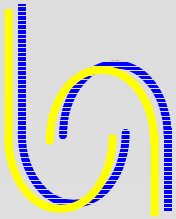


Semantic Web goes Mainstream

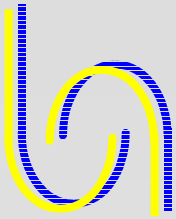
19. November 2002

Freie Universität Berlin

Andreas Eberhart

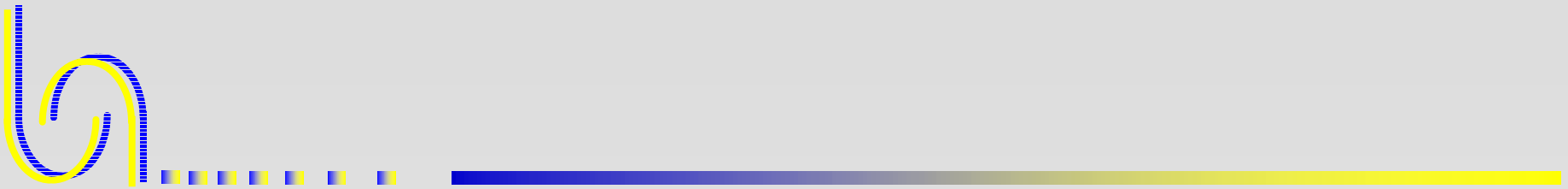
International University in Germany

http://www.i-u.de/schools/eberhart/

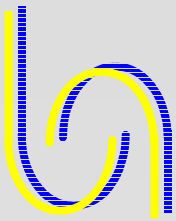


Overview

- What‘s wrong with today‘s web?
- A whirlwind tour of the Semantic Web
- Where are we today?
- The Semantic Web goes mainstream
  - OntoSQL – using a DBMS for inferencing
  - WSDF – integrating "legacy" web services
  - Sample Applications
- Outlook

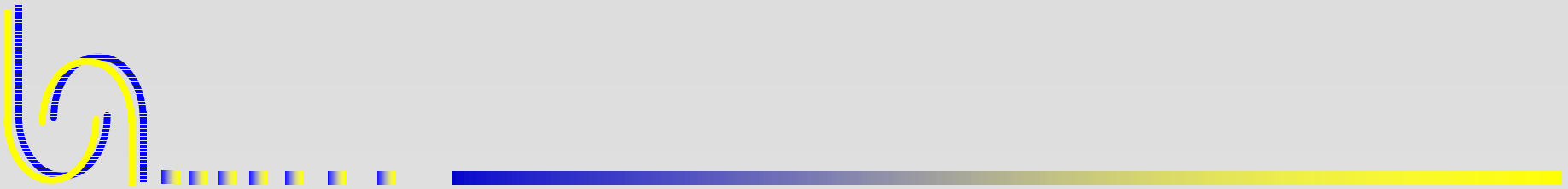


What‘s wrong with today‘s web?

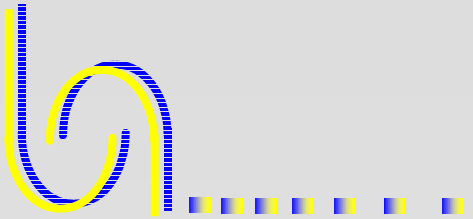


Problem: Web is geared towards Humans

- Almost anything can be ordered, managed, or bought over the web
- However, all these services must be used manually
- To organize a trip to London we must
  - Check www.bahn.de for the train schedule
  - Check www.lufthansa.com for the plane tickets
  - Use www.hertz.com for a rental car
  - Use www.holidayinn.com to book the hotel
- Synchronizing the schedule or finding the best price is a complicated task
  - That‘s why there are travel agents



A whirlwind tour of the Semantic Web

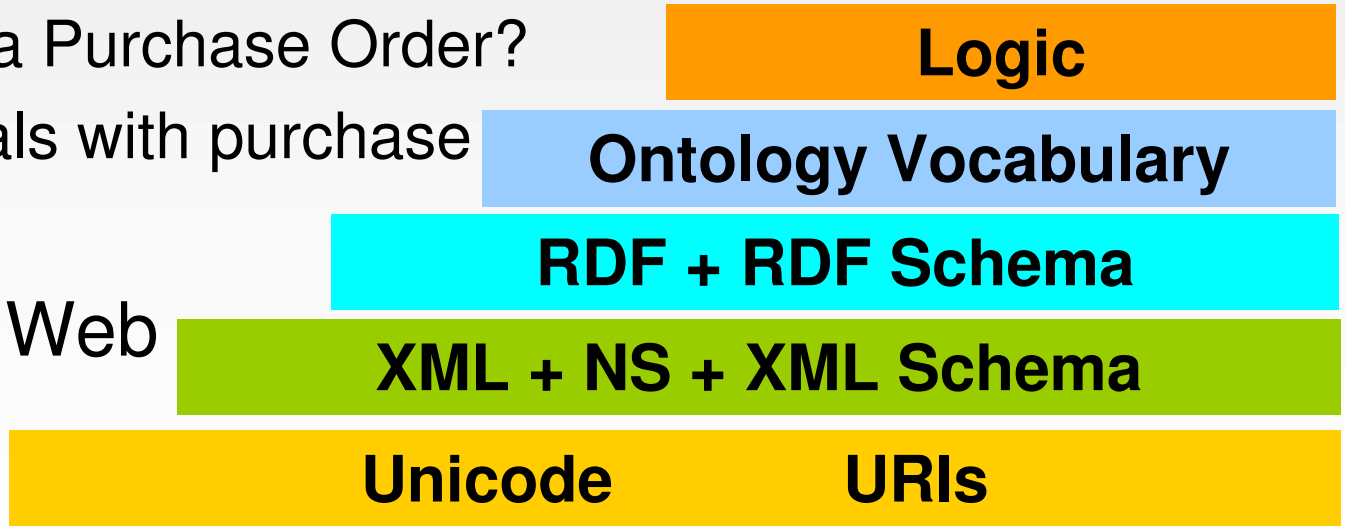


Semantic Web Vision

- Internet is currently geared towards manual interaction
- Web Services are the current state of the art technology in B2B systems
  - WSDL, UDDI allow some dynamic discovery
  - But, the approaches are based on standardized APIs such as RosettaNet, ebXML
- Vision
  - Use (cross-linked) Ontologies to describe a domain
  - Agents only need to be aware of the Ontology in order to interact with each another

Where do we go from here: Semantic Web

- The standardization of APIs is a successful approach
- Nevertheless it has shortcomings
- To arrive at a flexible environment, the software needs to be aware of the application domain
  - What is a Purchase Order?
  - Who deals with purchase
  - etc.
- Semantic Web ML Stack



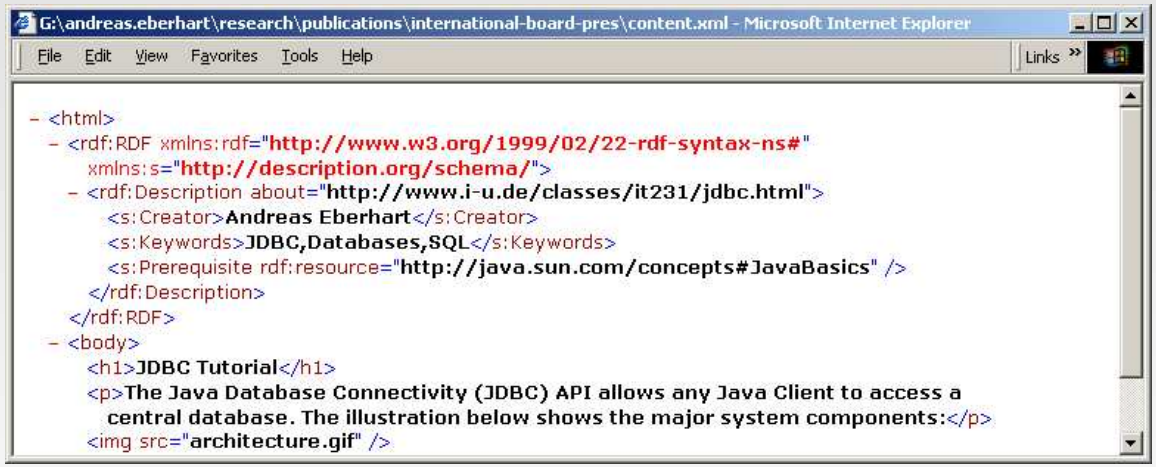
Resource Description Framework (RDF)

- W3C recommendation from 1999
  - Has its origins in the platform for internet content selection (PICS)
- RDF is a format to describe metadata
  - Everything can be identified via a URI
  - Graphical notation for the fact that a URI deals with the topic „Appliction Server“

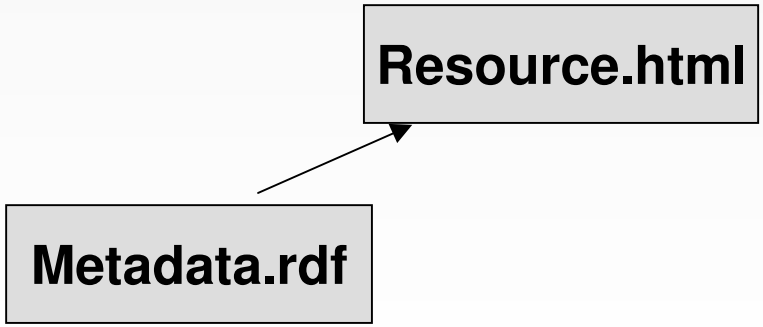
Triple: subject predicate object

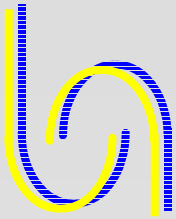
Technical representation

- RDF data can appear in various forms
  - Can be stored in the <head> element of HTML pages



  - Can be in transmitted as its own document (if generated from data stored in an information system)





RDF Standard Vocabularies

- At the moment, the only commonly used vocabulary is the Dublin Core
  - Typical content management information that is useful for searches (Title, Author, Format, Keywords, Language, Audience, etc.)
  - http://dublincore.org/
- Due to its wide acceptance, Dublin Core labels can be correctly interpreted by any application
  - DC website lists about 70 projects that use DC metadata (Australian Government Locator Service, Digital Library Catalog, etc.)
- There are related standards that borrow from DC
  - IEEE Learning Object Metadata (LOM)

RDF Schema (RDFS)

- RDF Schema defines the classes and the names and domains of the graph‘s arcs
  - In principle, RDFS relates to RDF like XML Schema relates to XML

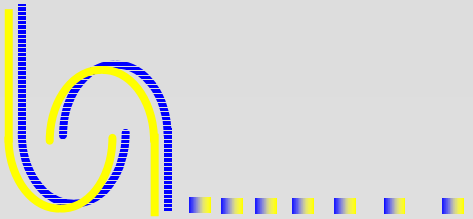
```
<rdf:Description ID="Truck">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="registeredTo">
    <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>
    <rdfs:domain rdf:resource="#MotorVehicle"/>
    <rdfs:range rdf:resource="#Person"/>
</rdf:Description>
```

Definition & use of the class `Truck`

- RDF data conforming to this schema:

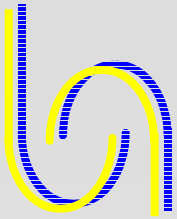
```
<rdf:RDF xmlns:schema="schema.rdfs"
    <rdf:Description about="#truck15">
        <rdf:type resource="schema.rdfs#Truck"/>
        <schema:registeredTo resource="#Peter">
    </rdf:Description>
```

Definition & use of the property `registeredTo`



What is an Ontology?

- Encyclopedia Britannica:
  Ontology: the theory or study of being as such; i.e., of the basic characteristics of all reality
  - Not really a suitable definition for us
- An ontology describes important concepts of a domain and consists of the following components
  - A taxonomy of concepts
  - Relationships between the concepts
  - Facts, Rules, Constraints
- Ontology vs. Schema
  - Ontologies are not concerned with a specific application
  - Bridges the “Conceptual Gap”



Distributed Agent Markup Language

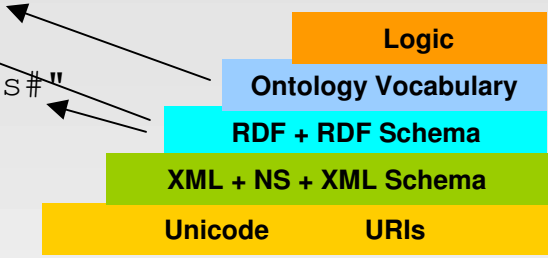
- DAML bases on RDFS
  - http://www.daml.org/
  - Now called DAML+OIL
  - European Ontology interchange language (OIL) effort joined forces with DAML, now W3C version called OWL
- Allows new modeling constructs
  - Classes can be marked as disjoint
  - Cardinalities of relations can be defined
  - Relations can be marked as transitive or symmetric
- Available software / data
  - Not many DAML+OIL engines yet
  - DAML checker
  - Very small library of ontologies. Major entry: http://www.opencyc.org/

DAML Example

- Namespace prolog:

```
<rdf:RDF
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#">
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

DAML builds on RDF(S)



- Defining classes:

```
<daml:Class rdf:ID="Female">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>
```

- Defining properties:

```
<daml:ObjectProperty rdf:ID="hasFather">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
    <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

- Defining property restrictions

```
<rdfs:Class rdf:"ID=Person">
    <rdfs:subClassOf rdf:resource="#Animal"/>
    <rdfs:subClassOf>
        <daml:Restriction>
            <daml:onProperty rdf:resource="#hasParent"/>
            <daml:toClass rdf:resource="#Person"/>
        </daml:Restriction>
    </rdfs:subClassOf>
```

Rule ML Initiative

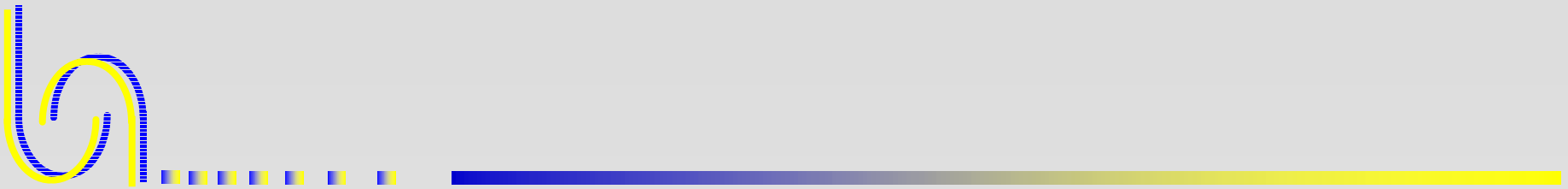
- There are many different rule mechanisms and languages
  - SQL Triggers
  - XSL Stylesheets
  - IBM CommonRules
  - Lisp / Prolog
- RuleML is an effort to promote the sharing of rules
  - http://www.dfki.uni-kl.de/ruleml/
- Latest news: DAML Rules

```
<imp>
  <_head>
    <atom>
      <_opr>
        <rel>premium</rel>
      </_opr>
      <var>customer</var>
    </atom>
  </_head>
  <_body>
    <atom>
      <_opr>
        <rel>spending</rel>
      </_opr>
      <var>customer</var>
      <ind>min 5000 euro</ind>
      <ind>previous year</ind>
    </atom>
  </_body>
</imp>
```

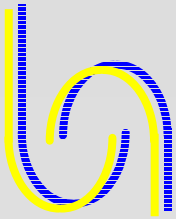
You are a premium customer if you spent at least 5000 € in the previous year

Free variable

Ontology vocabulary



Where are we today?

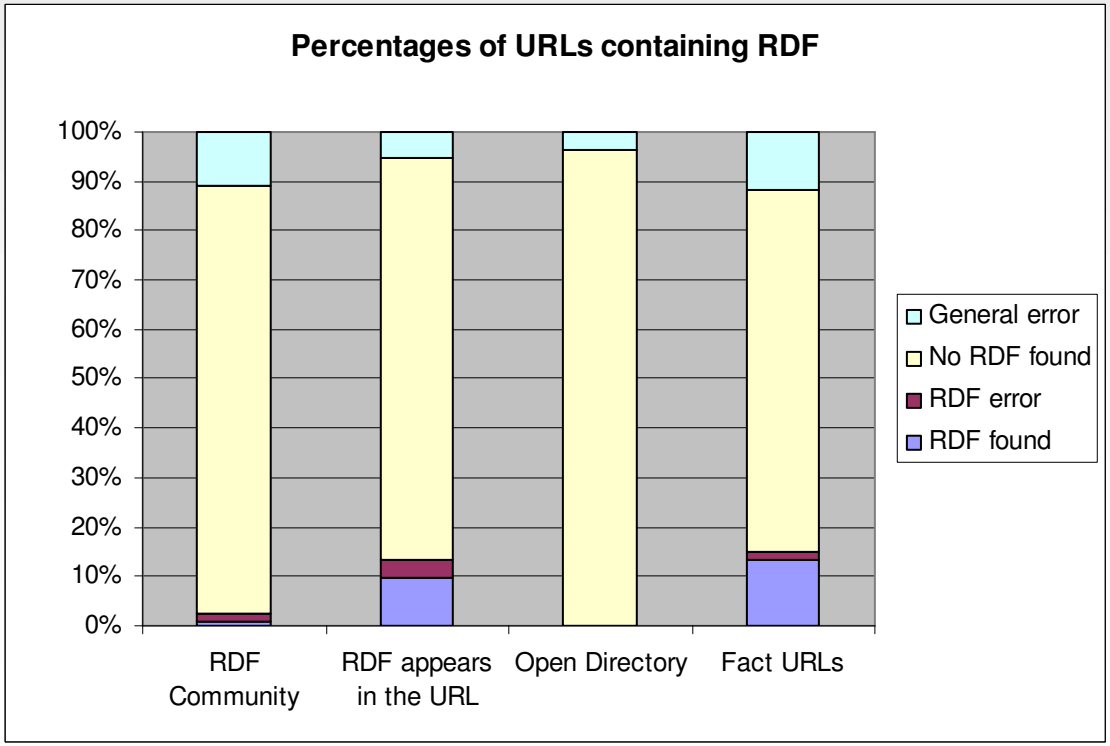


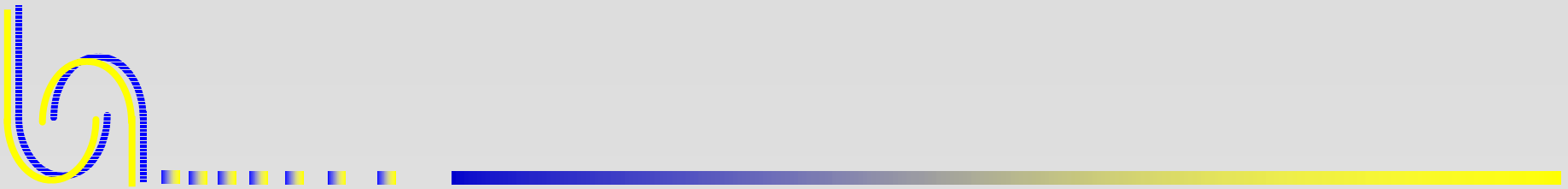
Status Report of the SW Effort

- Much interest and excitement
- Several tools are available
  - Ontology editing
  - Ontology storage
  - Inferencing
  - ...
- What‘s missing?
  - The killer app
  - Large deployments
  - Large collections of data and ontologies

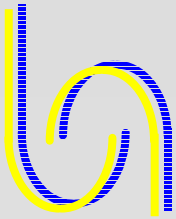
RDF Crawling Results

- How much RDF is there?
  - Searched 12507 pages close to SW portals
  - Searched 1256 pages containing RDF in the URL
  - Searched 527408 pages from Open Directory





The Semantic Web goes mainstream

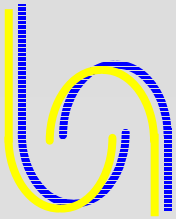


Vision

- Our view of the Semantic Web:
  - Enabling technology which makes the development of classical AI applications easier
  - Example: ThoughtTreasure Ontology format

```
===soda.¹z//soft#A drink*.z/fizzy#A drink*.gz/
    carbonated#A soft#A drink*.z/
pop,soda# pop*,coke.îz/tonic.oîz/soda.¹My/
```

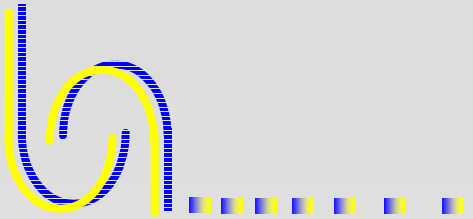
- What is important?
  - Sharing and reuse (definitely not the case today)
  - Scalability of an application is more important than its complexity
  - Example: CiteSeer
  - Integration of mainstream technology such as databases or Web Services



Data Storage

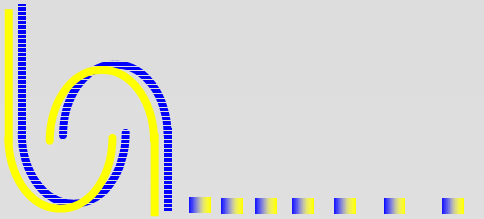
- A relational database is the obvious choice for storing data
  - Rich tool support
  - Example: MS Access as a data input front-end
- Triple based tables

Table: marriedToFact	
subject	object
Peter	Jane



Rules / Inference

- Rules are an important part of the Semantic Web architecture
- Problem:
  - Myrad of different rule semantics
  - Try to start with a simple rule dialect: Datalog
  - SW version of Datalog is RuleML 0.8



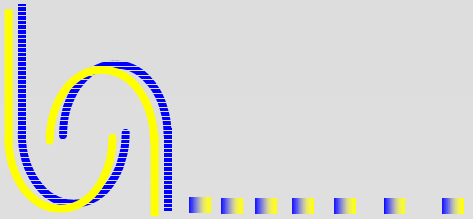
OntoSQL

- Idea: push the rule execution into the DB
- Datalog2SQL is „old“ technology

```
isSiblingOf(B, C)<-and(isParentOf(A, B), isParentOf(A, C))
```

```
create view isSiblingOf as
(
    select * from isSiblingOfFact
    union
    select t0.o s, t1.o o
      from isParentOf t0, isParentOf t1
      where t0.s = t1.s
)
```

- Advantage: rules are complete transparent in SQL interface



Recursive Datalog

- SQL99 allows for recursive view definitions
  - Implemented by IBM DB2 ver. 7
  - Workaround could also be a sequence of self-joins

```
create view isAncestorOf as
    with rec(subject, 'isAncestorOf', object, level) AS (
        select * from fact where predicate = 'isAncestorOf'
        union all
        select subject, 'isAncestorOf', object from Parent
        union all
        select a.subject, 'isAncestorOf', b.object, level+1
            from rec a, Parent b
            where a.object = b.subject and level < 9
    )
    select * from ancestor;
```

Integrity Constraints

- We also need to enforce RDFS semantics
- Use SQL triggers
  - check not powerful enough
  - create assertion not implemented
- Currently no tool sup.


```
create trigger fatherOfSignature
on fatherOfFact, typeFact
for insert, update, delete
as
    if exists (
        select * from fatherOf where subject not in
            (select subject from type where object = 'Male')
    )
        begin
            raiserror ('Fathers must be Male', 16, 1)
            rollback transaction
        end
```

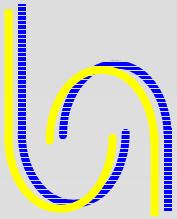
Integrating Legacy Apps.

- Normally data wouldn‘t be stored in fact tables
- Idea: use the view mechanism again:
  - SW view into the corporate IT system
  - Updatable views:
    update hasAge set age=40 affects base table

Table: personnel			
id	name	age	dept
23	Peter	35	IT

Table: hasAge	
subject	object
Peter	35

```
create view hasAge as
(
    select name as subject, age as object
    from personnel
)
```

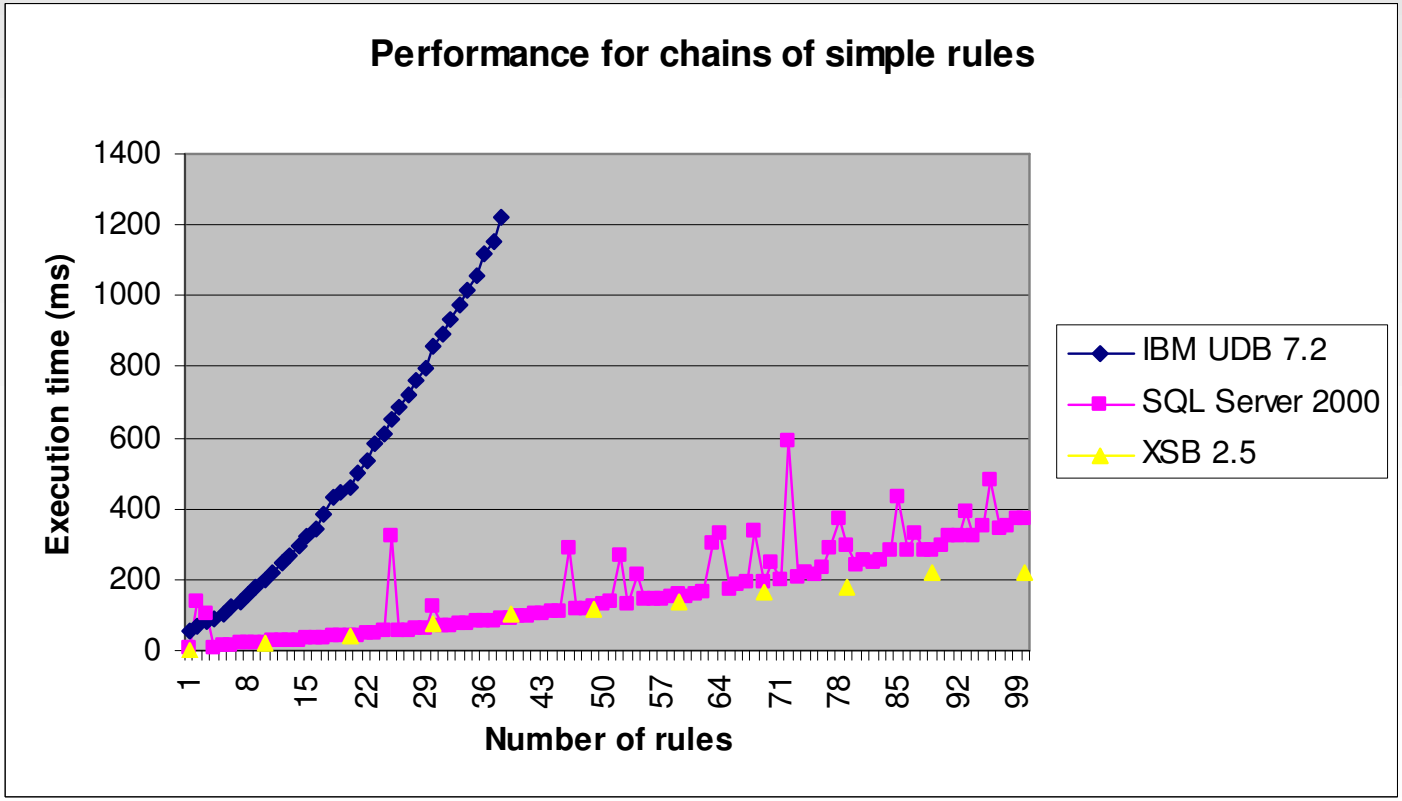


OntoSQL Performance Results

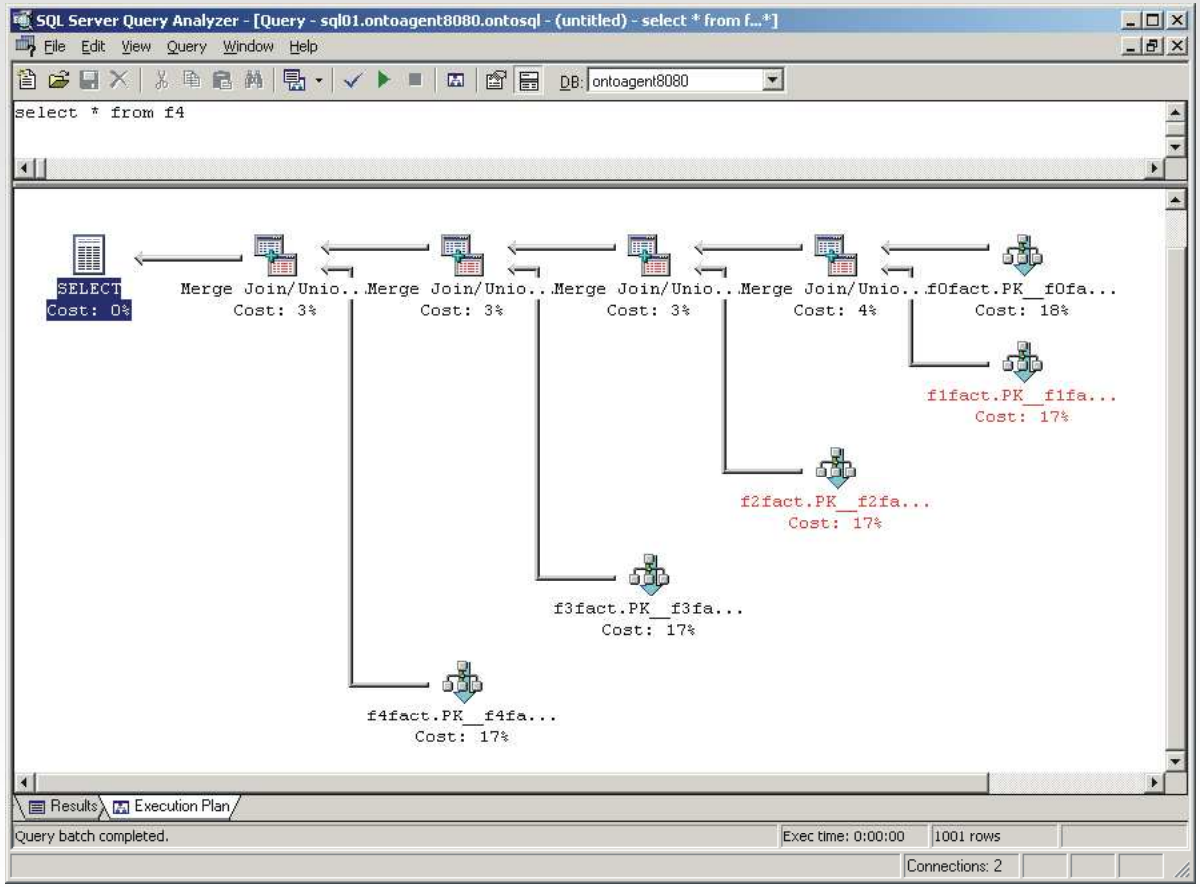
- Test the feasibility of using relational databases as inference engines
- Compare DBMS with popular XSB system which is the basis for many systems (Triple, OntologyWorks)
  - Microsoft SQL Server 2000 (measure in JDBC client server setting)
  - IBM DB2 (measure in JDBC client server setting)
  - XSB 2.5 (measure in process using the findall predicate)
- Very encouraging results ...

A Chain of Simple Rules

- Rules: p1 ← p0, p2 ← p1, ...
- Each predicate fact table contains 1000 values

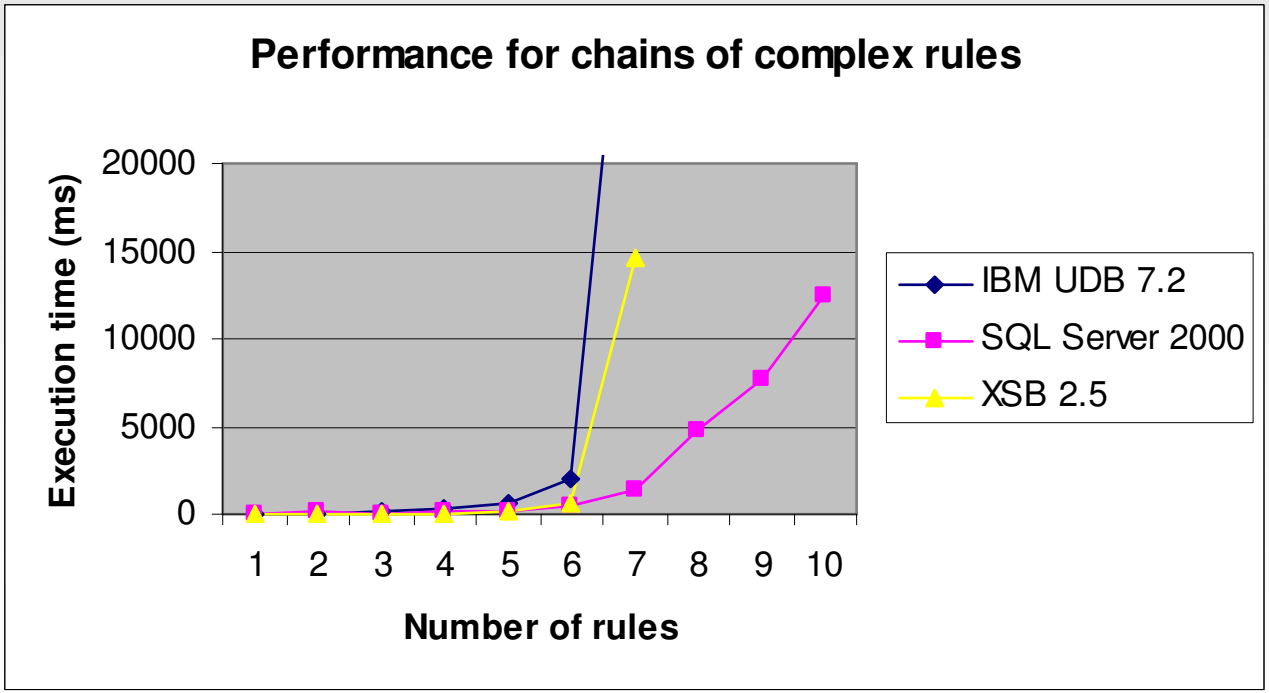


SQL Server Execution Plan



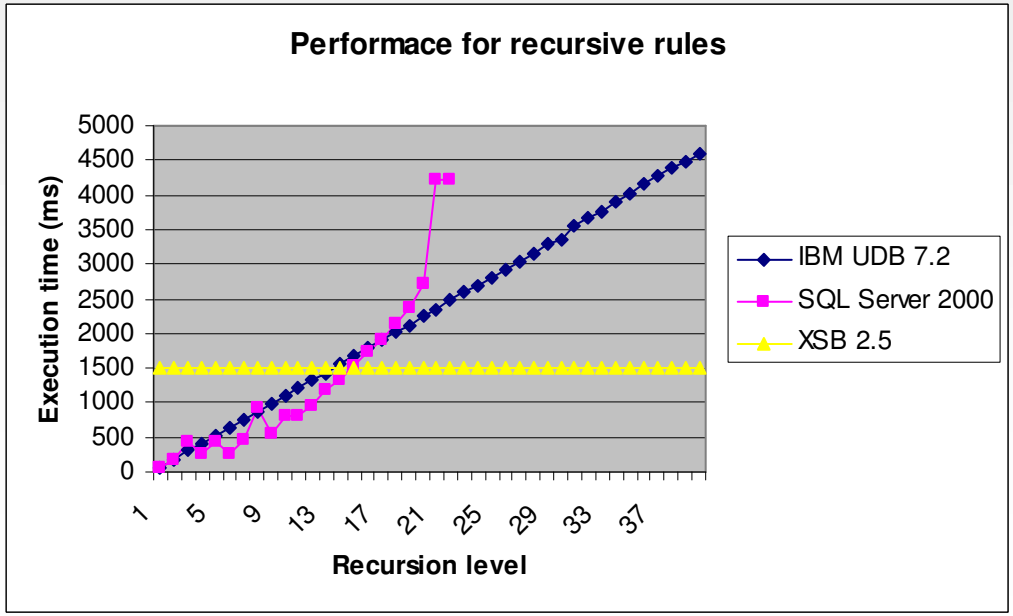
A Chain of Complex Rules

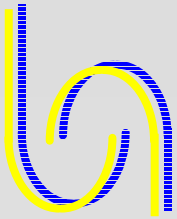
- Rules: p1 ← p0, p2 ← p1 ∧ p0, p3 ← p2 ∧ p1, ...
- Fibonacci like series



Recursive Rules

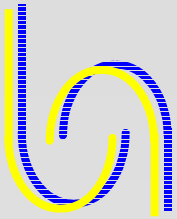
- Compute a transitive predicate
  - Self-join workaround used with SQL Server
  - SQL99 recursive queries with DB2
  - Omitted recursion depth with XSB





Web Services and the Semantic Web

- Web Services are quickly gaining momentum
- Previously unseen cross-vendor support and interoperability
- Problem: a priori knowledge of tModels necessary
- These developments are finally acknowledged in the SW community
- Projects
  - DAML-S
  - WSMF

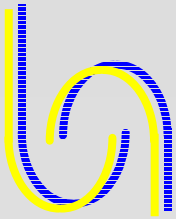


Web Service Description Framework

- WSDF takes a pragmatic approach
  - Bases on WSDL
  - Restricted to side effect-free services
  - Focus on the invocation

```
public String getStudentID(String fn, String ln, Date bd)
```

- Idea: semantically tag the parameters with concepts from an ontology
  - E.g. `fn` is not just a string but a first name

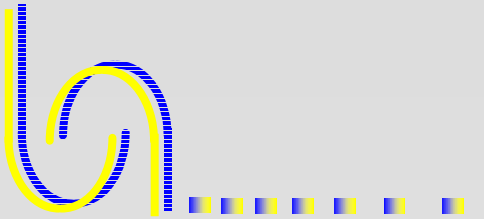


WSDL Rules

- Idea: use rules to
  - describe the prerequisites for calling a service
  - describe how the result should be interpreted

```
studentID(?S, ?I) <- hasMethod(?WSC, "getStudentID") and
                     returnValueOf(?I, ?WSC) and
                     hasParameter(?WSC, ?FN, ?LN, ?BD) and
                     firstName(?S, ?FN) and
                     lastName(?S, ?LN) and
                     birthday(?S, ?BD)

callable(?getStudentID) <- firstName(?S, ?FN) and
                           lastName(?S, ?LN) and
                           birthday(?S, ?BD)
```

- Allows generic logic-base WS clients!

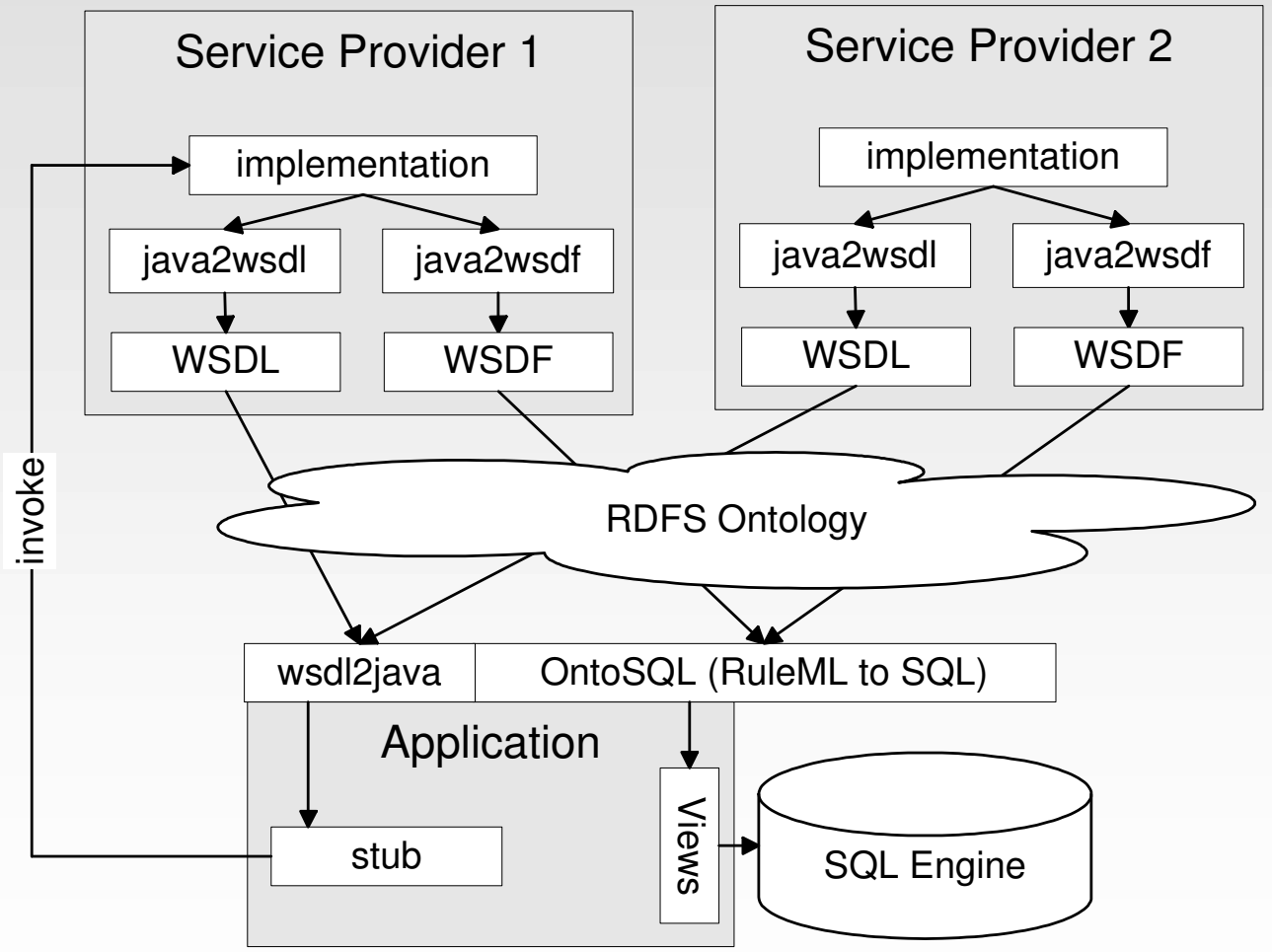


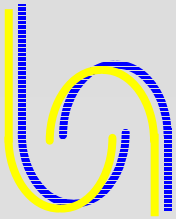
WSDF Tools

- prolog2ruleml
- java2wsdf
  - Generates WSDF descriptions in a web services like manner

```
/**
 * Returns the studentID of the student with a given
 * name, firstname, and birthday
 *
 *@param fn     The student's first name
 *              wsdf:ontotype="http://www.mit.edu/types#firstname"
 ...
 *@return       The list of courses takes by the student
 *              wsdf:ontotype="http://www.mit.edu/types#studentID"
 *@rules        callable(?getStudentID) <- firstName(?S, ?FN) and
 *                                         lastName(?S, ?LN) and
 *                                         birthday(?S, ? BD)
 ...
 */
public String studentID getStudentID(String fn, String ln, Date bd)
```

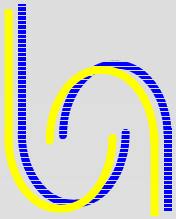
WSDF Scenario





OntoAgent

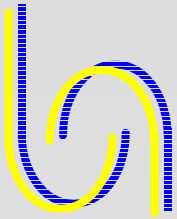
- Idea: declarative specification of an Agent / Workflow system using
  - RDF
  - RDF Schema
  - RuleML (ECA-Rules)



Declarative Agent Specification

- Idea presented in Boley et al‘s SWWS paper
- An agent can be specified entirely using markup

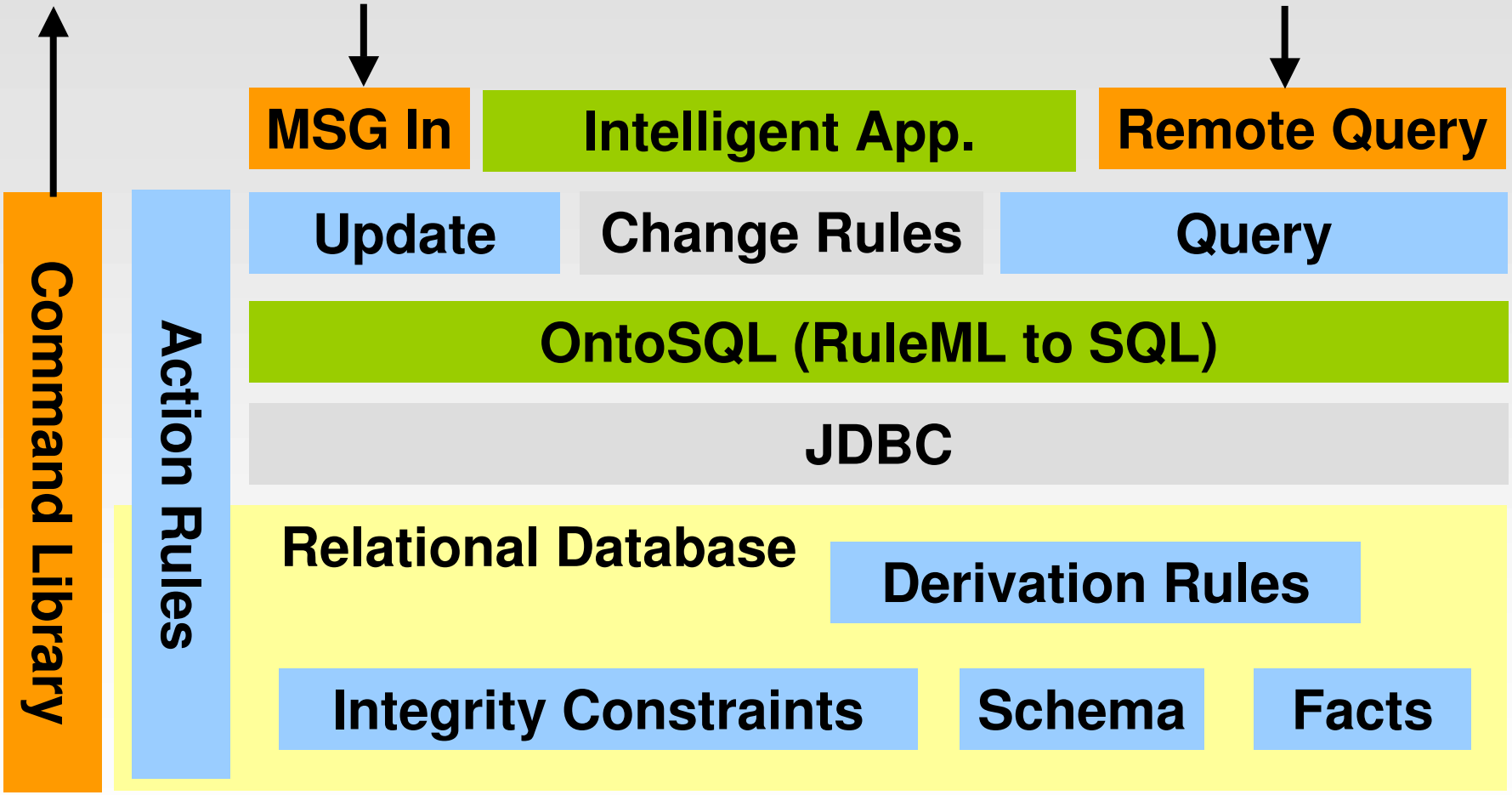
RDF	Mental state
RDF Schema	Schema (classes properties) for the mental state
RuleML	Integrity constraints
	Derivation rules
	Reaction rules



The OntoAgent Framework

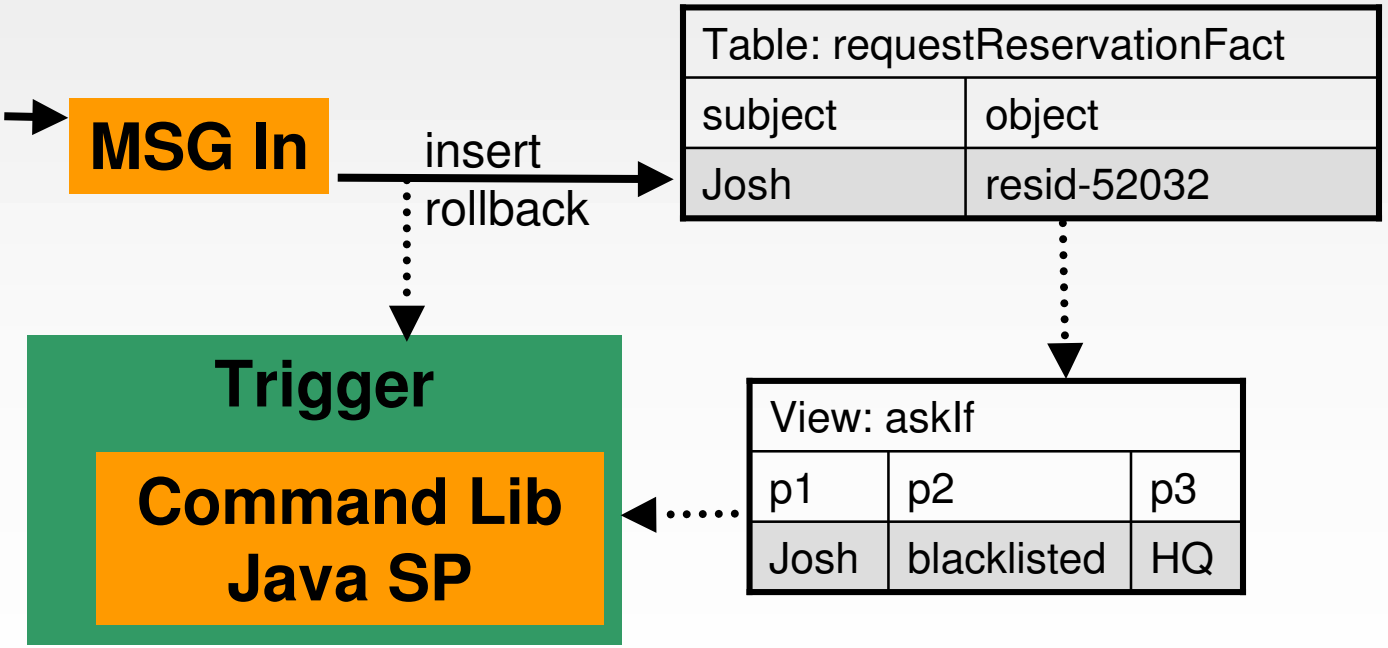
- Declarative specification is a promising approach
  - Helps the adoption of agent technology
  - So far a large number of incompatible systems
- Need a runtime execution framework
  - This is OntoAgent
- OntoAgent bases on Java and relational databases
  - We use OntoSQL to create views and tables
  - Java classes handle reaction rules and message IO

OntoAgent Architecture



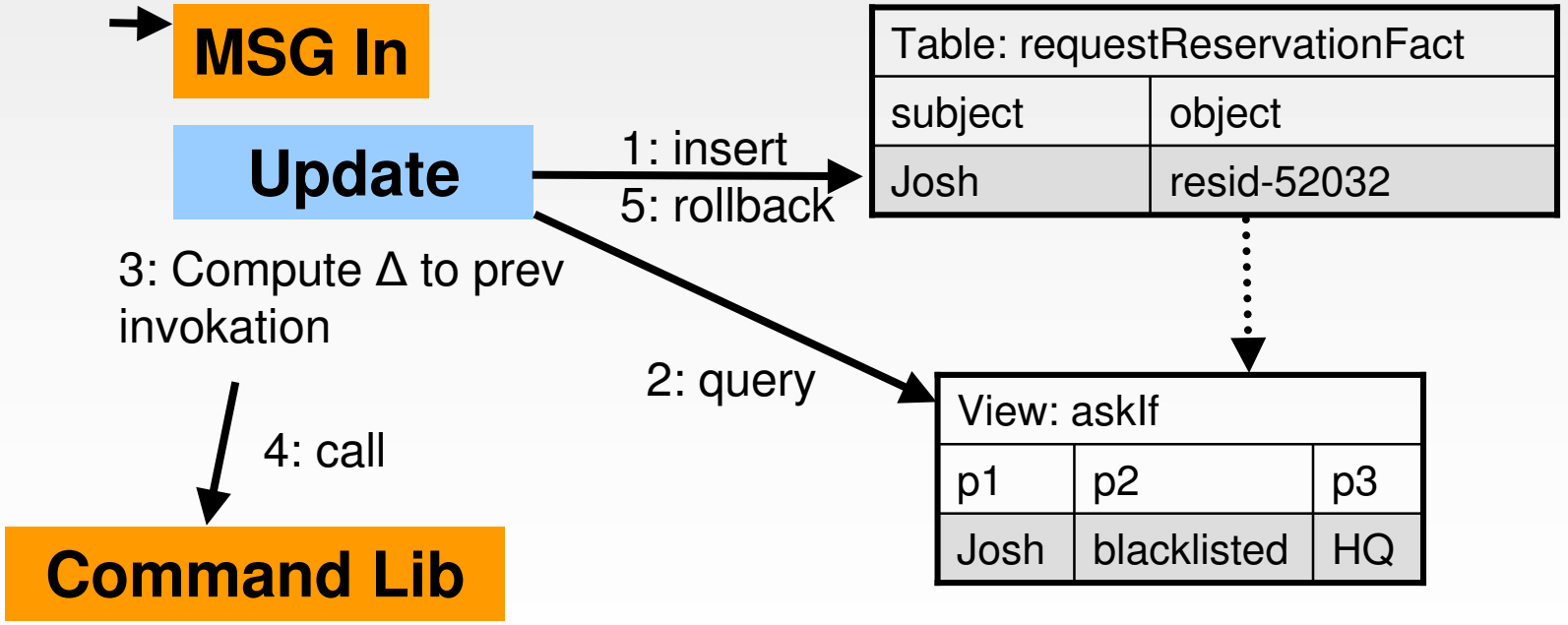
Possible Reaction Rule Implementation

- Incoming messages are treated like regular facts which are valid temporarily
- Actions are treated like derived predicates
- Triggers check action predicates upon updates to the message fact tables



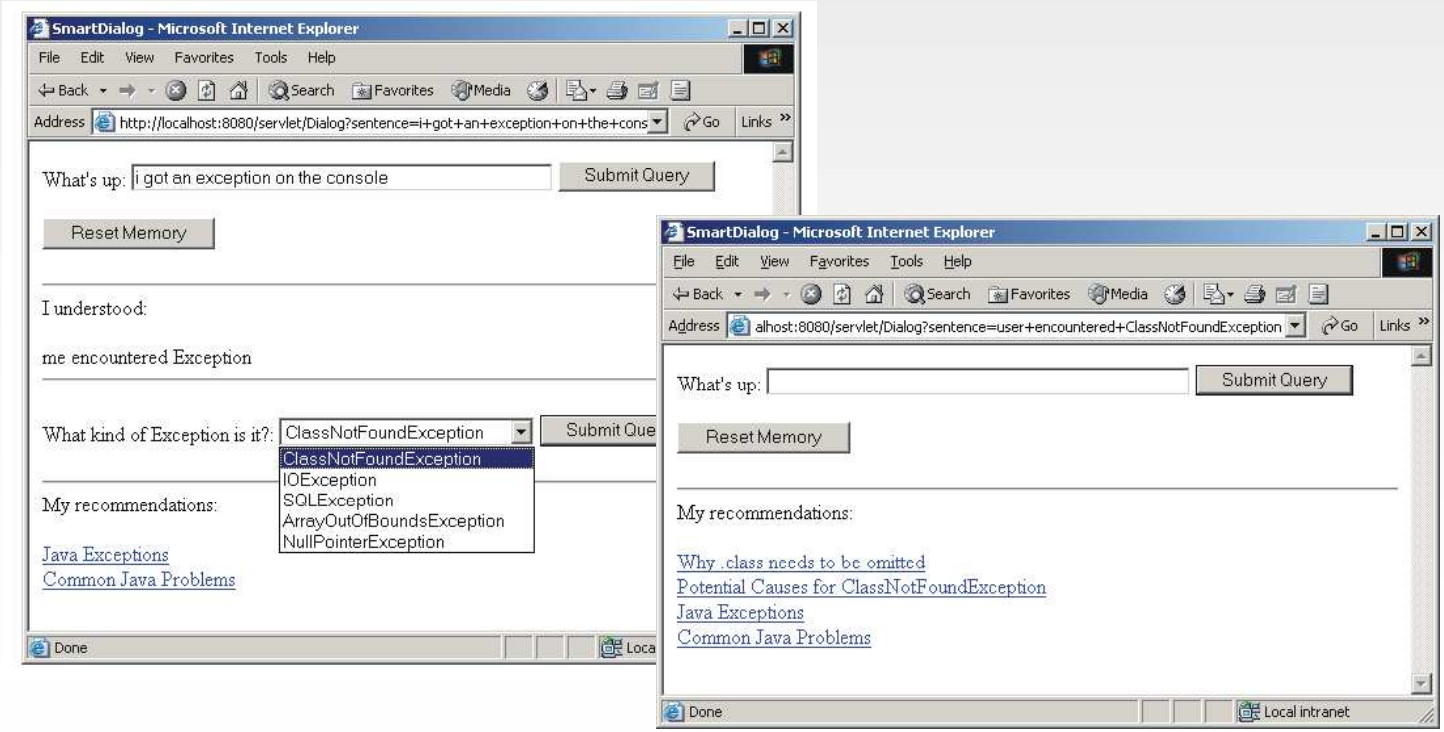
OntoAgent ECA Rule Implementation

- Upon updates, all action views are checked
- Agent maintains a state of previous actions
  - Only perform new actions w.r.t. old state
  - Makes sure action is caused by insert



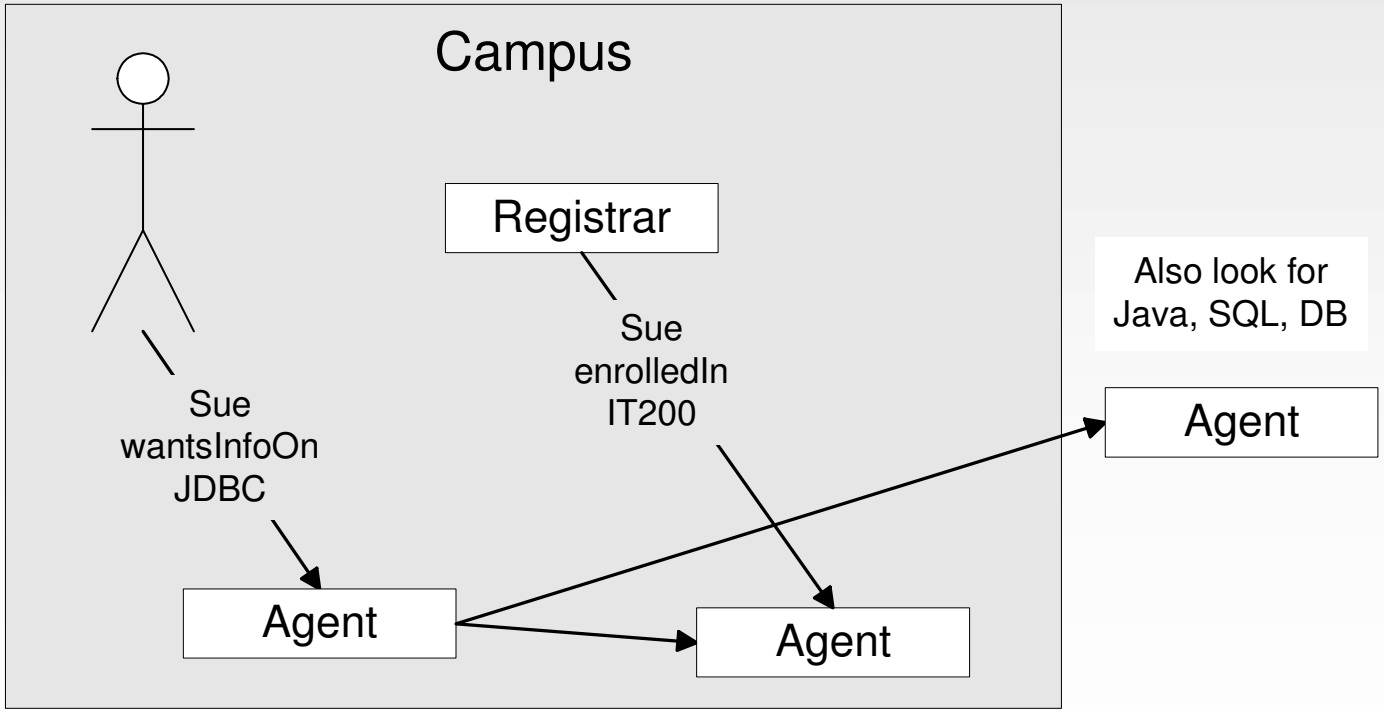
Sample Application: SmartGuide

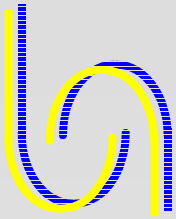
- Take a different appraoch to doument retrieval
  - Collect facts on the user, context, and situation
  - Reason using domain rules
  - Recommend docs / ask clarification questions



Personal Information Agents

- Idea: everyone has a personal information retrieval agent + personal bookshelf
- Leverage different contexts





Outlook

- We see the semantic web as a key enabler for a new generation of applications
  - Ease of development
  - Metadata managed in a distributed way
- Keys are:
  - Adoption
  - Scalability (see NEC Reasearch Index)
  - Lots of data is more important than sophisticated functionality