wrt relational and Web DBMS
3. Theme on KR expressiveness: Reducibility
   - LP feature extensions built up in layers
   - E.g., Lloyd-Topor, Hilog, Frame syntax, Courteous Defaults each reduce tractably to Normal LP

Key Directions for Future Research

1. Expressiveness
   - Relationship between FOL and Default LP
   - Disjunction, Probabilistic, Abduction, Fuzzy
   - Misc. smaller issues, e.g., aggregation
2. Reasoning performance
   - Forward-direction, truth maintenance, termination
   - Parallelization
3. Knowledge acquisition and UI
   - Explanation
   - Limited natural language
   - Business users / Subject Matter Experts (SME's)
   - Collaboration
4. Applications and Tools
   - Build, experiment

ADDITIONAL REFERENCES & RESOURCES FOLLOW

N.B.: some references & resources were given on various earlier slides

Longer Previous Version of This Tutorial

- http://www.mit.edu/~bgrosof/#ISWC2006RulesTutorial
  - Full-day length tutorial
  - But from three years ago, for Semantic Web -y audience
  - Includes much additional material on KR features, applications

References & Resources I: Standards on Rules and Ontologies

- http://www.ruleml.org RuleML *Includes links to some tools and examples.*
- http://www.w3.org/Submission/2004/SUBM-SWRL-20010521 SWRL
  - http://www.daml.org/committee Joint Committee. Besides SWRL (above) this includes:
    - http:///www.daml.org/2004/11/fol/ SWRL-FOL
    - http://www.ruleml.org/fol FOL RuleML (also see RuleML above)
  - http://www.daml.org/rules DAML Rules
- http://www.swsi.org Semantic Web Services Initiative. Especially:
  - Semantic Web Services Language (SWSL), incl. SWSL-Rules and SWSL-FOL and overall requirements/tasks addressed
- http://cl.tamu.edu Common Logic (successor to Knowledge Interchange Format)

- Also: Object Management Group (OMG) has efforts on rules and ontologies (cooperating with RuleML and W3C)
- Also: JSR94 Java API effort on Rules (cooperating with RuleML)

References & Resources II: Standards on Rules and Ontologies

- http://www.w3.org World Wide Web Consortium, esp.:
  - …/2005/rules/ Rule Interchange Format
  - …/2001/sw/ Semantic Web Activity, incl. OWL and RDF
  - …/2002/ws/ Web Services Activity, incl. SOAP and WSDL
  - www-rdf-rules@w3.org Rules discussion mailing list
  - www-sws-ig@w3.org Semantic Web Services discussion mailing list
  - P3P privacy policies
  - XQuery XML database query
- http://www.oasis-open.org Oasis, esp. on web policy & web services:
  - XACML XML access control policies
  - ebXML e-business communication in XML
  - Legal XML
  - BPEL4WS Business Processes as Web Services
  - Web Services Security

Ref's & Resources III: LP with Negation

- Przymusinski, T., "Well Founded and Stationary Models of Logic Programs", Annals of Artificial Intelligence and Mathematics (journal), 1994. *Constructive model theory, and proof theory, of well founded semantics for LP.*
- Van Gelder, A., Schlipf, J.S., and Ross, K.A., "The Well-Founded Semantics for General Logic Programs", Journal of the ACM 38(3):620-650, 1991. *Original theory of well founded semantics for LP.*
- Gelfond, M. and Lifschitz, V., The Stable Model Semantics for Logic Programming, Proc. 5th Intl. Conf. on Logic Programming, pp. 1070-1080, 1988, MIT Press. *Original theory of stable semantics for LP. Answer set programs extend this.*
- Lloyd, J.W., "Foundations of Logic Programming" (book), 2nd ed., Springer-Verlag, 1987. *Includes Lloyd-Topor transformation, and correspondence of semantics to FOL in definite Horn case. Reviews theory of declarative LP. Somewhat dated in its treatment of theory of NAF since it preceded well founded and stable semantics.*
- Baral, C., and Gelfond, M., "Logic Programming and Knowledge Representation", J. Logic Programming, 1994. *First and last parts review theory of declarative LP. Stronger on stable semantics than on well founded semantics.*
- Gelfond, M., "Answer Sets" (book chapter 7). In: Handbook of Knowledge Representation. Elsevier, 2007. *Up-to-date exposition of answer set programs.*

References & Resources IV: Misc. on Rules and Ontologies

- http://ccs.mit.edu/ph MIT Process Handbook, incl. Open Process Handbook Initiative
- Grosof, B., Horrocks, I., Volz, R., and Decker, S., "Description Logic Programs: Combining Logic Programs with Description Logic", Proc. 12th Intl. Conf. on the World Wide Web., 2003. *On DLP KR and how to use it.*
- Grosof, B., "Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML", Electronic Commerce Research and Applications (journal) 3(1):2-20, 2004. *On situated courteous LP KR, RuleML overview, and e-commerce applications of them.*
- Grosof, B. and Poon, T., "SweetDeal: Representing Agent Contracts with Exceptions using Semantic Web Rules, Ontologies, and Process Descriptions", Intl. Journal of Electronic Commerce 8(4), 2004. *On SweetDeal e-contracting app.*
- Firat, A., Madnick, S., and Grosof, B., "Financial Information Integration in the Presence of Equational Ontological Conflicts", Proc. Workshop on Information Technologies and Systems, 2002. *On ECOIN. Also see A. Firat's PhD thesis, 2003.*

- *(Additional references to be posted on website version of these slides.)*

References & Resources V: Misc. on Rules and Ontologies

- Grosof, B., Gandhe, M., and Finin, T., "SweetJess: Translating DamlRuleML To Jess". Proc. Intl. Wksh. On Rule Markup Languages for Business Rules on the Semantic Web, 2002 (the 1st RuleML Workshop, held at ISWC-2002). See extended and revised working paper version, 2003. *On SweetJess translation/interoperability between RuleML and production rules.*
- Forgy, C.L., "Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem". Artificial Intelligence 19(1):17-27, 1982. *On the key Rete algorithm for production rules inferencing.*
- Friedman-Hill, E., "Jess in Action" (book), 2003. *On Jess and production rules.*
- Ullman, J., "Principles of Knowledge Base and Database Systems Vol. I" (book), 1988. *See esp. the chapter on Logic Programs, incl. algorithm for stratification.*
- http://xsb.sourceforge.net XSB Prolog. See papers by D. Warren *et al.* for theory, algorithms, citations to standard Prolog literature (also via http://www.sunysb.edu/~sbprolog)
- *(ff. needs tweaking:)* Horrocks, I., and Patel-Schneider, P., paper on OWL Rules and SWRL, Proc. WWW-2004 Conf., 2004. *On SWRL theory incl. undecidability*.
- *(ff. needs tweaking:)* Horrocks, I.., and Bechhofer, S., paper on Hoolet approach to SWRL inferencing via FOL theorem-prover, Proc. WWW-2004 Conf., 2004. *On SWRL inferencing.*

References & Resources VI: More on Courteous and Situated

- Grosof, B., Labrou, Y., and Chan, H., "A Declarative Approach to Business Rules in Contracts", Proc. 1st ACM Conf. on Electronic Commerce, 1999, ACM Press. *On courteous LP KR with mutex's, and its e-contracts applications.*
- Grosof, B., "Courteous Logic Programs: Prioritized Conflict Handling for Rules", Proc. Intl. Logic Programming Symposium., 1997. See extended version: IBM Research Report RC 20836, 1997. *Basic version courteous LP (since generalized).*
- Grosof, B., "A Courteous Compiler from Generalized Courteous Logic Programs To Ordinary Logic Programs", (IBM) research report extension to "Compiling Courteous Logic Programs Into Ordinary Logic Programs", 1999. Available via http://ebusiness.mit.edu/bgrosof or IBM incl. in CommonRules documentation. *Details on courteous compiler/transform.*
- Grosof, B., Levine, D.W., Chan, H.Y., Parris, C.J., and Auerbach, J.S., "Reusable Architecture for Embedding Rule-based Intelligence in Information Agents", Proc. Wksh. on Intelligent Information Agents, at ACM Conf. on Information and Knowledgte Management, ed. T. Finin and J. Mayfield, 1995. Available also as IBM Research Report RC 20305. *Basic situated LP paper. Also see 1998 patent.*
- Grosof, B., "Building Commercial Agents: An IBM Research Perspective (Invited Talk). Proc. 2nd Intl. Conf. on the Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM97), pub. The Practical Applications Company, 1997. Also available as IBM Research Report RC 20835. *Overview of situated LP.*

Resources VII: Web Services Applications

- http://zdnet.com.com/2100-1106-975870.html Fidelity’s web services for EAI
- http://www.amazon.com/gp/browse.html/ref=smm_sn_aws/002-8992958-7364050?node=3435361 Amazon’s web services – 1000’s of developers

Resources VII: Web Services Applications

- http://zdnet.com.com/2100-1106-975870.html Fidelity's web services for EAI
- http://www.amazon.com/gp/browse.html/ref=smm_sn_aws/002-8992958-7364050?node=3435361 Amazon's web services – 1000's of developers

Resources VIII: Papers

The following papers, available on the web, cover major portions of the tutorial's content (altogether roughly half):

- "Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML", by B. Grosof, *Electronic Commerce Research and Applications (ECRA)* 3(1):2-20, Spring 2004.

- "Semantic Web Services Framework" (SWSF), V1.0+, by Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D., Su, J., and Tabet, S. (alphabetic), May 2005. Technical Report (~200 pages).

- "Logical Foundations of Object-Oriented and Frame-Based Languages", by M. Kifer, G. Lausen, and J. Wu, *.J. ACM* 42:741-843, 1995.

- "SweetDeal: Representing Agent Contracts with Exceptions using Semantic Web Rules, Ontologies, and Process Descriptions", by B. Grosof and T. Poon, *International Journal of Electronic Commerce (IJEC)* 8(4):61-98, Summer 2004.

- "HiLog: A Foundation for Higher-Order Logic Programming", by W. Chen, M. Kifer, and D.S. Warren, *J. Logic Programming* 15(3):187-230, Feb. 1993.

- "Description Logic Programs: Combining Logic Programs with Description Logic", by B. Grosof, I. Horrocks, R. Volz, and S. Decker, Proc. 12th Intl. Conf. on the World Wide Web (WWW-2003), 2003.

Resources IX: Papers (cont'd)

- "SWRL: A Semantic Web Rules Language Combining OWL and RuleML", V0.7+, by I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, Nov. 2004. Technical Report.
- RuleML website, especially design documents and list of tools. Ed. by H. Boley, B. Grosof, and S. Tabet, 2001-present.

Content for the tutorial will also be drawn, to a lesser degree, from about a dozen other papers/resources available on the web, e.g.,:

- "Web Service Modeling Ontology (WSMO)" by J. de Bruijn et al., 2005. Technical Report.
- "A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML", by B. Grosof et al., Proc. EC-99.
- "A Policy Based Approach to Security for the Semantic Web", by Kagal et al., Proc. ISWC-2003.
- "Financial Information Integration in the Presence of Equational Ontological Conflicts", by A. Firat et al., WITS 2002 conf.
- "DAML+OIL for Application Developers", http://www.daml.org/2002/03/tutorial/Overview.html
- "Delegation Logic: A Logic-based Approach to Distributed Authorization", *ACM Trans. on Info. Systems Security (TISSEC)*, by N. Li et al., 2003

THANK YOU!

Disclaimer: The preceding slides represent the views of the authors only.
All brands, logos and products are trademarks or registered trademarks of their respective companies.

END OF ALL SLIDES

Disclaimer: The preceding slides represent the views of the authors only.
All brands, logos and products are trademarks or registered trademarks of their respective companies.