s a n d p i p e r   s o f t w a r e

MDA & Semantic Web Services
An ODM Tutorial & SWS Case Study

Elisa Kendall
Sandpiper Software

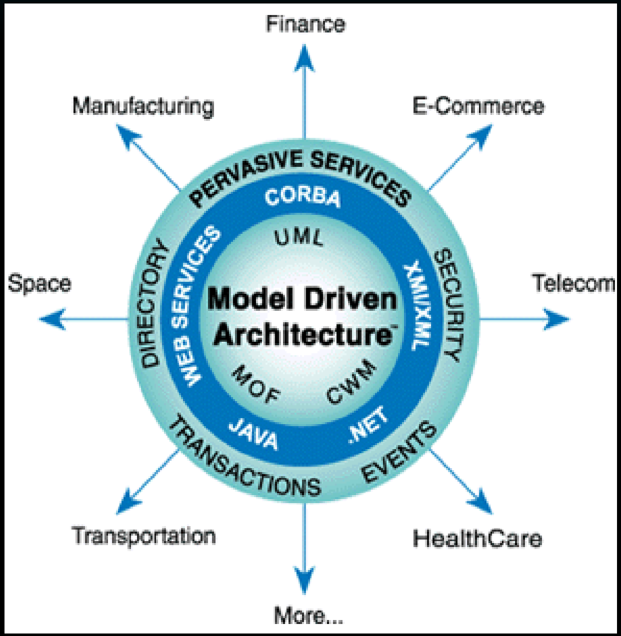
May 24, 2006

Agenda

- Brief Review of OMG MDA
- Semantics & MDA – Complementary Technologies
- The Ontology Definition Metamodel (ODM)
  - What it is
  - A quick walk through the RDF & OWL metamodels
  - Developing ontologies in UML – highlights from the UML profile for RDF & OWL
  - Status & Relationship to other OMG, W3C, ISO standards
- Semantics for Web Services
  - What they provide
  - Overview of OWL-S
  - Overview of the Semantic Web Services Framework (SWSF)
  - Status of standards
- Implementation Strategies
  - Semantic Service Oriented Architecture (SSOA) – work in progress
  - InferenceWeb – semantics supporting registration, explanations & trust for semantically-enabled services

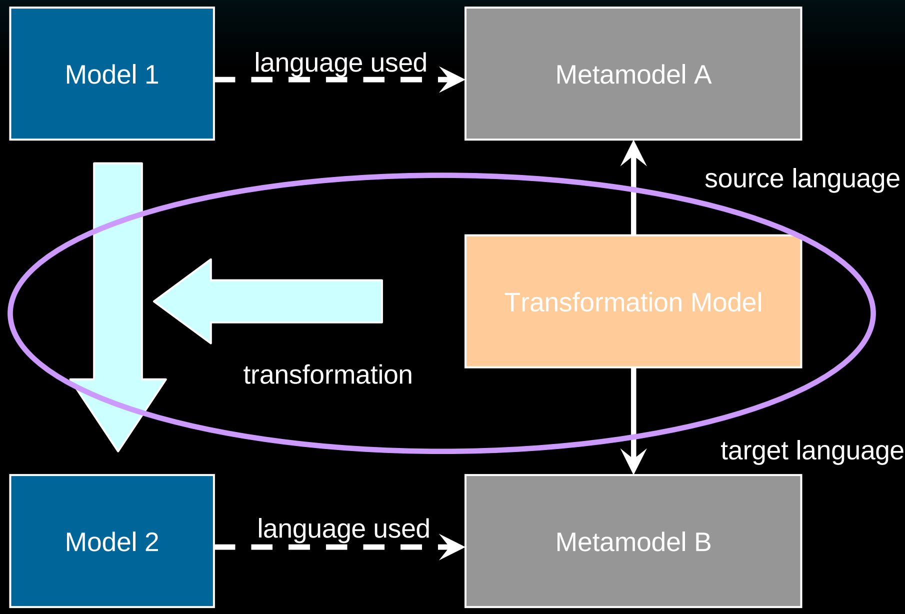
Model Driven Architecture® (MDA®)

- Insulates business applications from technology evolution, for
  - Increased portability and platform independence
  - Cross-platform interoperability
  - Domain-relevant specificity
- Consists of standards and best practices across a range of software engineering disciplines
  - The Unified Modeling Language (UML®)
  - The Meta-Object Facility (MOF™)
  - The Common Warehouse Metamodel (CWM™)
- MOF defines the metadata architecture for MDA
  - Database schema, UML and ER models, business and manufacturing process models, business rules, API definitions, configuration and deployment descriptors, etc.
  - Supports automation of physical management and integration of enterprise metadata
  - MOF models of metadata are called *metamodels*

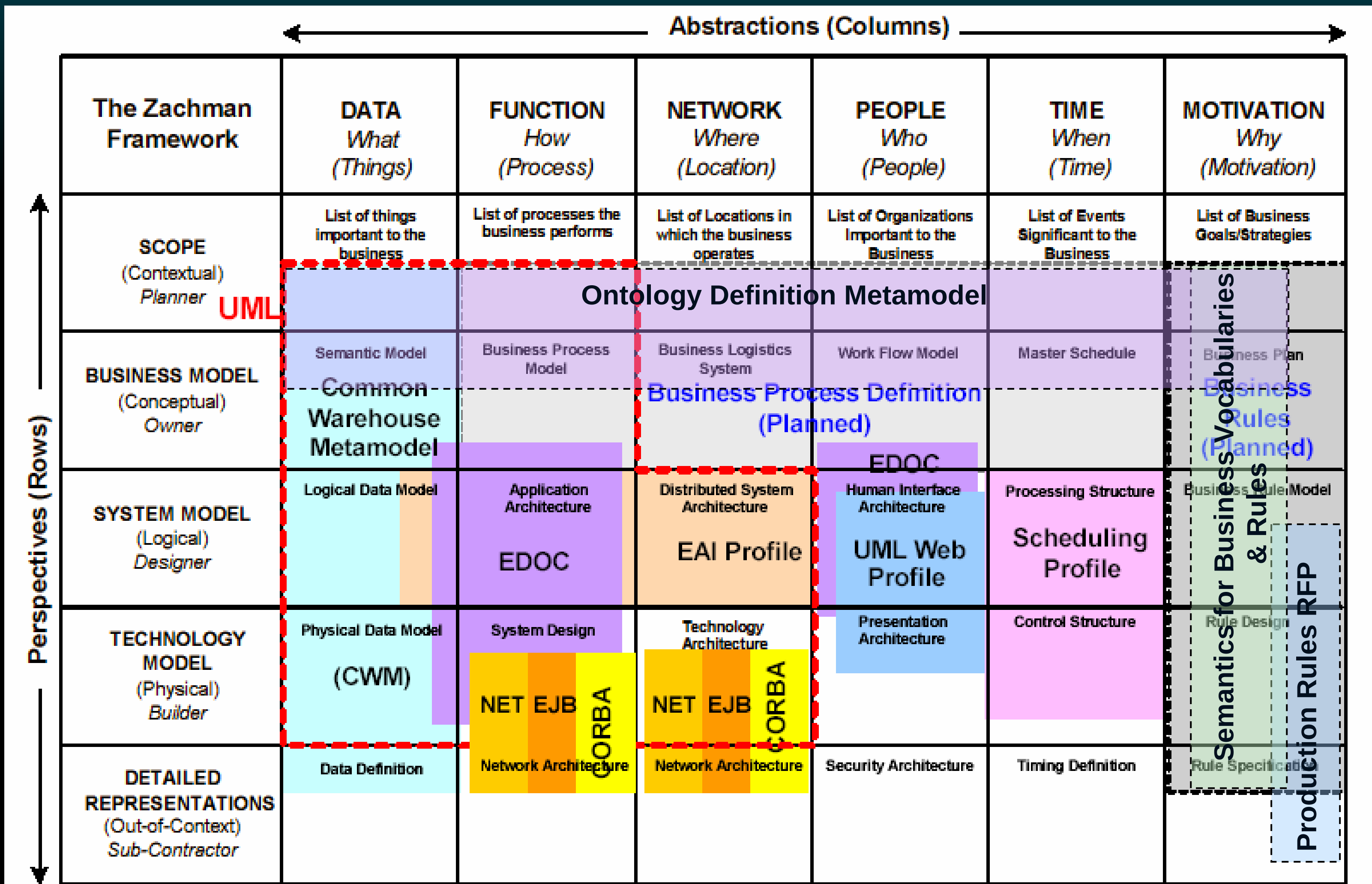


MOF-Based Metadata Management

- MOF tools use metamodels to generate code that manages metadata, as XML documents, CORBA objects, Java objects
- Generated code includes access mechanisms, APIs to
  - Read and manipulate
  - Serialize/transform
  - Abstract the details based on access patterns
- Related standards:
  - XML Metadata Interchange (XMI®)
  - CORBA Metadata Interface (CMI)
  - Java Metadata Interface (JMI)
- Metamodels are defined for
  - Relational and hierarchical database modeling
  - Online analytical processing (OLAP)
  - Business process definition, business rules specification
  - XML, UML, and CORBA IDL



OMG Standards & Zachman Framework



MDA from the KR Perspective

- EII solutions rely on strict adherence to agreements based on common information models that take weeks or months to build
- Modifications to the interchange agreements are costly and time consuming
- Today, the analysis and reasoning required to align multiple parties' information models has to be done by people
- Machines display only *syntactic* information models and informal text describing the semantics of the models
- Without formal *semantics*, machines cannot aid the alignment process
- Translations from each party's syntactic format to the agreed-upon common format have to be hand-coded by programmers
- MOF® and MDA® provide the basis for automating the syntactic transformations

MOF and KR Together

- MOF technology streamlines the *mechanics* of managing models as XML documents, Java objects, CORBA objects
- Knowledge Representation supports *reasoning* about resources
  - Supports semantic alignment among differing vocabularies and nomenclatures
  - Enables consistency checking and model validation, business rule analysis
  - Allows us to ask questions over multiple resources that we could not answer previously
  - Enables policy-driven applications to leverage existing knowledge and policies to solve business problems
    - Detect inconsistent financial transactions
    - Support business policy enforcement
    - Facilitate next generation network management and security applications

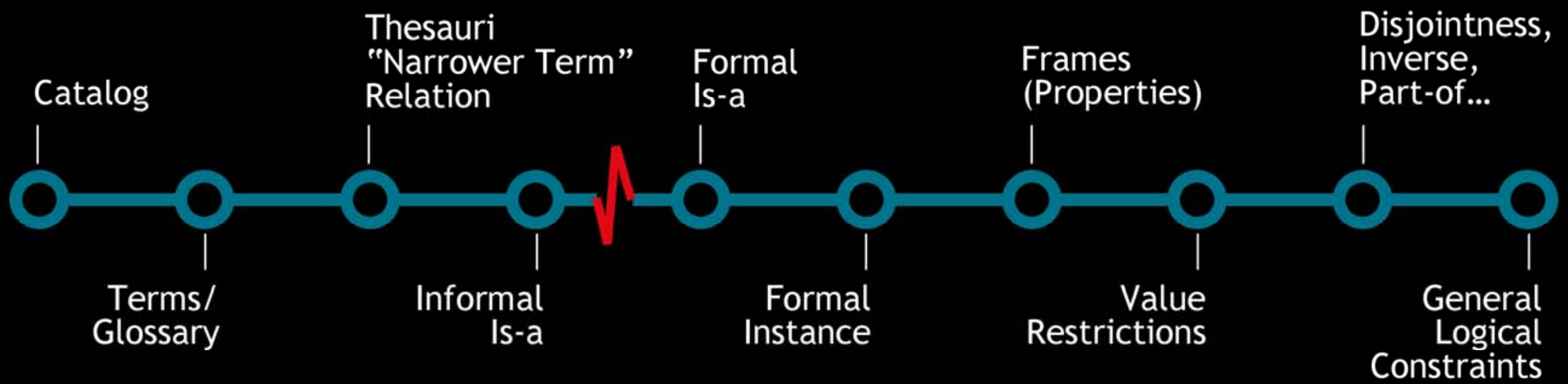
  while integrating with existing RDBMS and OLAP data stores
- MOF provides no help with reasoning
- KR is not focused on the mechanics of managing models or metadata
- Complementary technologies – despite some overlap

Level Setting

An ontology specifies a rich description of the

- **Terminology, concepts, nomenclature**
- **Properties explicitly defining concepts**
- **Relations among concepts (hierarchical and lattice)**
- **Rules distinguishing concepts, refining definitions and relations (constraints, restrictions, regular expressions)**

relevant to a particular domain or area of interest.

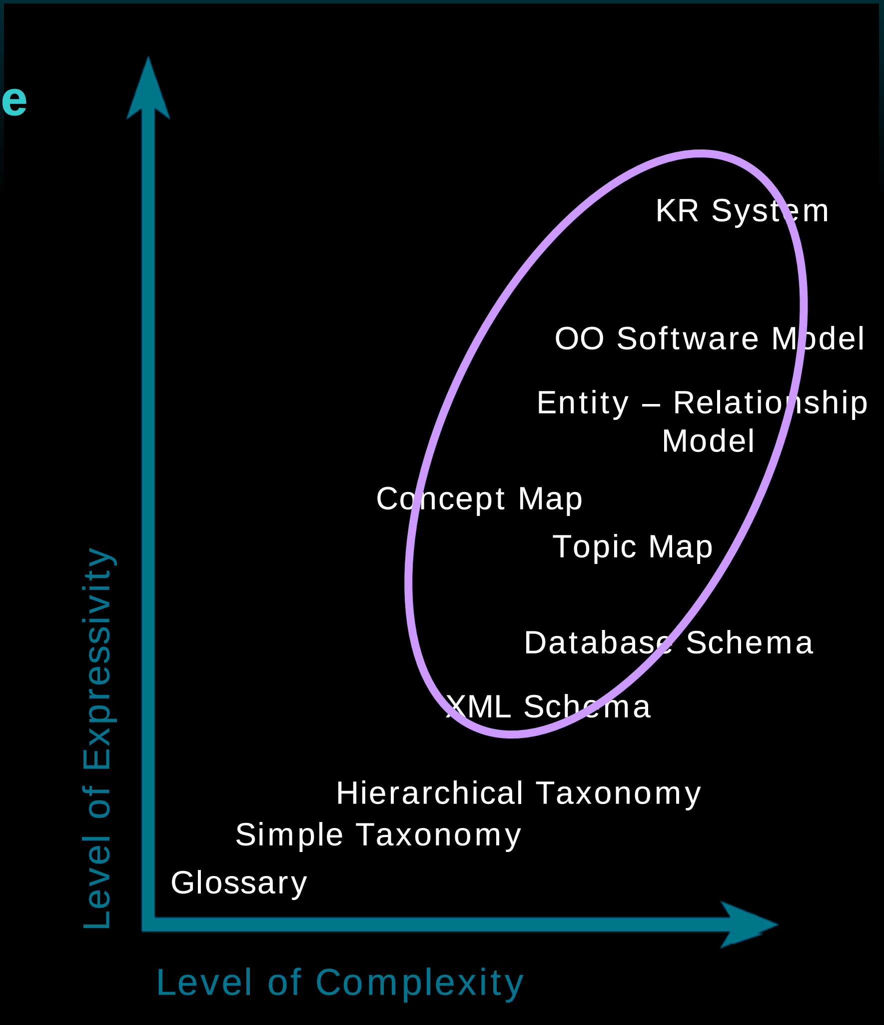


*Based On Aaai '99 Ontologies Panel - Mcguinness, Welty, Ushold, Gruninger, Lehmann

Classifying Ontologies

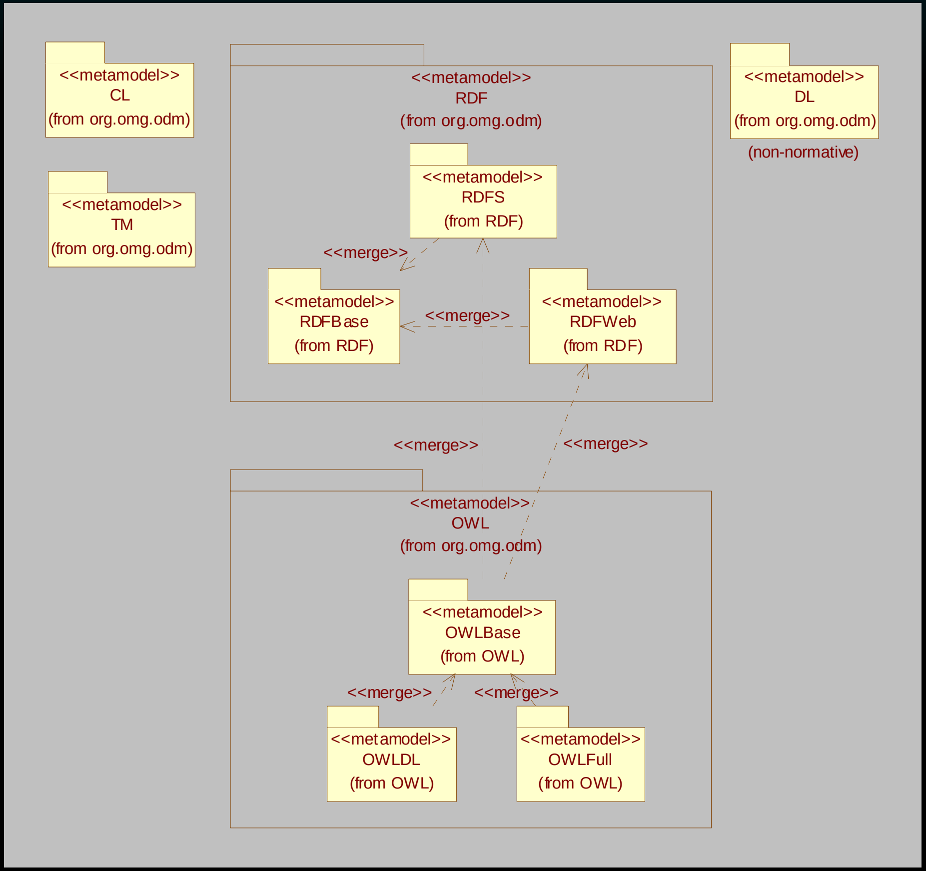
Classification techniques are as diverse as conceptual models; and generally include understanding

- Methodology
- Target Usage
- Level of Expressivity
- Level of Complexity
- Reliability / Level of Authoritativeness
- Relevance
- Amount of Automation
- Metrics Captured and/or Available



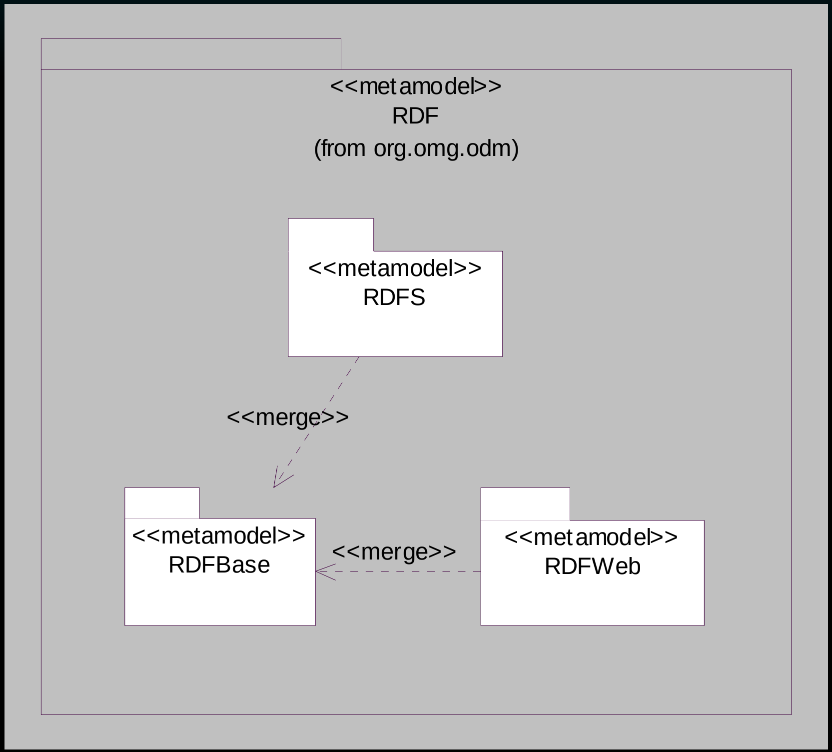
Towards a Model Driven Semantic Web – ODM

- Five EMOF platform independent metamodels (PIMs), four normative
- Mappings (MOF QVT)
- UML2 Profiles
  - RDFS & OWL
  - TM
- Collateral
  - XMI
  - Java APIs
  - Proof-of-concepts
- Conformance
  - RDFS & OWL
  - Multiple Options
  - TM, CL Optional
  - Informative Mappings

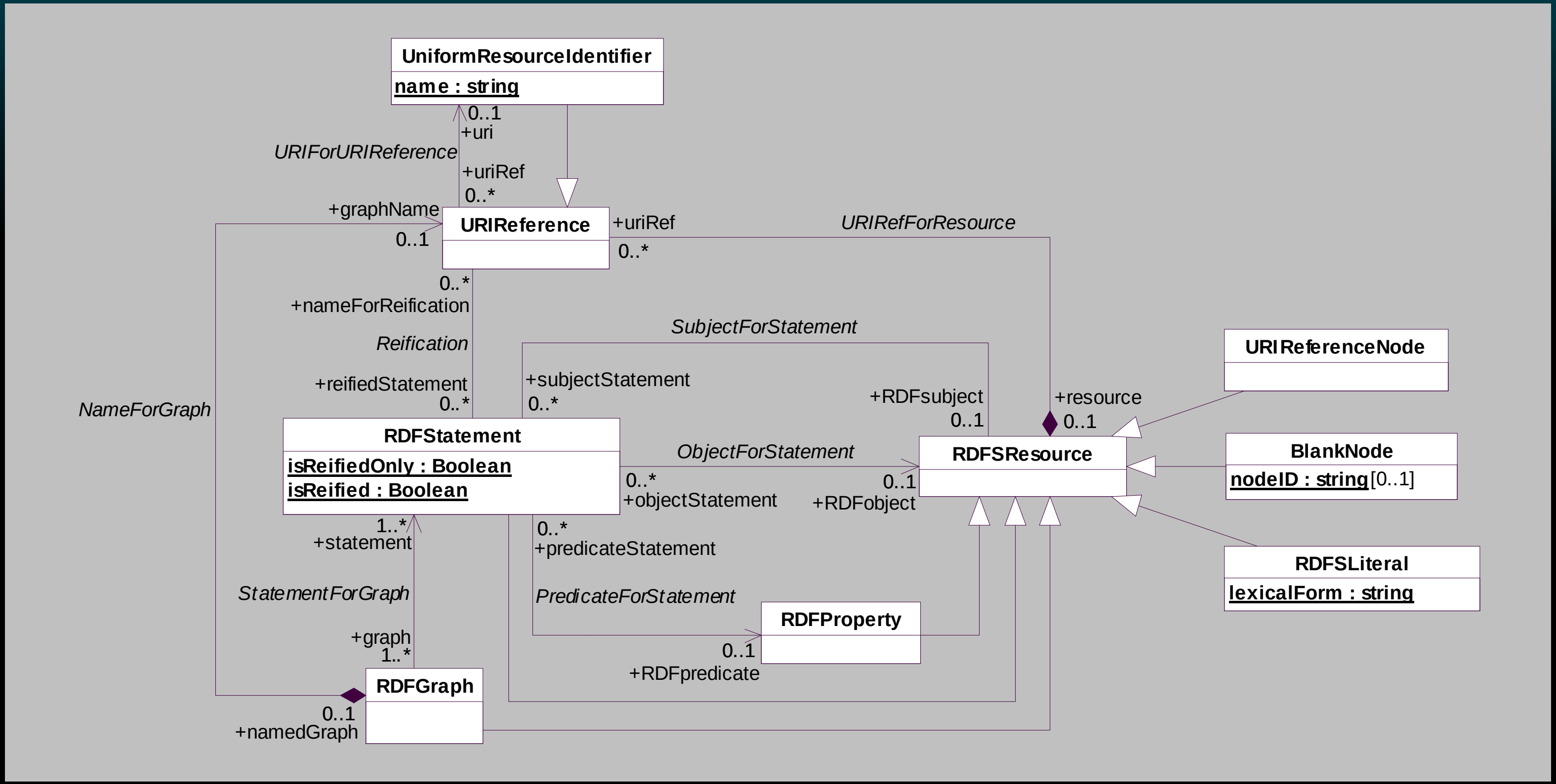


Resource Description Framework (RDF) Metamodel Overview

- RDFBase – primary package
  - Reflects basic abstract syntax from RDF Concepts
  - Minimal implementation requirements, e.g., for RDF triple/quad store
- RDFS – adds vocabulary related to RDF Schema, few additional RDF features
- RDFWeb – fits the model to the Web via document model, required for RDF/XML syntax, among others



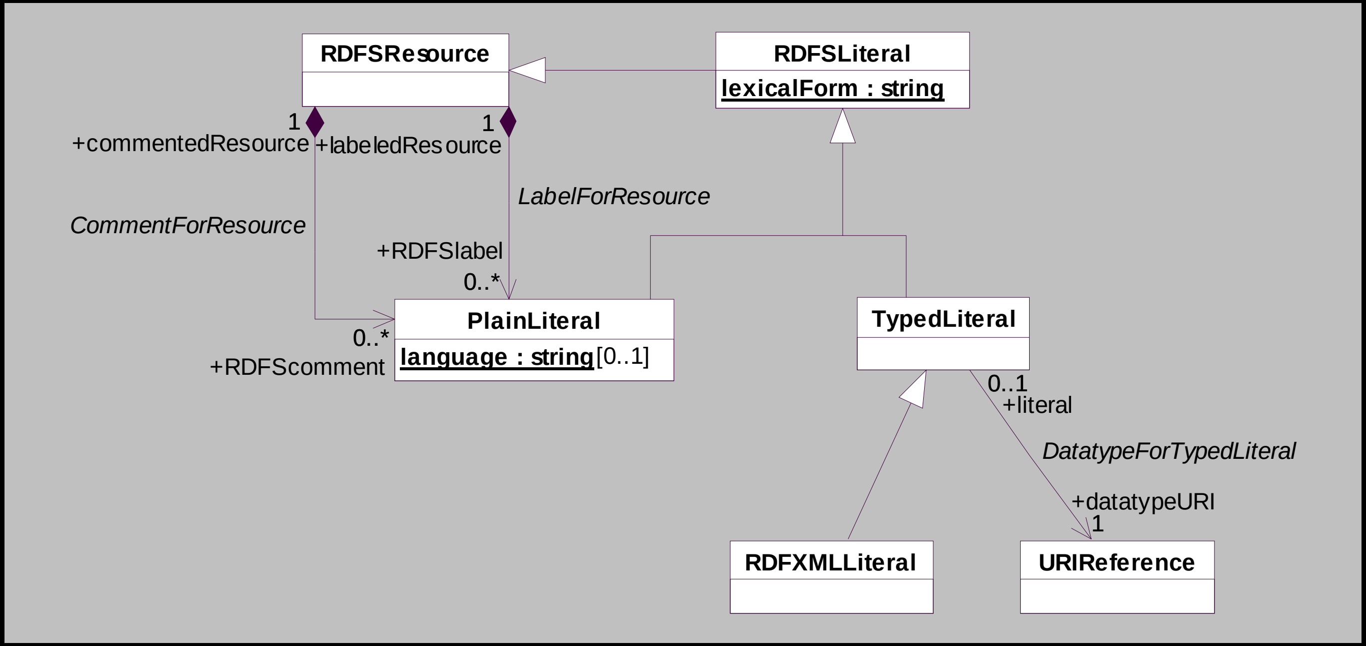
RDFBase Package – Statements



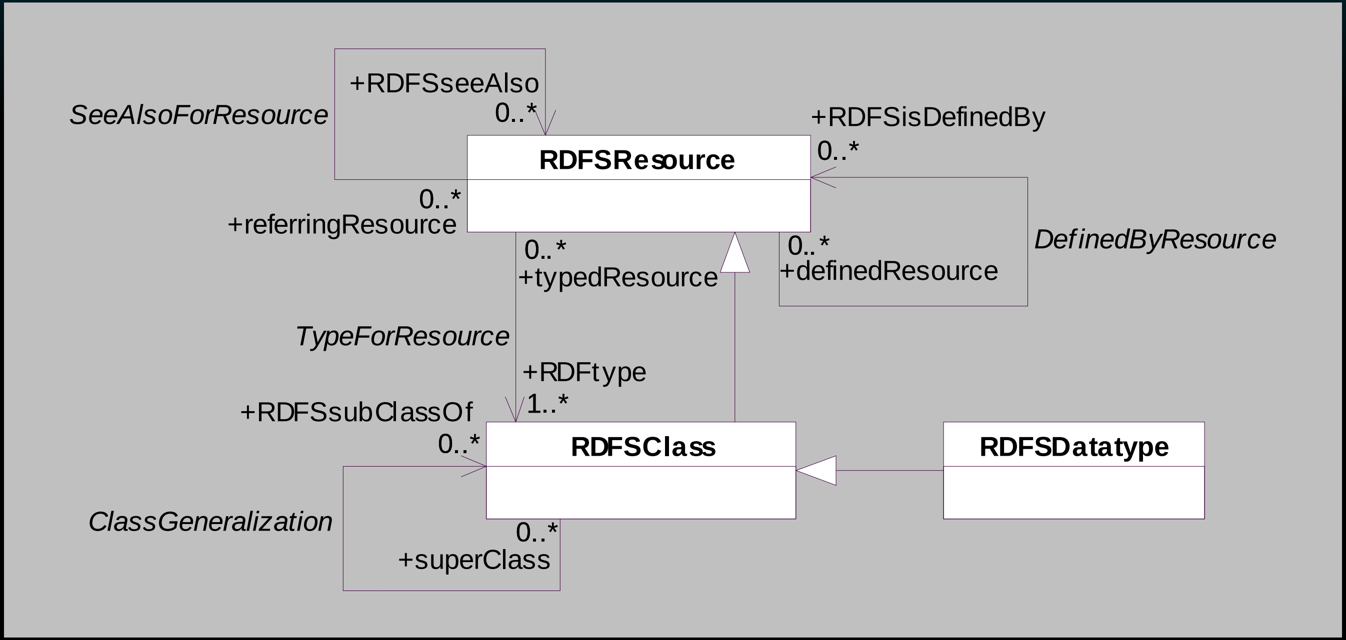
- Supports named graphs (*e.g.*, per SPARQL), reification, blank node identifiers, essentially RDF basics
- Limited coverage to RDF Concepts document rather than along namespace boundaries, which didn't work from a UML perspective
- Promotion of the blank node identifier to RDFSResource addresses MOF multiple classification, non-normative work-around

RDFBase Package - Literals

- Remaining support for RDF basics
- Note that for this package / model, TypedLiteral has a property that points to its datatype URI through URIReference (distinct from OWL)

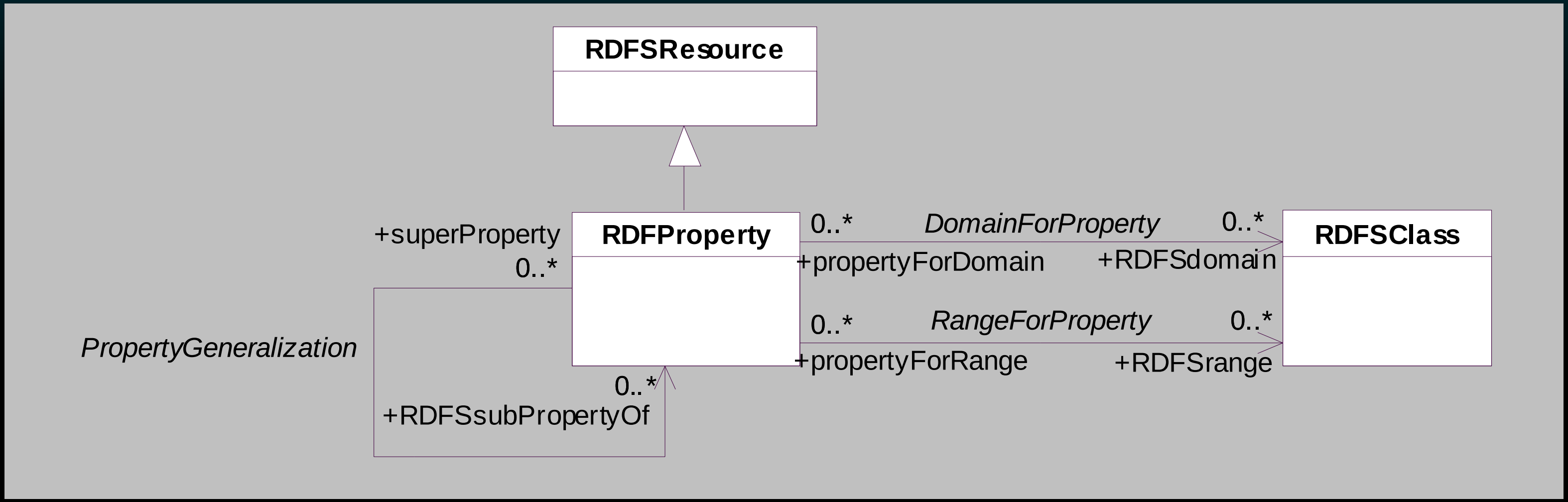


RDFS Package – Classes & Utilities



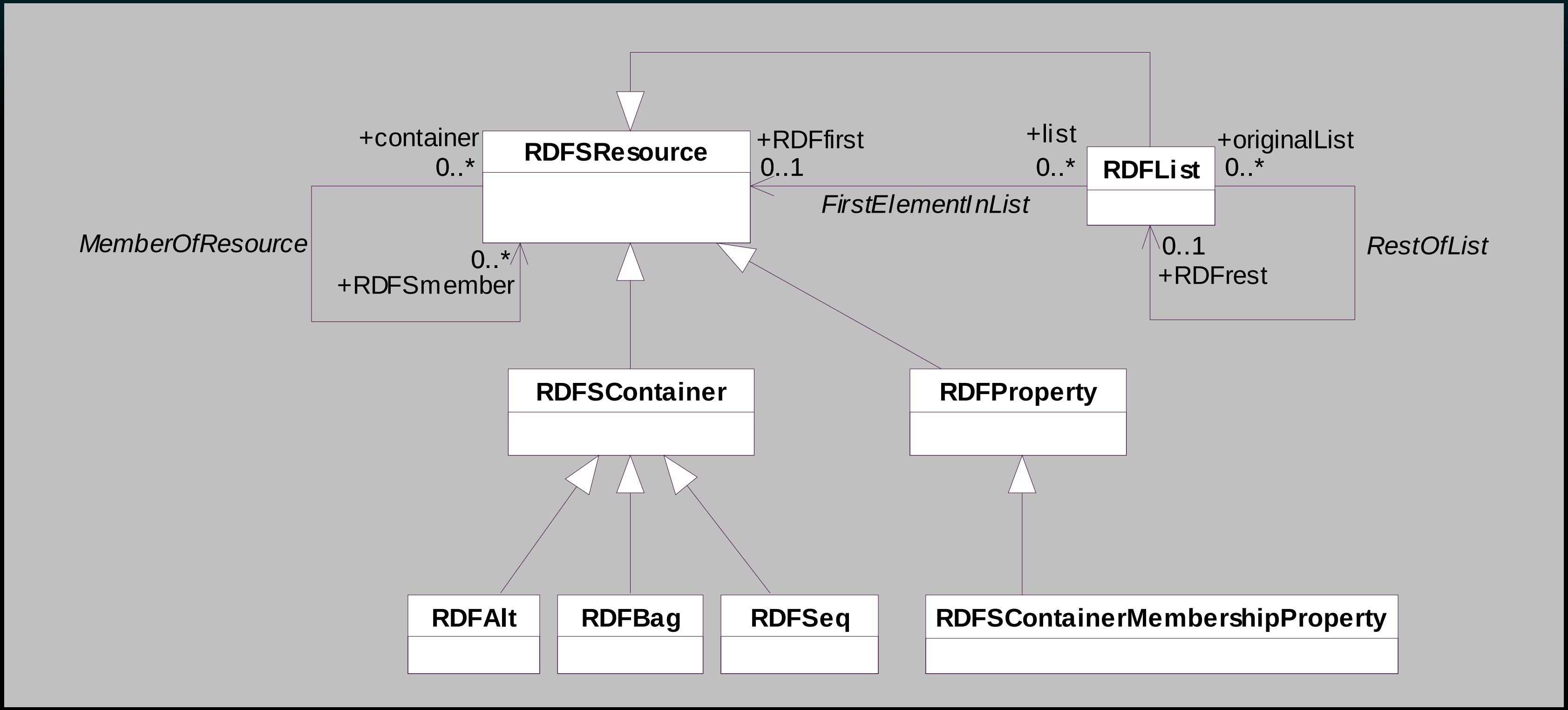
- **RDFS assists us in "getting around" MOF multiple classification limitations through rdf:type**

RDFS Package - Properties

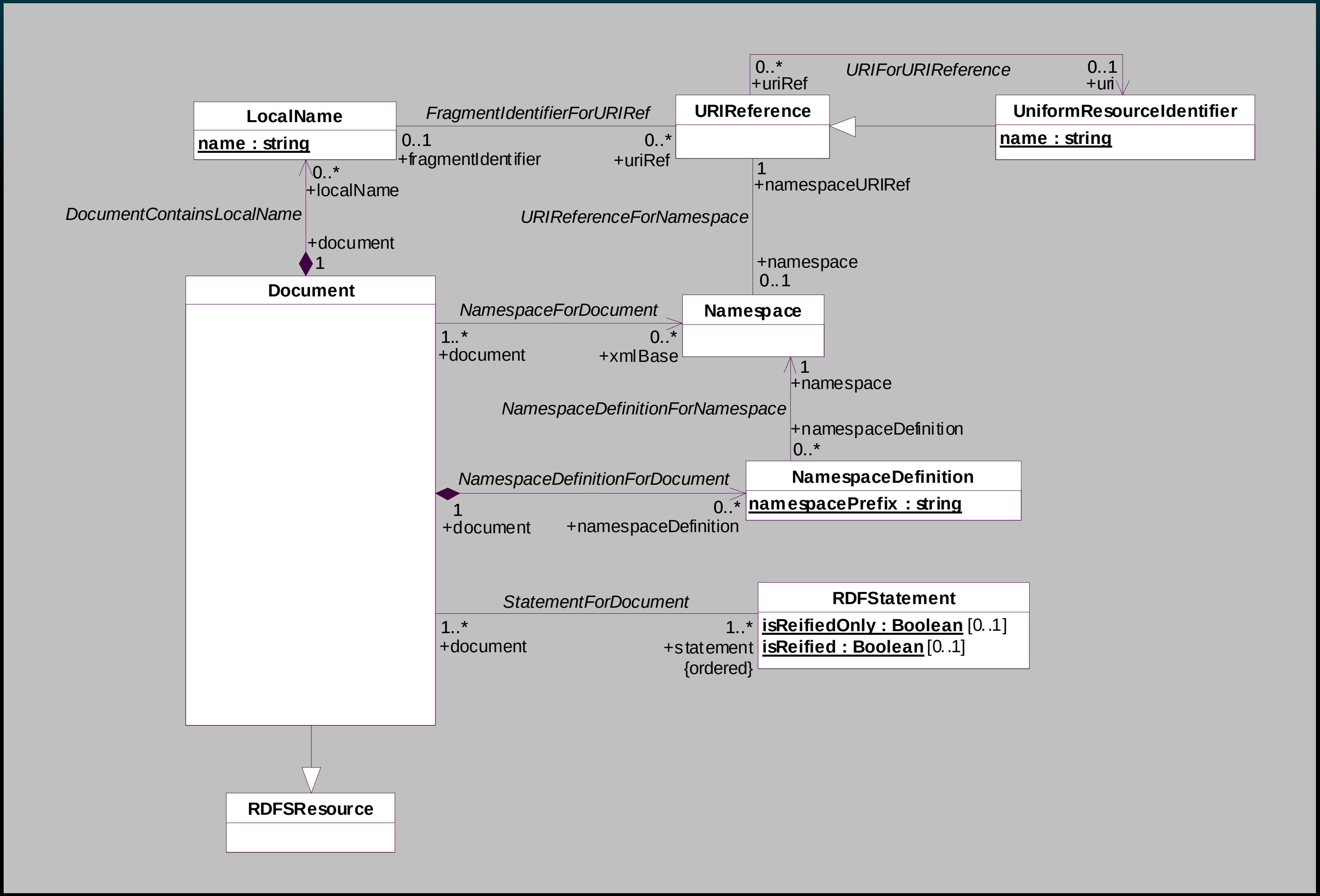


- **Note that rdf:domain and rdf:range are global properties – limiting their usage enhances reusability of ontology components**

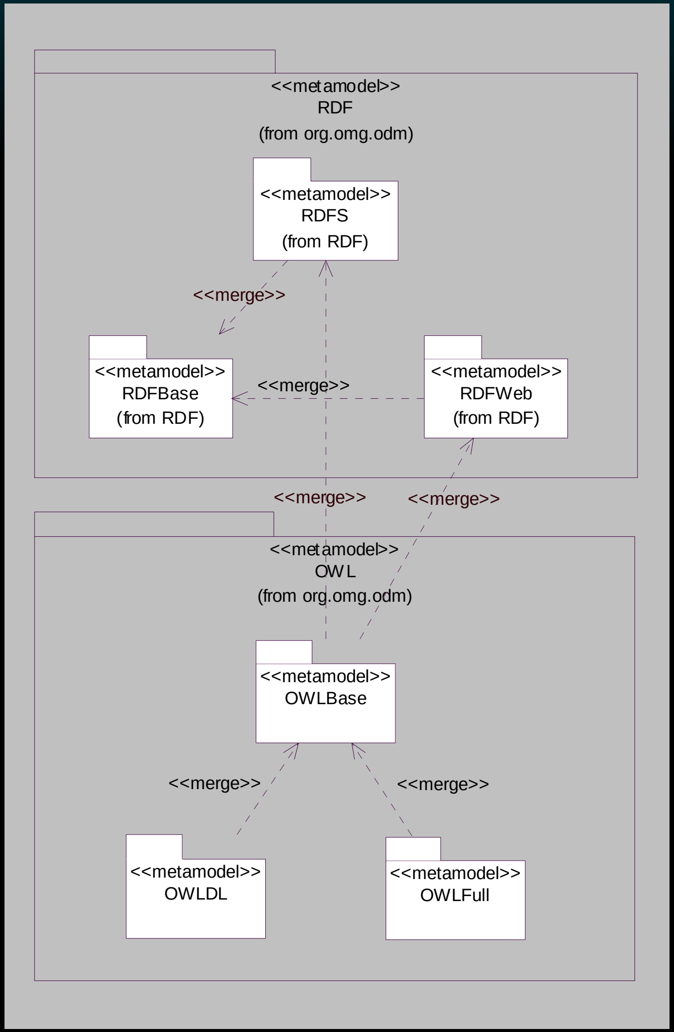
RDFS Package – Containers & Collections



RDFWeb Package - Documents

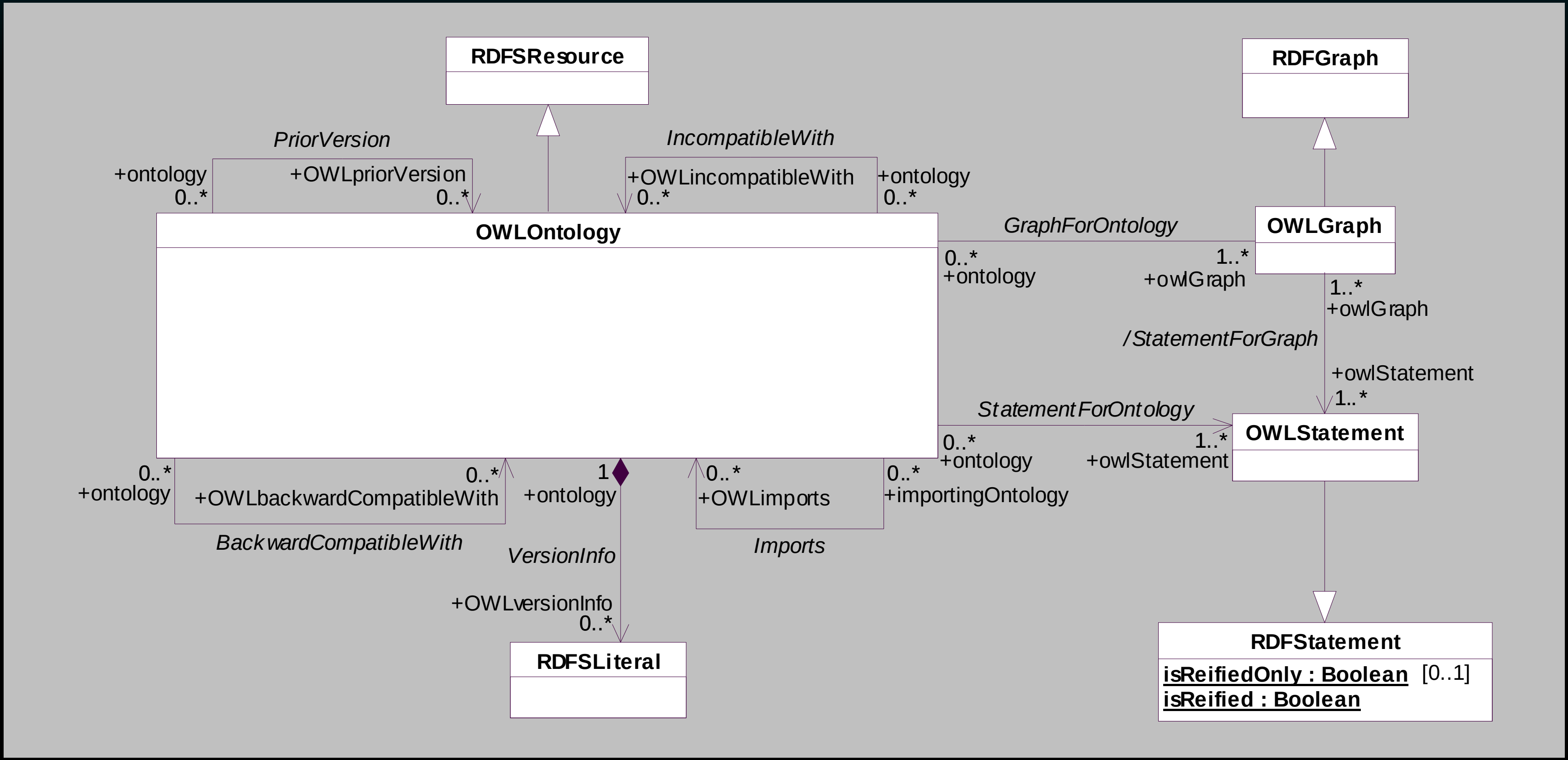


Web Ontology Language (OWL) Metamodel Overview

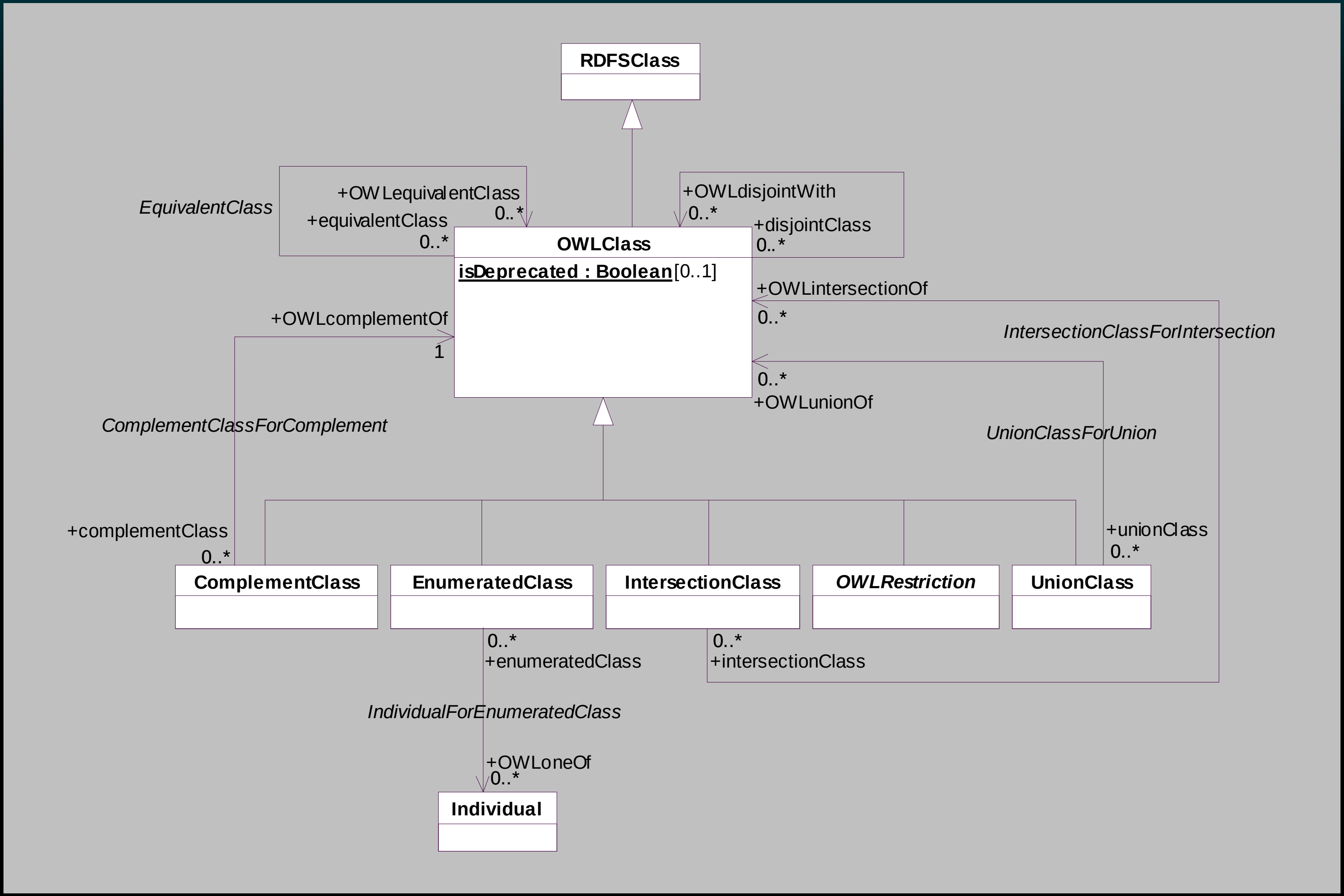


- OWL metamodel components include:
  - OWLBase, covering all common abstract syntax & constraints
  - OWLDL – containing OWL DL constraints
  - OWLFull – containing OWL Full constraints
- Non-normative models for OWL, including changes to property representation & intersection classes for OWL Full, to address MOF multiple classification, are posted to the OMG web site

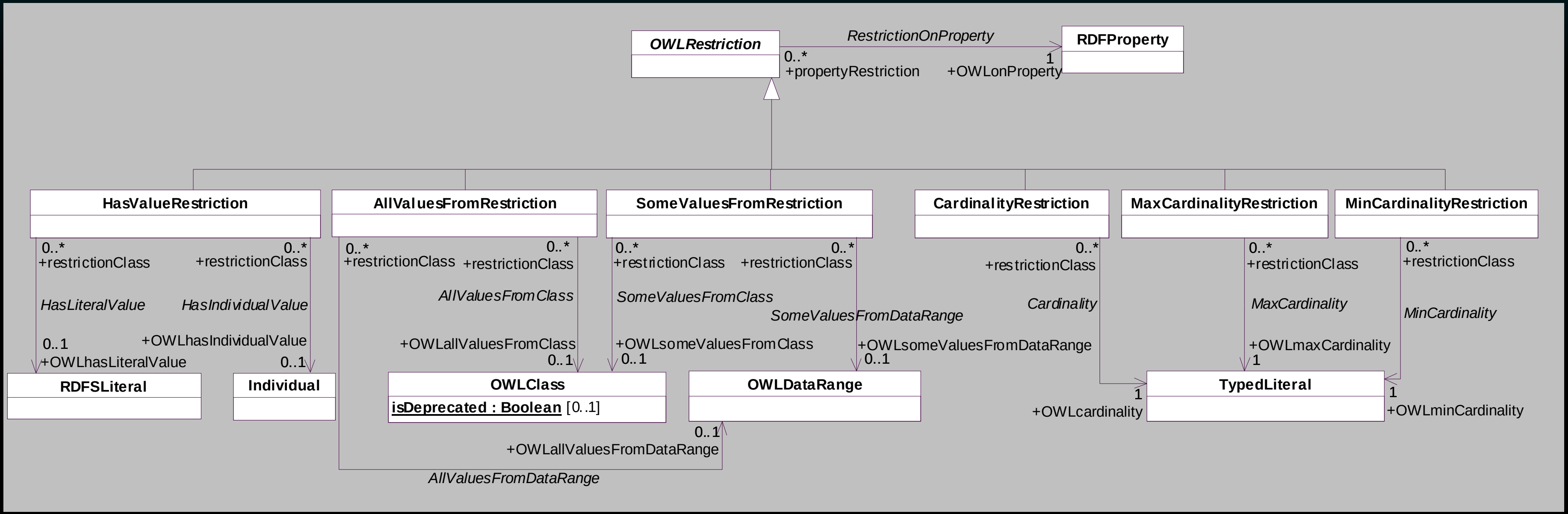
OWLBase Package – OWL Ontology



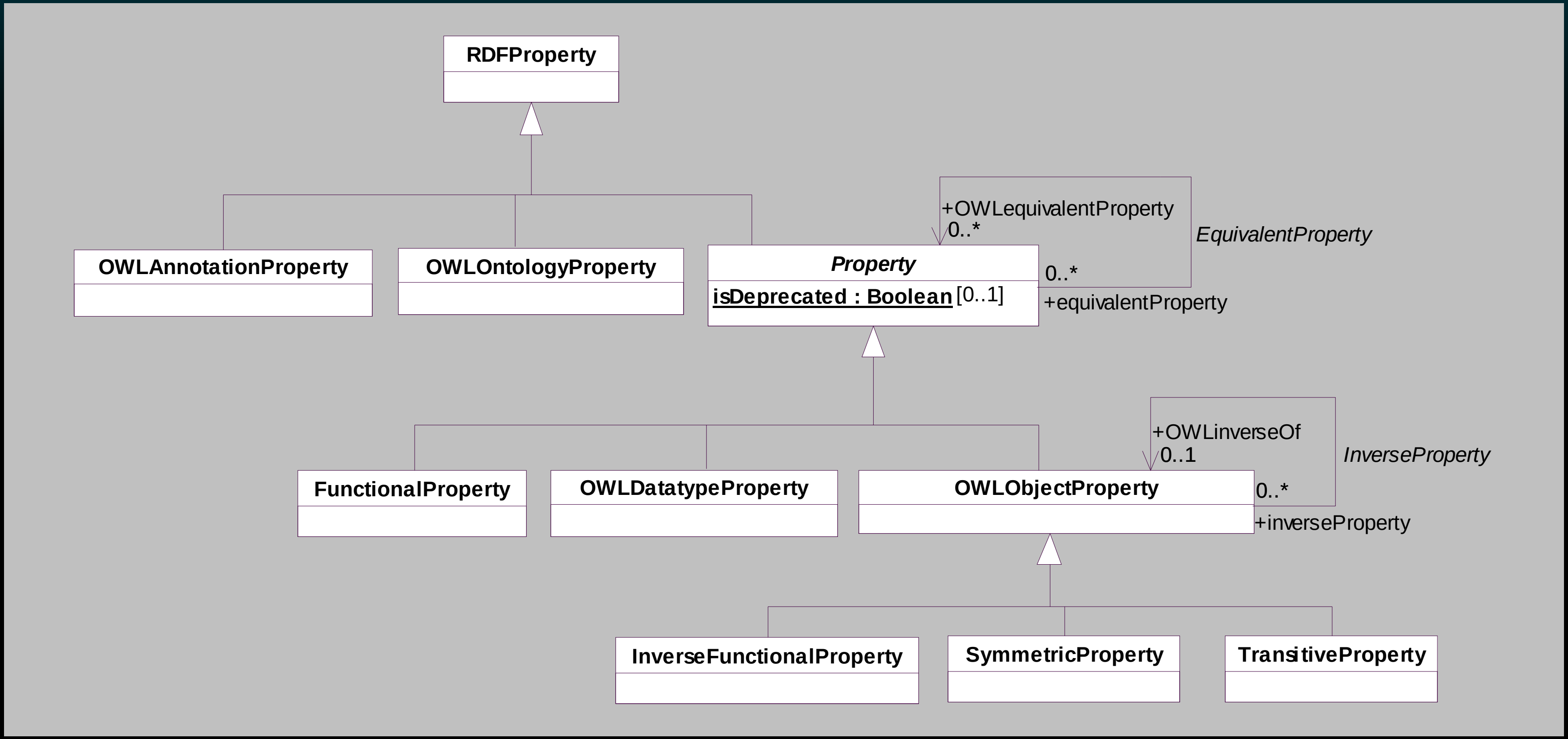
OWLBase Package – OWL Classes



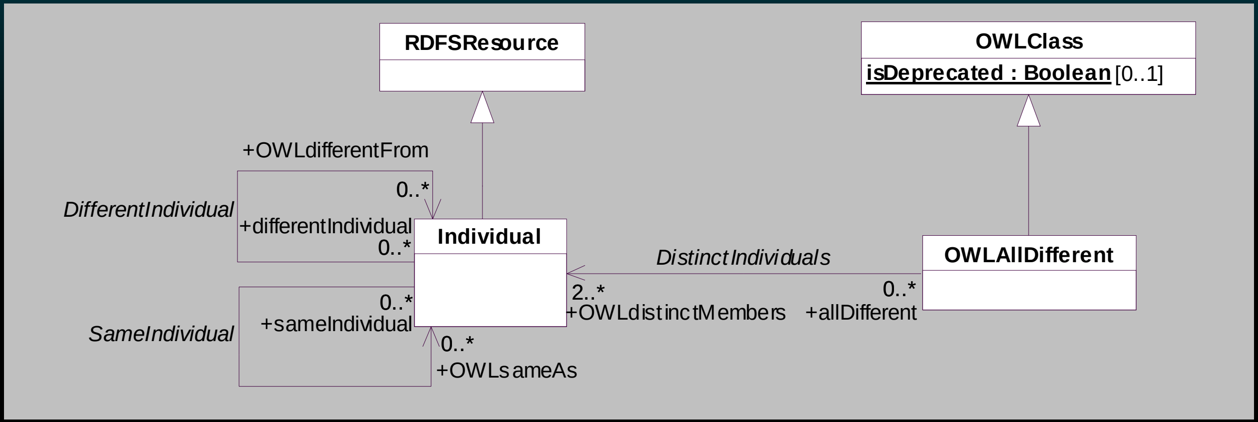
OWLBase Package – Restrictions

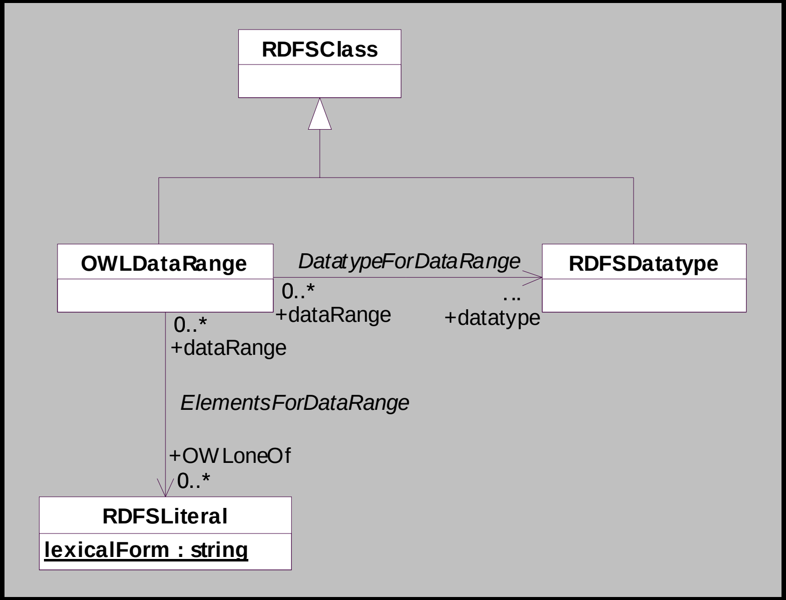


OWLBase Package – OWL Properties

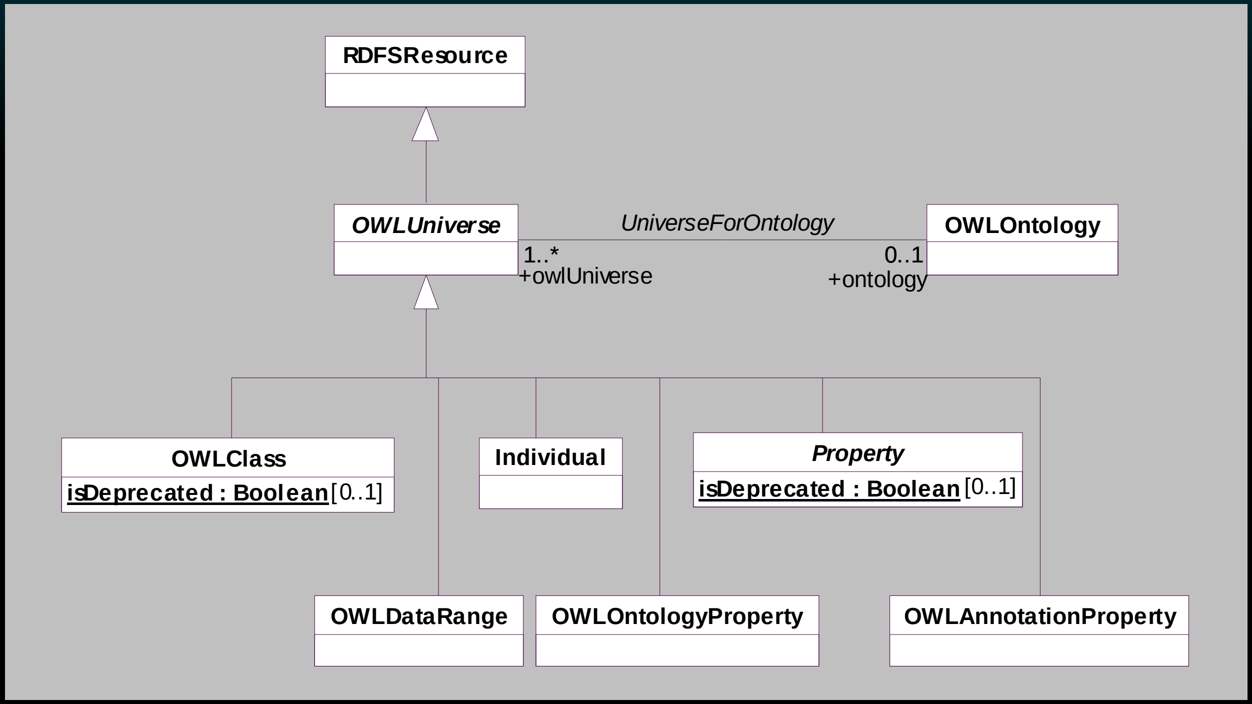


Individuals & Datatypes





OWL Universe

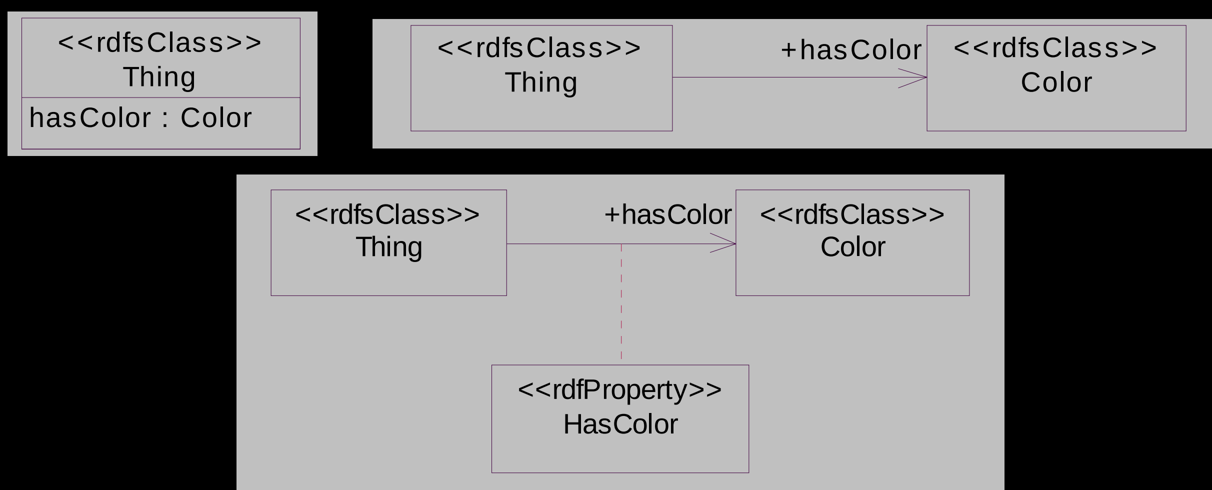


The UML Profile for RDF & OWL

- Intended to be highly intuitive for UML users
- Reuses UML constructs when they have the same semantics as OWL
- When this is not possible, stereotypes UML constructs that are consistent and as close as possible to OWL semantics
- Uses standard UML 2 notation
- In the few cases where this is not possible, follows the clarifications and elaborations of stereotype notation defined in UML 2.1
- Leverages the model library included in Appendix A for a number of constructs, for example statements, rdf:value, container and list elements, as well as built-in properties

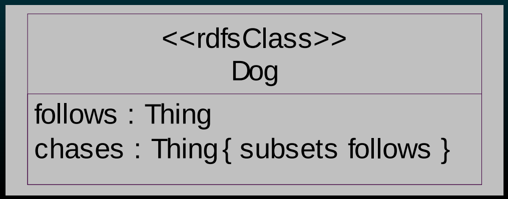
Key Features of the RDF Profile

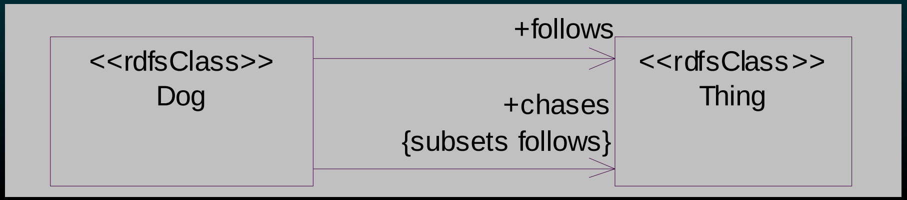
- **rdfs:Resource is modeled as UML::InstanceSpecification**
- **Introduction of <<reifies>> stereotype of UML::Dependency to allow such instance specifications to reify classes, properties, individuals, statements, etc.**
- **rdf:Property is modeled as UML::AssociationClass *and* UML::Property, to provide greatest possible flexibility**
- **Several possible representations of various aspects of rdf:Property:**

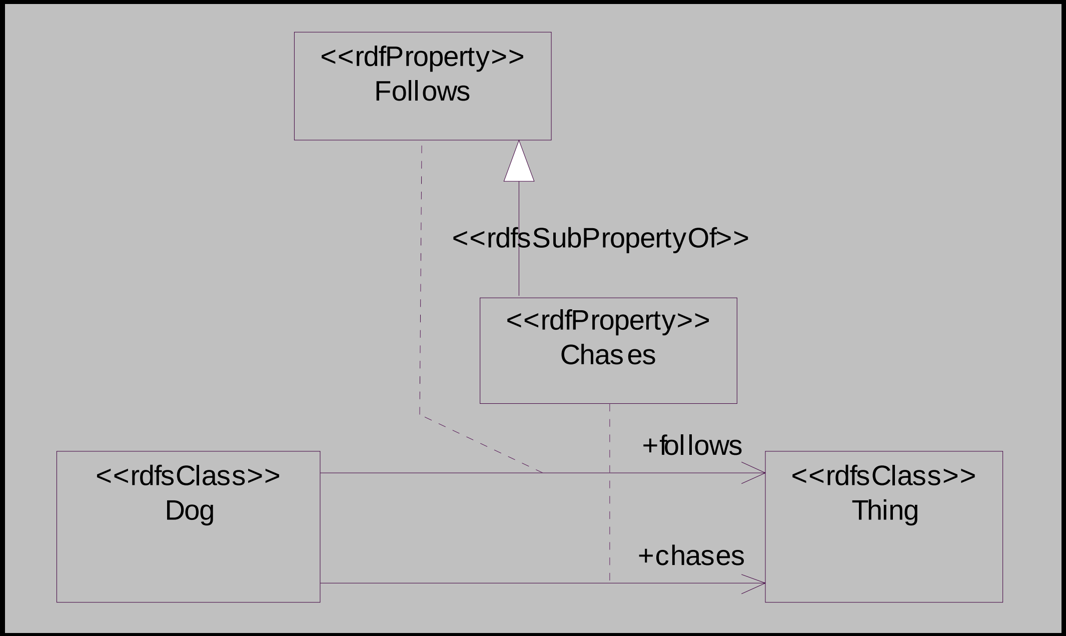


Alternate forms for rdf:Property, without a specified domain

RDF Property Subsetting Options

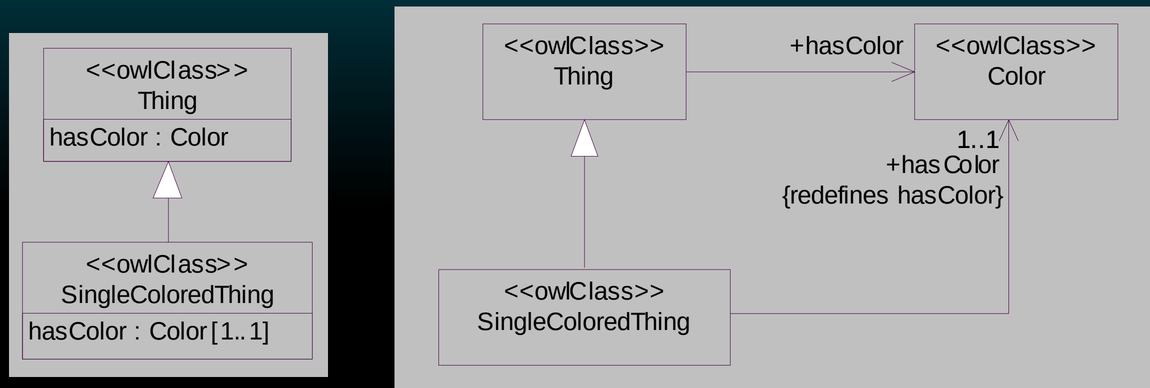




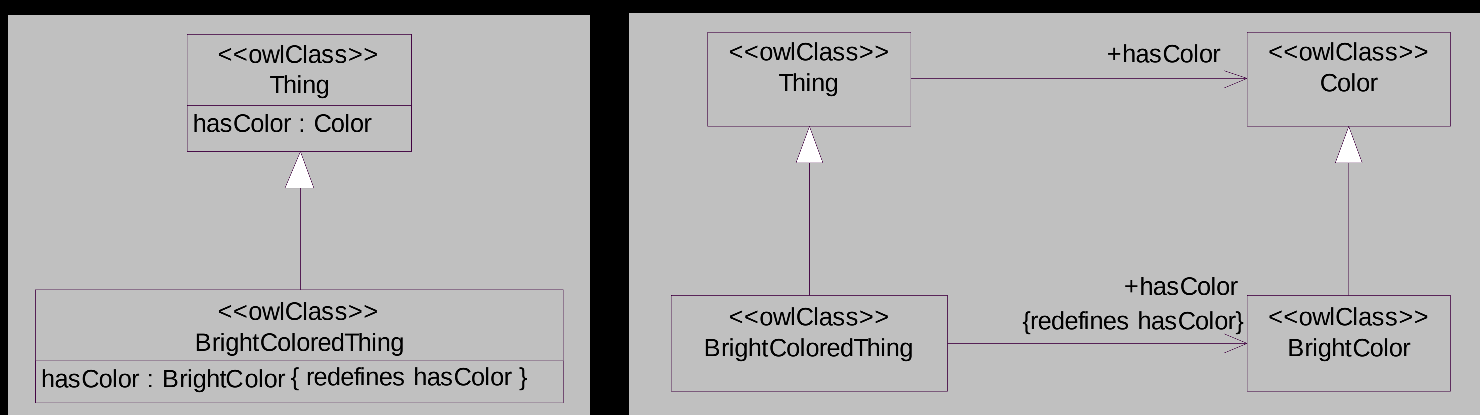


Alternate forms for rdf:Property, without a specified range

Example OWL Number, Value Constraints

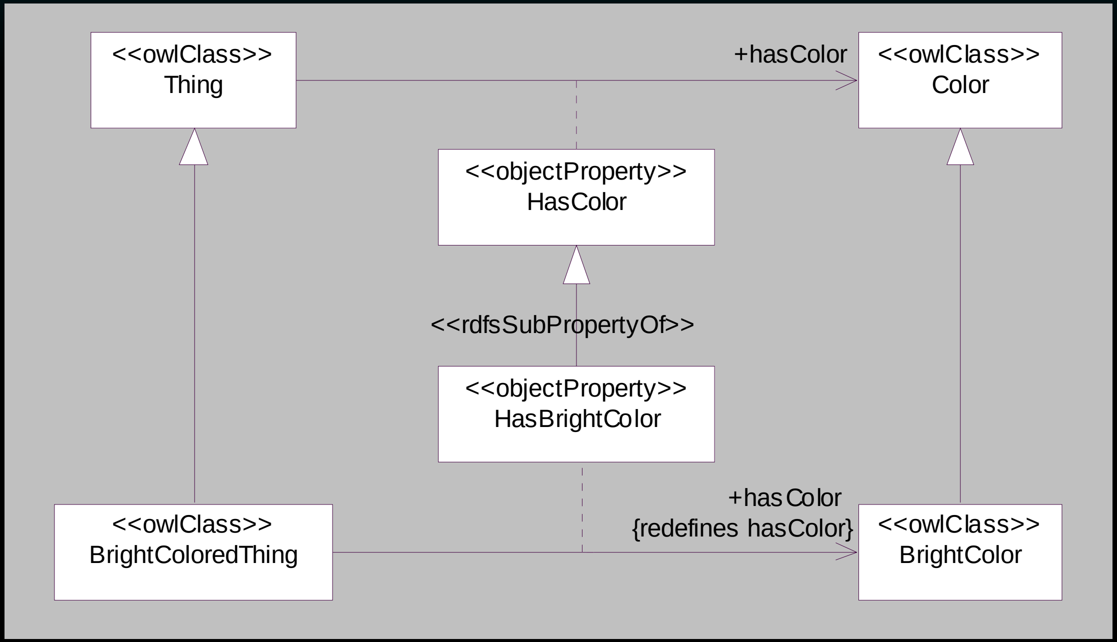


OWL Cardinality – Restricted Mulitplicity in Subtype

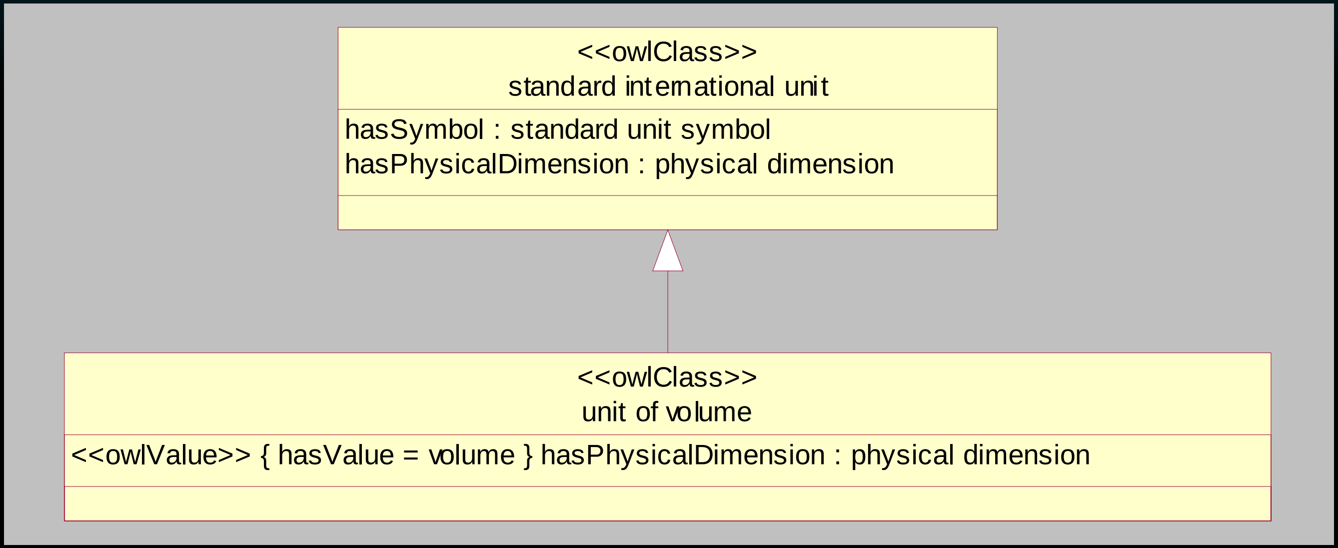


OWL allValuesFrom – Property Redefinition

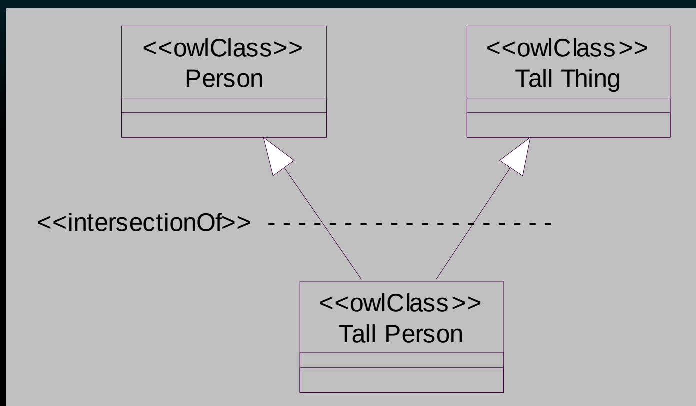
OWL Property Redefinition (allValuesFrom) Using Association Classes

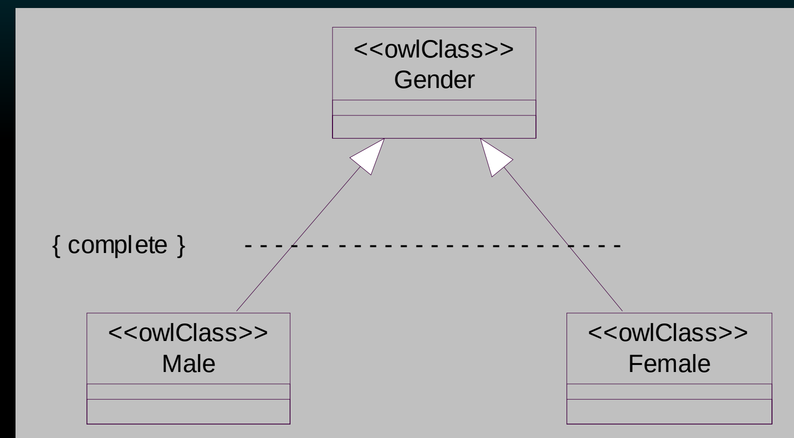


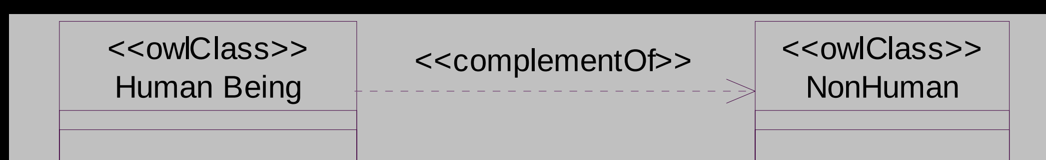
OWL Property Redefinition (hasValue)



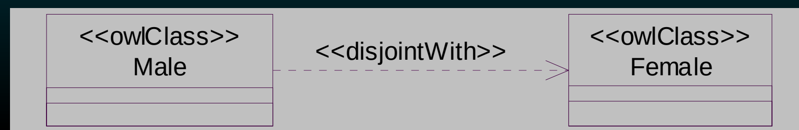
OWL Intersection, Union, Complement



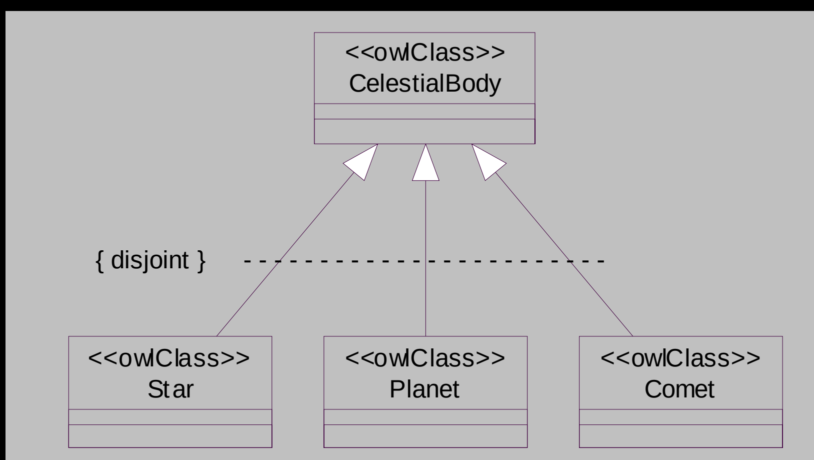




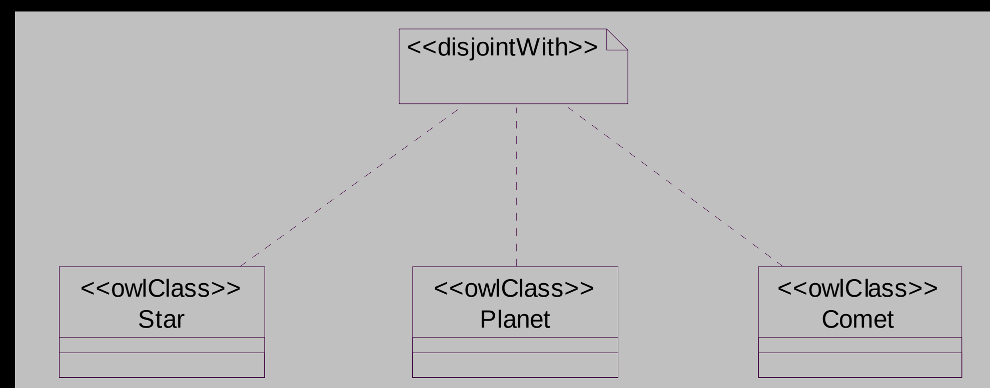
OWL Disjointness Options



Simple binary disjoint relationship

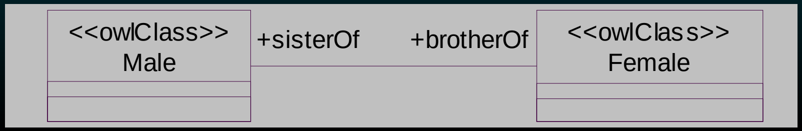


Disjointness, multiple participants, common parent

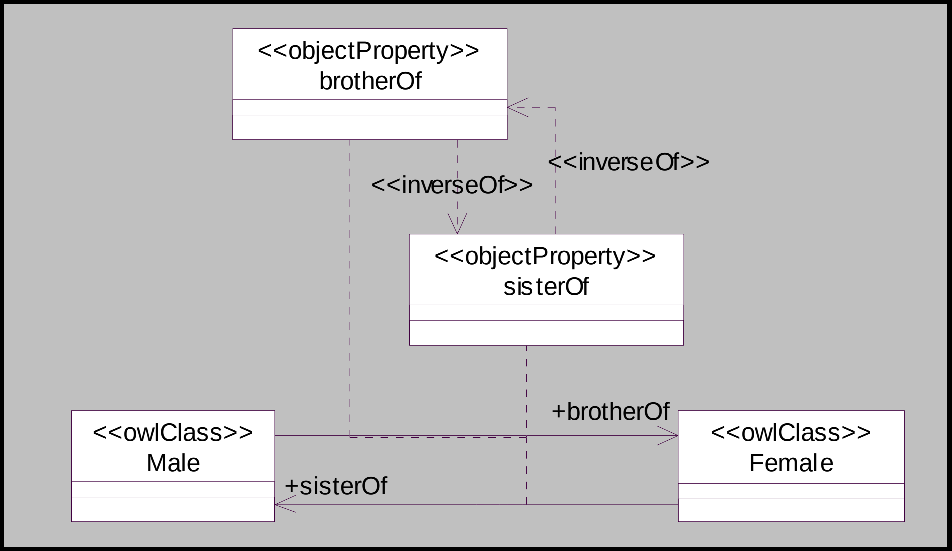


Disjointness, multiple participants, no common parent

OWL Inverse Options



Simple inverse relationship

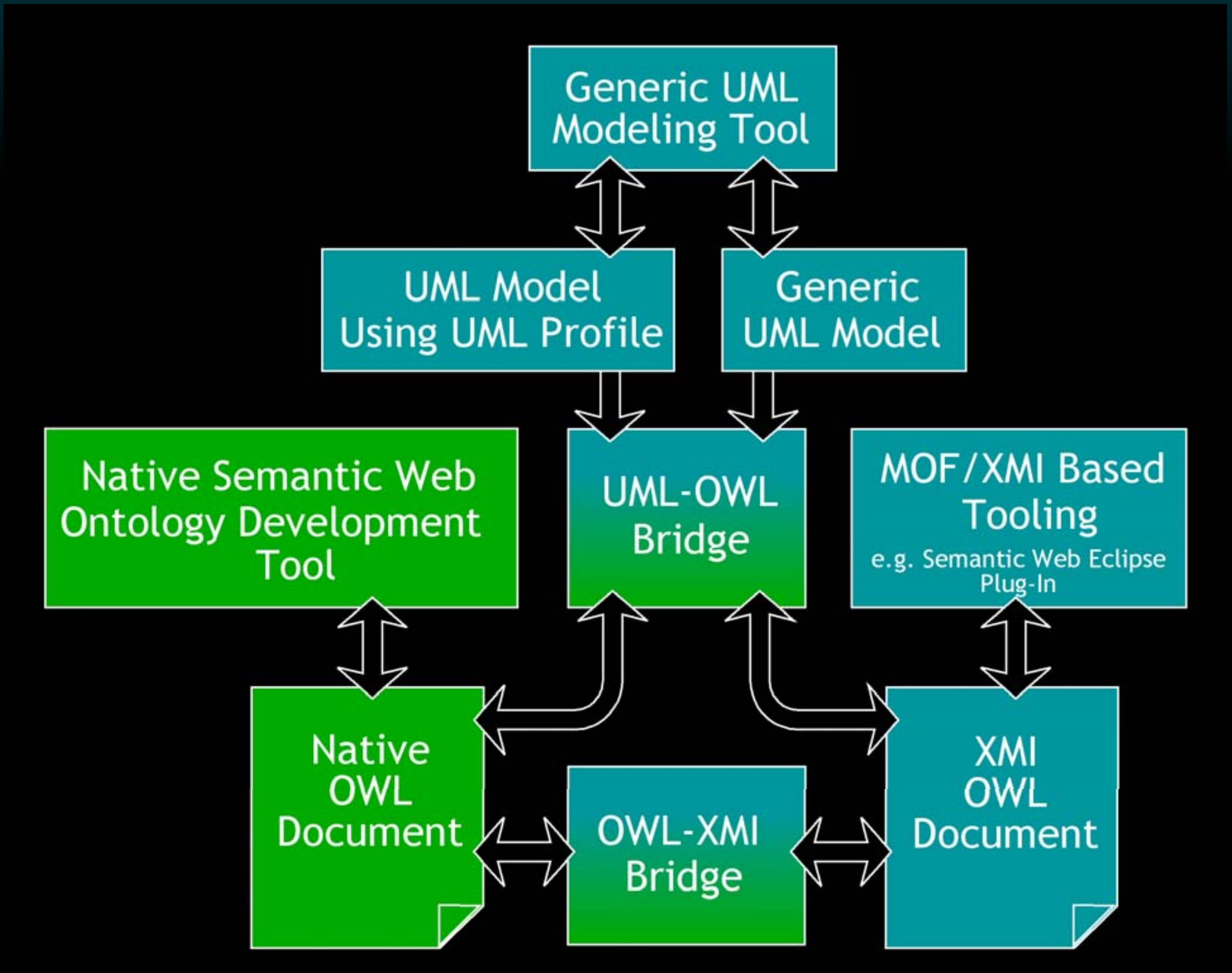


Inverse relationship among association classes

ODM Status

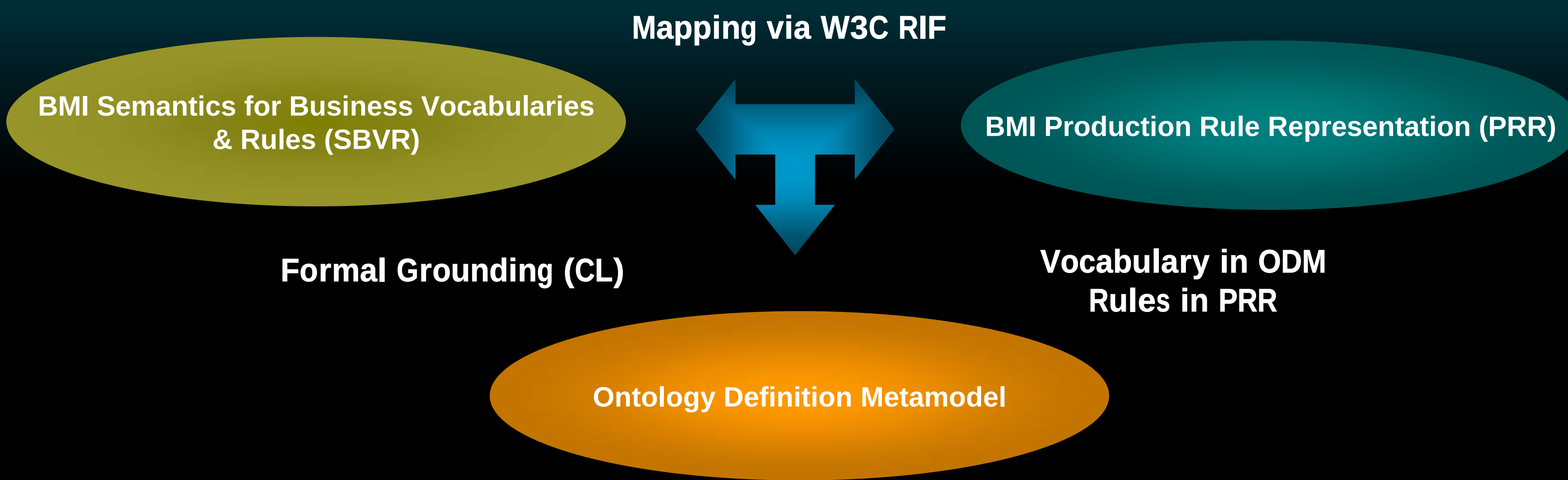
- Platform Independent (Normative) Metamodels (PIMs) include
  - RDF & OWL – abstract syntax, constraints for OWL DL & OWL Full, several compliance options
  - ISO Common Logic (CL)
  - ISO Topic Maps (TM)
- Informative Models
  - DL Core – high-level, relatively unconstrained Description Logics based metamodel (non-normative, informational)
  - Identifier (keys) model extension to UML for ER
- Latest revised submission posted 4/3 to the OMG web site *(http://www.omg.org/docs/ad/06-01-01.pdf)*
- Update includes minor metamodel changes, new MOF QVT mappings, revised RDF & OWL profile, mini-tutorial on use of QVT, etc.
- Next revision will be posted June 5 (three weeks prior to the Boston meeting) – vote for adoption planned for Boston, with remaining clean-up anticipated in finalization

Bridging KR and MDA



sandpiper software

ODM Relationship to Other OMG Standards



ODM extensions under consideration

- Lossy mapping from CL to RDF/S & OWL
- Support for Semantic Web Services (SWSF, OWL-S), bindings to WSDL & SOAP
- Mappings for W3C Rule Interchange Format (RIF) (*i.e.* vocab/ontology → rules, including PRR)
- Mappings for Emerging OMG Information Management Metamodel (IMM) - including potentially ER, ISO Express
- New requirements from SOA ABSIG anticipated

Relationship to ISO Standards

- CL Metamodel is included in ISO FCD 24707
- High degree of synergy between ODM and Topic Maps ISO FCD 13250-2 working group
- All ODM metamodels are referenced and used in ISO CD 19763 (MMF – Metamodel Framework, Model Registry specification)
- All ODM metamodels inform latest modifications proposed in ISO draft 11179 Metadata Registration specification
- ODM team is working with DoD XMDR team to promote interoperability among ODM, ISO 19763, ISO 11179 metadata standards efforts
- Current work in OMG to develop a metamodel for ISO Express will include mappings to ODM
- Mappings from multiple components of IMM (*e.g.*, ER, ISO Express) are under consideration
- Sandpiper provides standards liaison for emerging DoD Semantic Service Oriented Architecture (SSOA) framework development

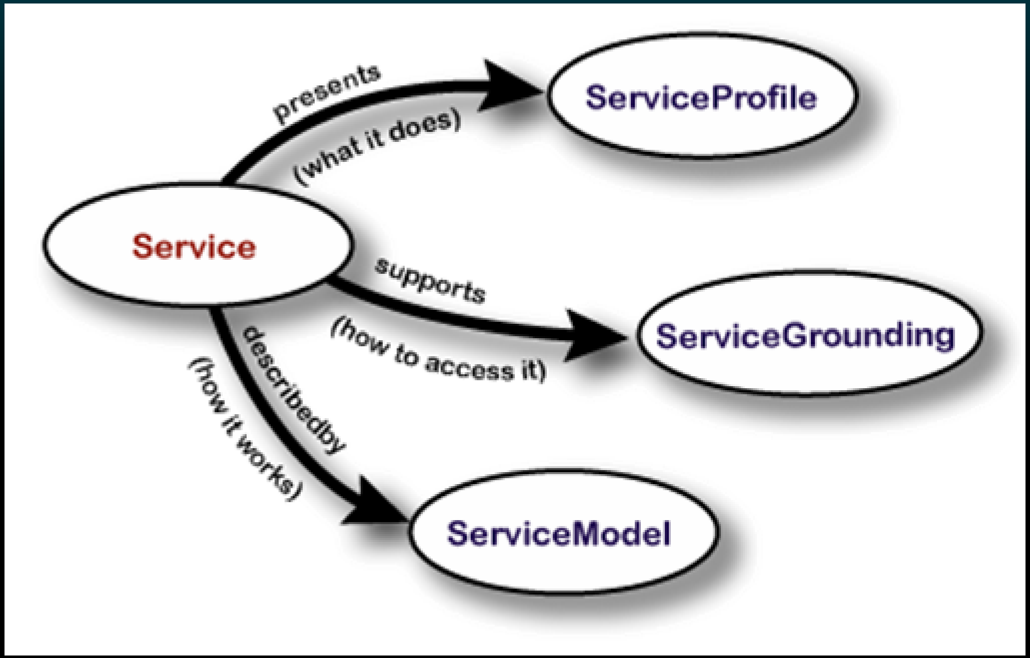
Why Semantics for Web Services – Quick Review

- Ontologies provide a common vocabulary and definition of rules for use by independently developed services
- Companies and organizations sharing common services can declaratively specify the *behaviors*, *policies* and *agreements* relevant to their usage
- Automation of service use by software agents
  - Goal/vision: dynamic discovery & use of new services, previously unknown, to complete task
  - Reasoning about services: support on-the-fly composition
  - Integrated use with other information resources: ultimate, fully-automated customized, user experience
- Composition, mapping and vocabulary brokering for independently developed resources and services – enables information sharing & process enactment consistently, accurately, and dynamically
- OWL-S, SWSF complement WSDL by providing an abstract or application level description lacking in WSDL

OWL-S: Enabling Infrastructure for Web Services

- Based on research from the DARPA/DAML program in DAML-S (2000/2001 - primarily at SRI, Stanford & CMU)
- OWL-S – an ontology that sits at the application level, above WSDL, and describes *what* is being exchanged and *why*, not just the *how*
- OWL-S enables
  - *discovery* – of services that meet particular requirements and adhere to specified constraints
  - *invocation* – and execution by agents or other services
  - *interoperation* – through specification of the appropriate vocabularies (semantics) and message parameter translation as required based on service specifications
  - *composition* – automated service composition and interoperation to provide new services
  - *verification* – of service properties
  - *execution monitoring* – tracking of execution of complex services and transactions

Top-Level of the Service Ontology

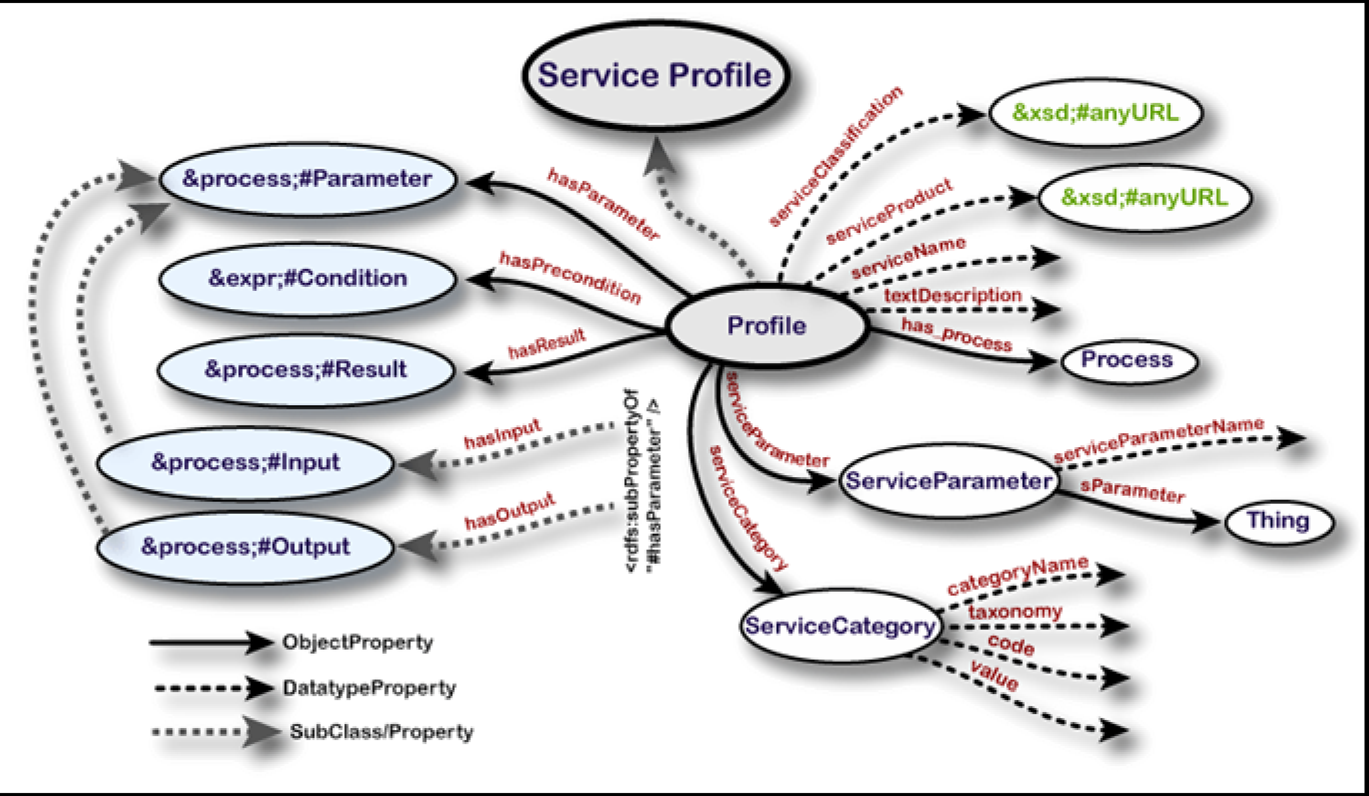


Three essential types of knowledge about services

- **The *what*, its capabilities and parameters, through a *ServiceProfile*, which can answer questions such as what does the service require of agents and provide for them**
- **The *how*, through a *ServiceModel* that describes the workflow and possible execution paths**
- **Accessibility and usage through a *ServiceGrounding***

OWL-S Structure

- Service profiles are used to request or advertise services with discovery services and capabilities registries, including
  - Descriptions of services and providers
  - Functional behavior & attributes



Semantic Web Services Framework (SWSF)

- Emerged from work in services composition
  - May require more expressivity than is available in OWL
  - Based on logic programming, first-order logic, policy research
- Considered smorgasbord of standards
  - Web Services Description Language (WSDL) – for input & output messaging, invocation (W3C)
  - Business Process Execution Language for Web Services (BPEL4WS) – workflows of basic services (OASIS)
  - Choreography Description Language (WS-Choreography) – more global view of information exchange from a transaction perspective (W3C)
  - UDDI – standard approach for service registration, discovery, & advertising
- Builds on DAML-S, OWL-S, WSMO
- Provides rich semantics for greater automation of discovery, selection & invocation, content transformation, composition, monitoring & recovery, verification

Semantic Web Services Framework
SWSL & SWSO

- Semantic Web Services Language (SWSL)
  - SWSL-FOL – first order language for ontology representation, builds on CL
  - SWSL-Rules – logic programming to enable ontology use in reasoning and execution environments
- Semantic Web Services Ontology (SWSO)
  - Conceptual model, complete axiomatization expressed in SWSL-FOL
  - Called FLOWS – First-Order Logic Ontology for Web Services
  - Includes model theoretic semantics
  - Ontology translated to SWSL-Rules is slightly more constrained,
  - Called ROWS – Rules Ontology for Web Services
- W3C Note & member submission
  - *http://www.w3.org/Submission/SWSF/*

Additional Candidates, Historical Perspective

- Web Service Modeling Ontology submitted to W3C April 2005
  - http://www.w3.org/Submission/2005/06/
  - WSMO Web Service Modeling Ontology
  - WSML – Web Service Modeling Language
  - WSMX – Web Service Execution Environment (WSMX)
- WSDL-S Web Service Semantics submitted Nov 2005
- Semantic Web Services Interest Group formed: http://www.w3.org/2002/ws/swsig/
- June 2005 Meeting held in Innsbruck http://www.w3.org/2005/04/FSWS/program.html

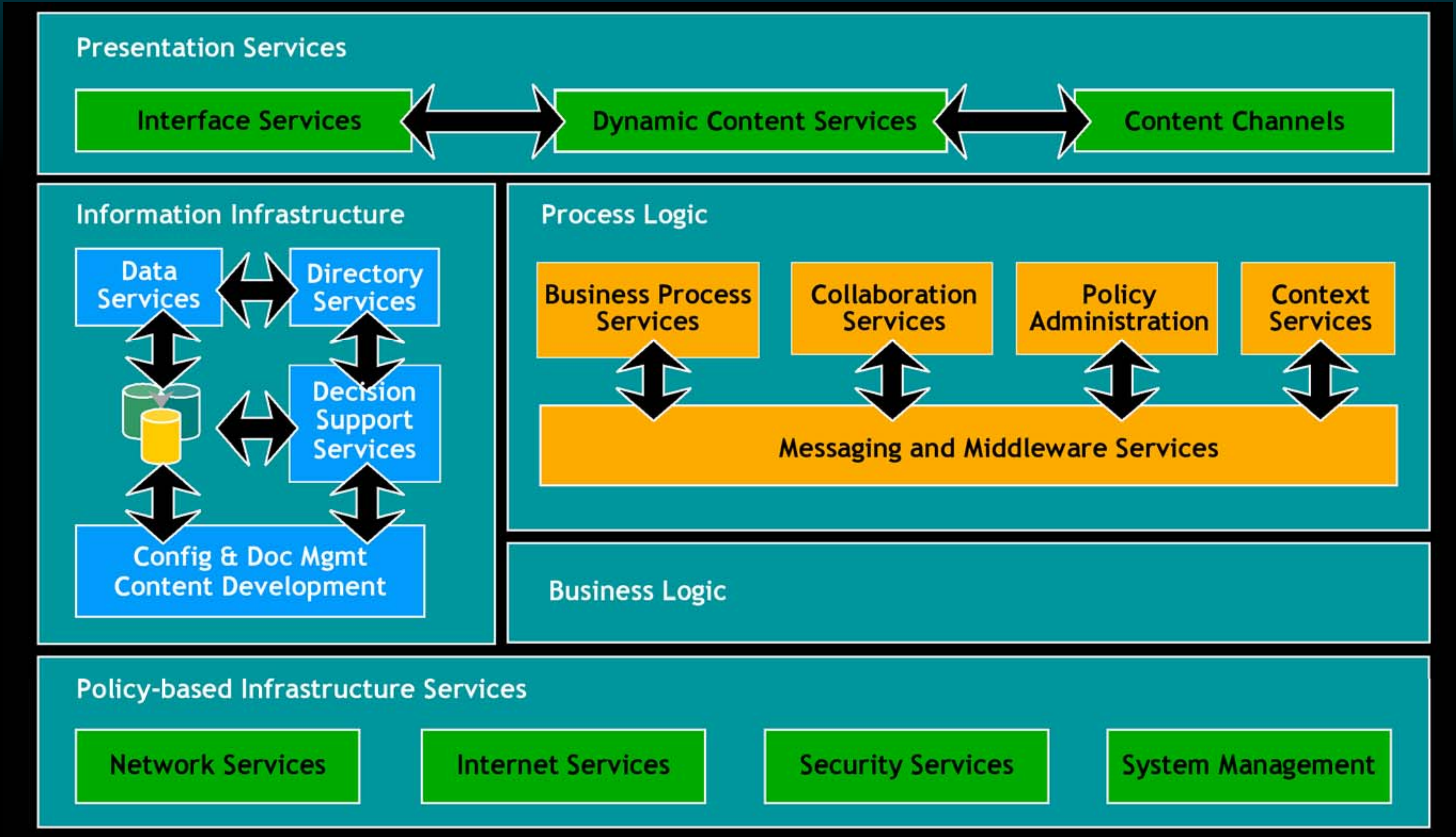
Progress & Current Status

- **Semantic Annotations for WSDL Working Group chartered March 21st:** *http://www.w3.org/2002/ws/sawsdl/*
- **Working draft for SW Rule Interchange Format (RIF) Use Cases and Requirements published March 27th:** *http://www.w3.org/TR/2006/WD-rif-ucr-20060323/*
- **SPARQL (RDF Query Language) promoted to Candidate Recommendation Status April 6th: see** *http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/*
- **Working group participation is still in formation, initial focus appears to be on semantics of WSDL (*i.e.*, WSDL-S) than on general semantics of services and service interoperability**

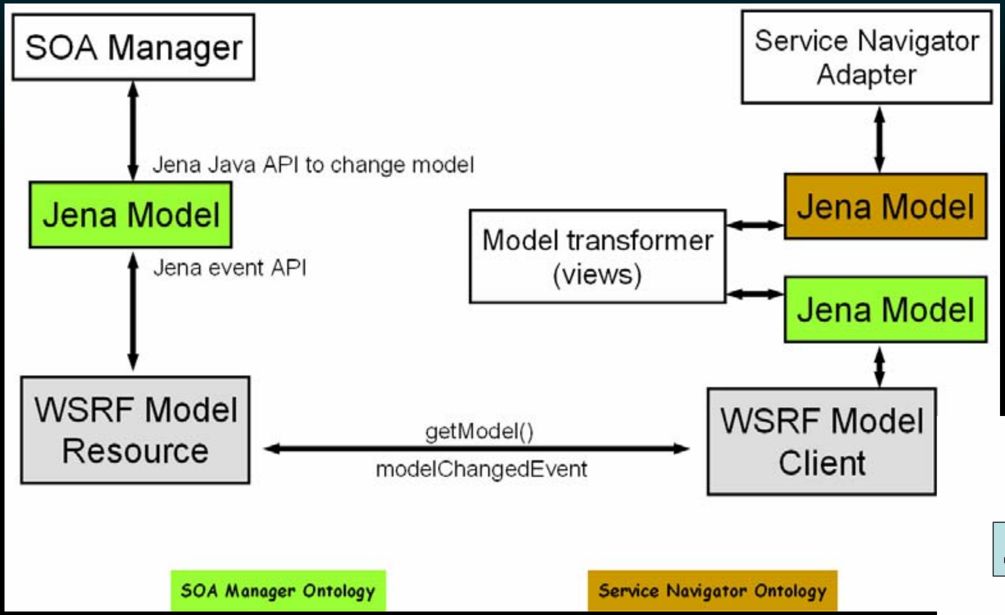
Summary

- Ongoing work in the W3C is moving the ball forward on a number of relevant fronts: RDF Query, Rules, SWS
- Near term roadmap for Ontology PSIG includes MOF revisions to support multiple classification, "Reverse ODM" – representation for MOF in RDF
- Longer term: considering extensions to ODM to support Semantic Web Services, mappings to IMM Metamodels for ER & ISO Express, Rules
  - OWL-S, building on the RDF & OWL metamodels
  - SWSF, building on the CL metamodel, with mappings to OWL-S
  - Mappings to standardize bindings to WSDL, SOAP
- OMG BMI DTF Semantics for Business Vocabularies & Rules (SBVR) will be logically grounded in Common Logic / ODM CL Metamodel
- Planned mapping to forthcoming Production Rule Representation (PRR) specification
- Should also consider leveraging mapping from UML for BPEL to ODM extensions (*e.g.*, to the PSL component of SWSF) downstream
- Requirements and assistance needed

Implementation Strategies



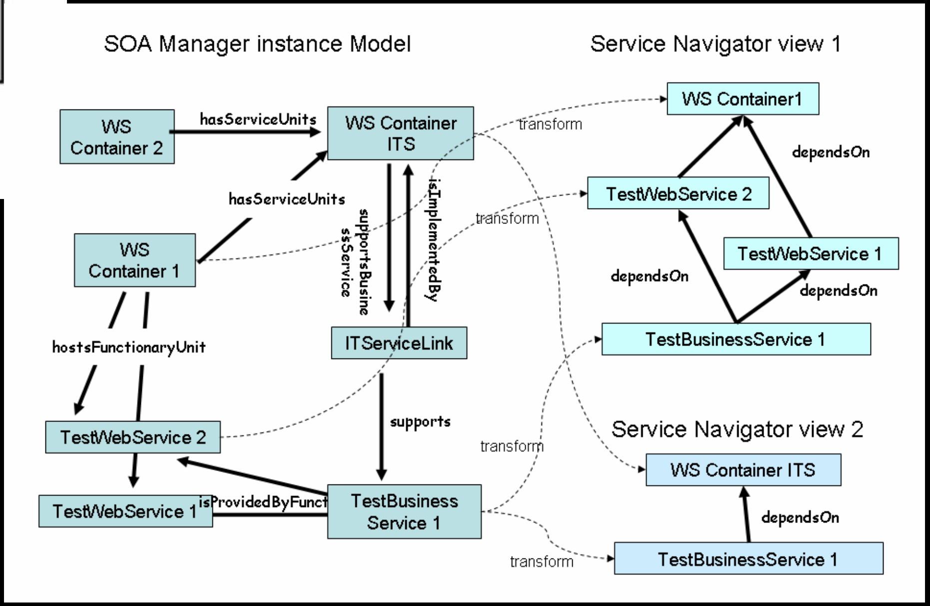
HP OpenView: Management Application Integration Framework



Synchronization of model repositories using RDF/S & OWL based representation & transformations provides new integration capabilities for HP OpenView

Ontology was developed using an ODM-based development environment;

Jena Rules support model transformations



Semantic Service Oriented Architecture (SSOA)

IC Analysts Lack Awareness of Available Capabilities (Services)

- Sheer Volumes of Data and Services Compounds the Problem
  - Word of Mouth Awareness Typical
- Available [XML] Web Services Solutions are Pervasive, but…
  - Lack Ability to Easily Discover Services
  - Are Location Dependent; "Stale" References Possible
  - Have Weak, or No Semantics
  - Include Ever-growing Multitude of Largely Unimplemented Standards (re: WS-*)
- When Found, Services Typically Not Easy to Use, Not Interoperable
  - Services are Not Described by "What They Provide"
  - Machine Interpretable Standards Immature, Not Implemented
  - Resulting Processes are Human-Centric, Ad-hoc and Intermittently Repeated
- Resulting in Fragmented, Sub-Optimal Analysis
  - Long Standing Problem – Analysts Aren't Able to Focus on Analysis

SSOA slides courtesy Sam Chance, Scientific Research Corporation, Concurrent Technologies Corporation

SSOA Value Proposition

Complements current XML-based Web services architectures:

- *Semantically enabled*
  - **Efficient publishing, discovery, and execution of all available services**
  - **Recommends interesting services to users when new services come online**
  - **Enables software agents to dynamically construct workflows and substitute services upon failure**
  - **Designed and implemented based on current and emerging Semantic Web standards**
  - **Enables composition of virtual applications based on semantics**

SSOA Value Proposition

- *Powered by an extended Jini™ based platform*
  - Flexible location independent services, spontaneous networking & services interchange
  - Self-healing from network failures, proactive system health monitoring
  - Enables near-real time collaboration & capabilities sharing
  - Grounded on a Proven Enterprise-Scale Distributed Computing Model
- *Demonstrates resource sharing across disparate organizations*
  - Enhancing current SOA projects by acting as risk reduction/complementary task

SSOA System View

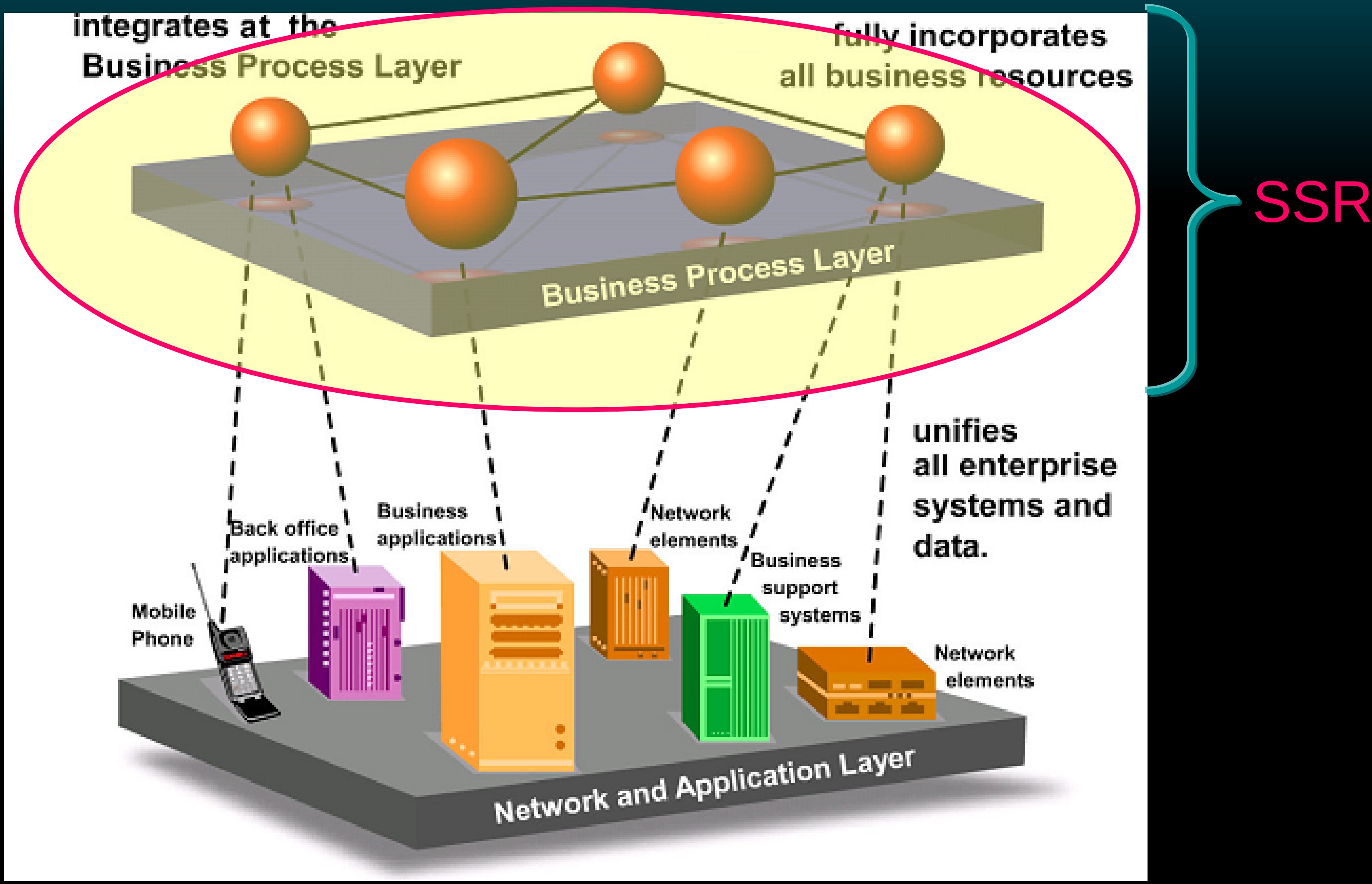
⑩ Semantic & Agent Components

⑩ Services

...ervices
...cally Exchanged
...g/Awareness)

sandpiper software

Powerful Jini-Based Abstraction



sandpiper software

CSAR Process (Greatly) Distilled

IAW Doctrine JP 3.50-2

Pr on

Manpower
Assets

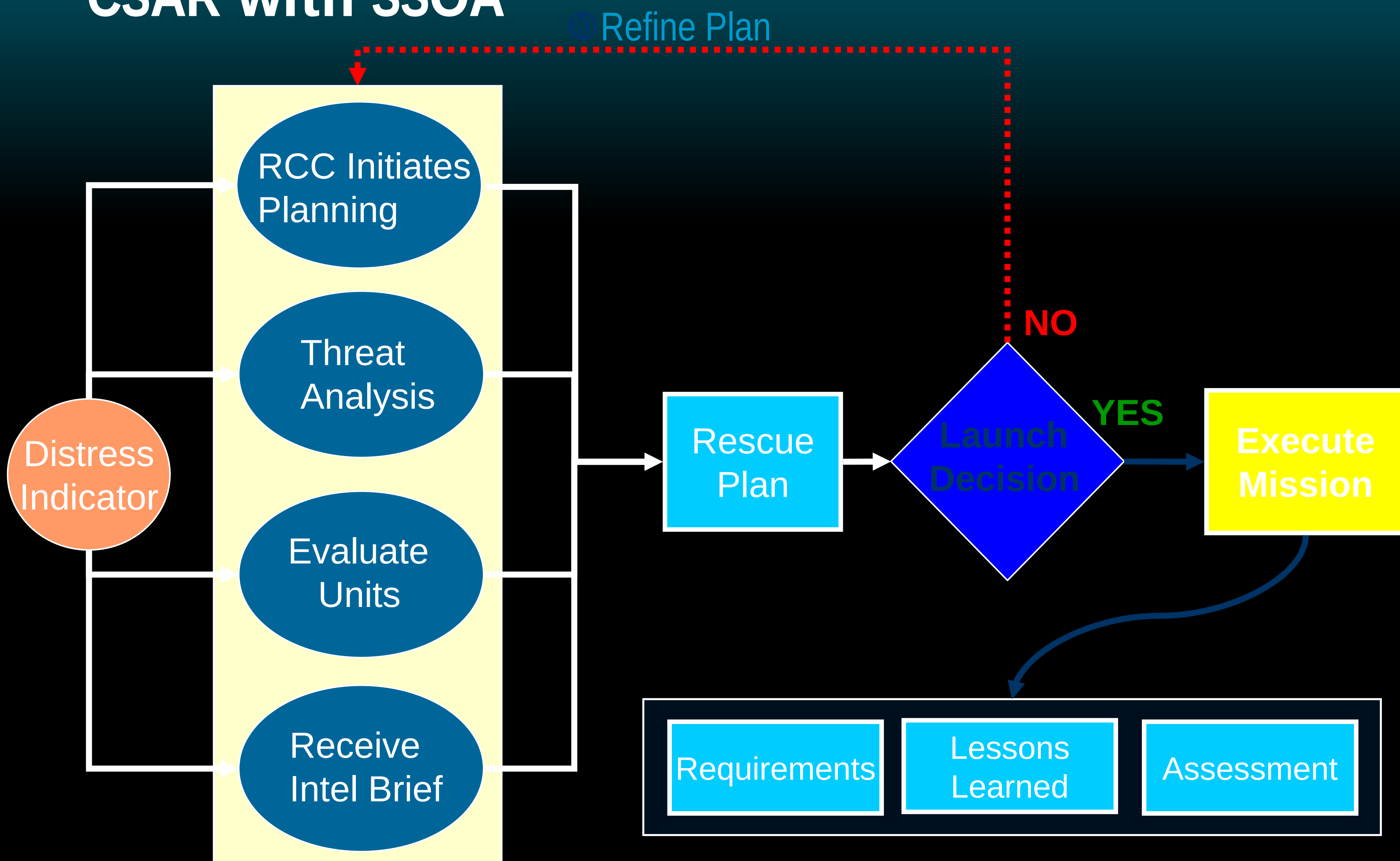
Weather
Imagery
Enemy Intel
Terrain

Planning

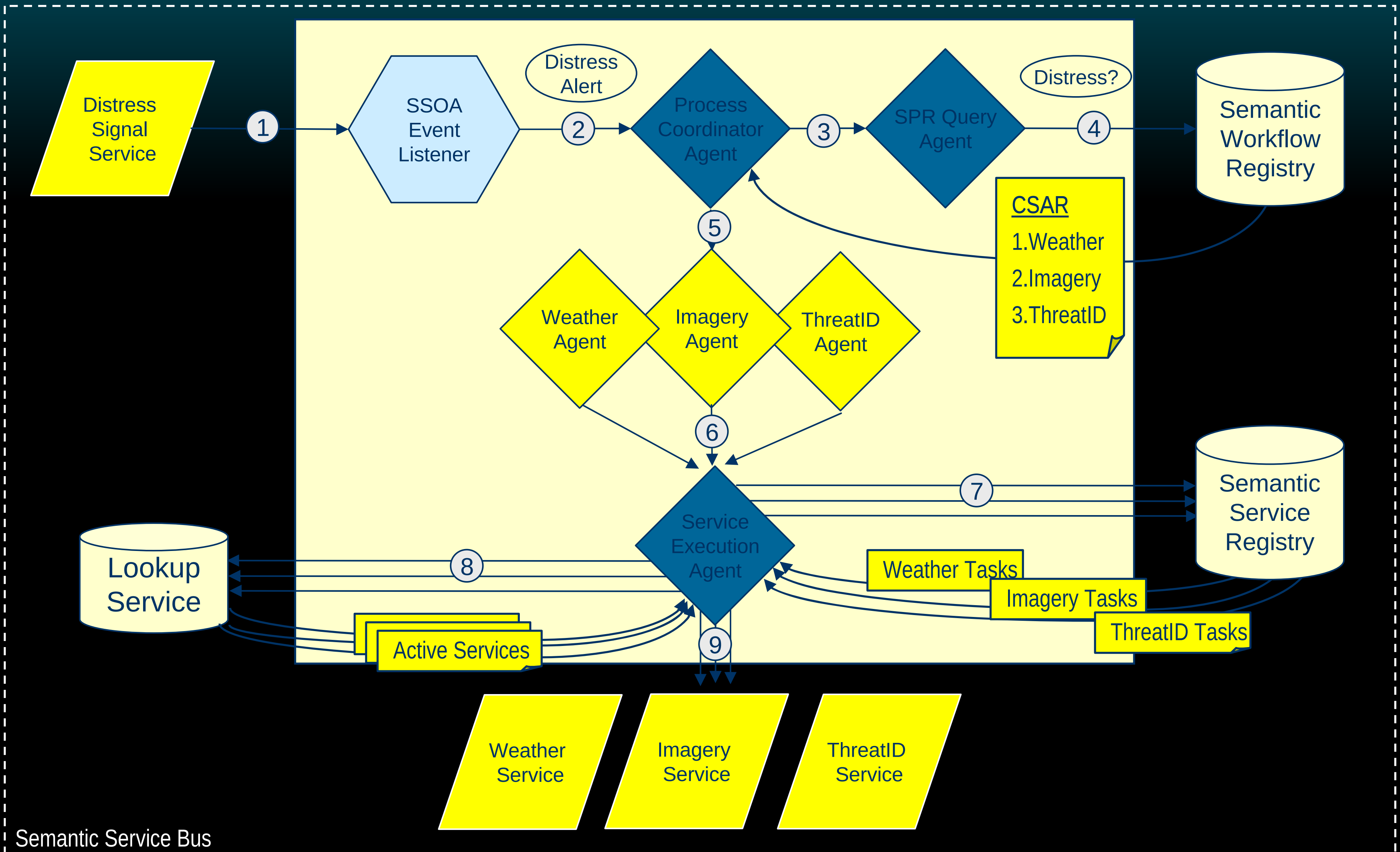
Execution

Adaptation

CSAR with SSOA



SSOA Event Flow – CSAR Demonstration

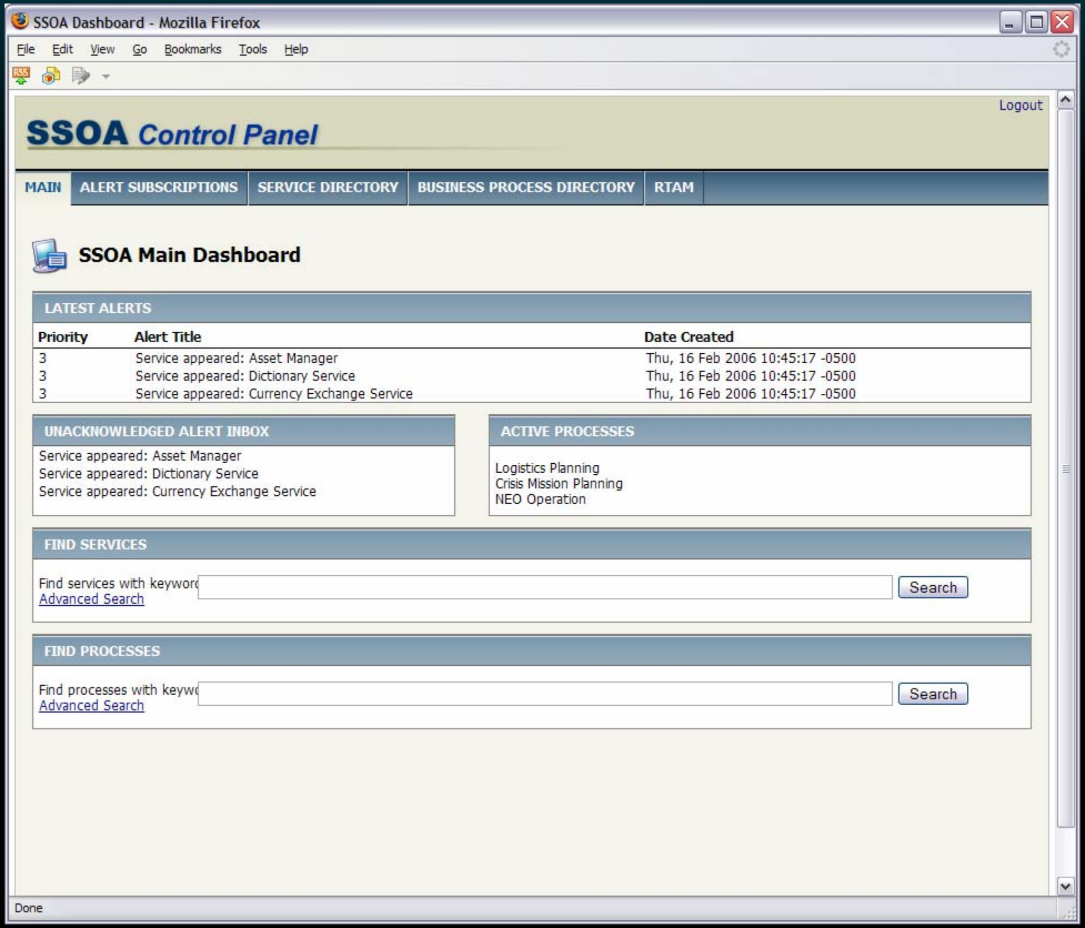


SSR Functionality

- Key Concept: Every Service Type is a Collection of Tasks
  - Each Task Semantically Corresponds to a Specific Operation or Action
- Every Service is a Running Instance of a *Service Type*
  - Multiple Instances of a Specific Service Type May be Deployed to:
    - Build in Redundancy & Provide Load Balancing
- Task Selection Depends on a Semantic Description, Comprised of:
  - Input, Output & Action Types
- Given a Set of these Input, Output & Action Types, the SSR will:
  - Return the Candidate Tasks and Associated Service Types,
  - Provide Necessary Information to Discover and Execute Any of the Tasks Within the SSB
- Weather Service Example:
  - Inputs are: { Location, TimeStamp }
  - Outputs are: { TemperatureC }
    - SSR Matches 3 Potential Tasks:
      - TemperatureC getTemp(Location,TimeStamp)
      - TemperatureC temp(Location,TimeStamp)
      - TemperatureC getTempCelsius(TimeStamp,Location)
  - But Not:
      - TemperatureC getTempCelsius(TimeStamp,Location,Altitude)

SOA Events Dashboard

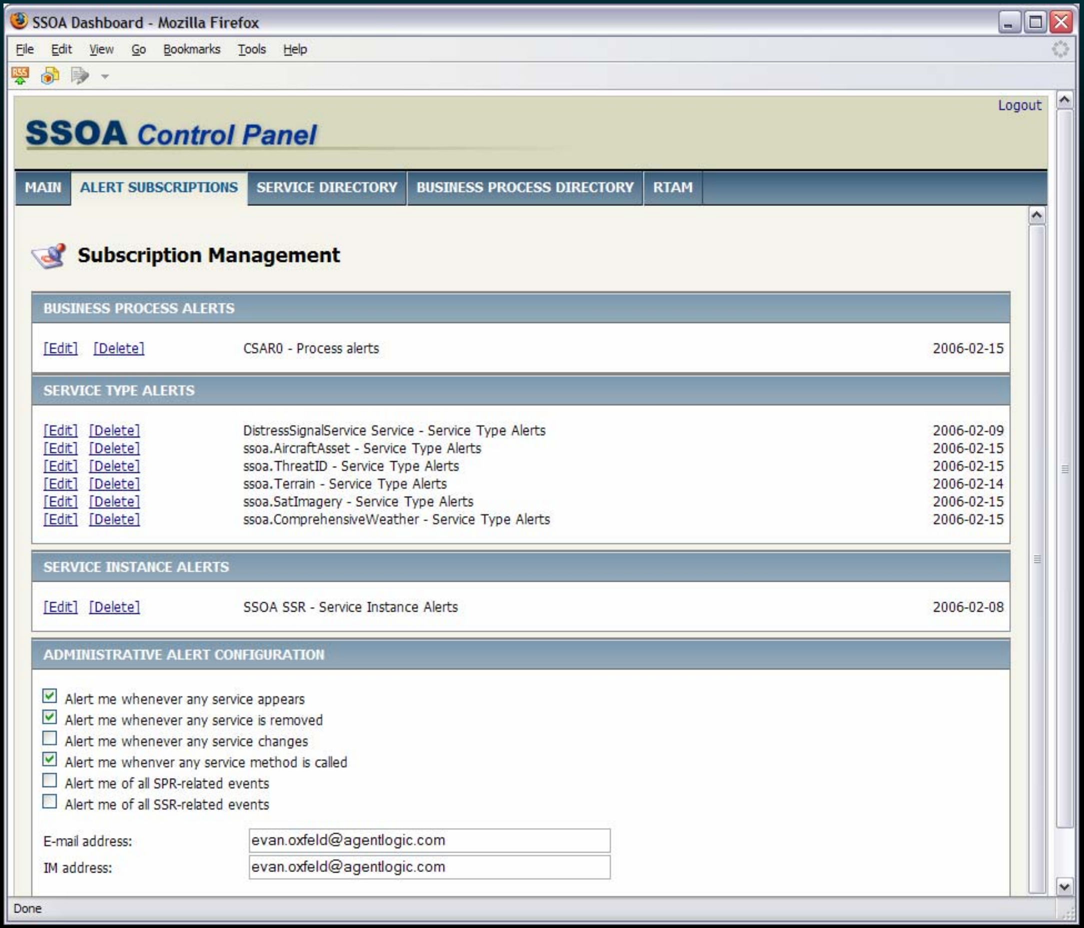
- Overall user experience is delivered via EAS
- Main Dashboard provides summary of EAS alerts and link to the Real Time Alert Manager™



SSOA Demonstration Slides courtesy Mauricio Renzi, AgentLogic

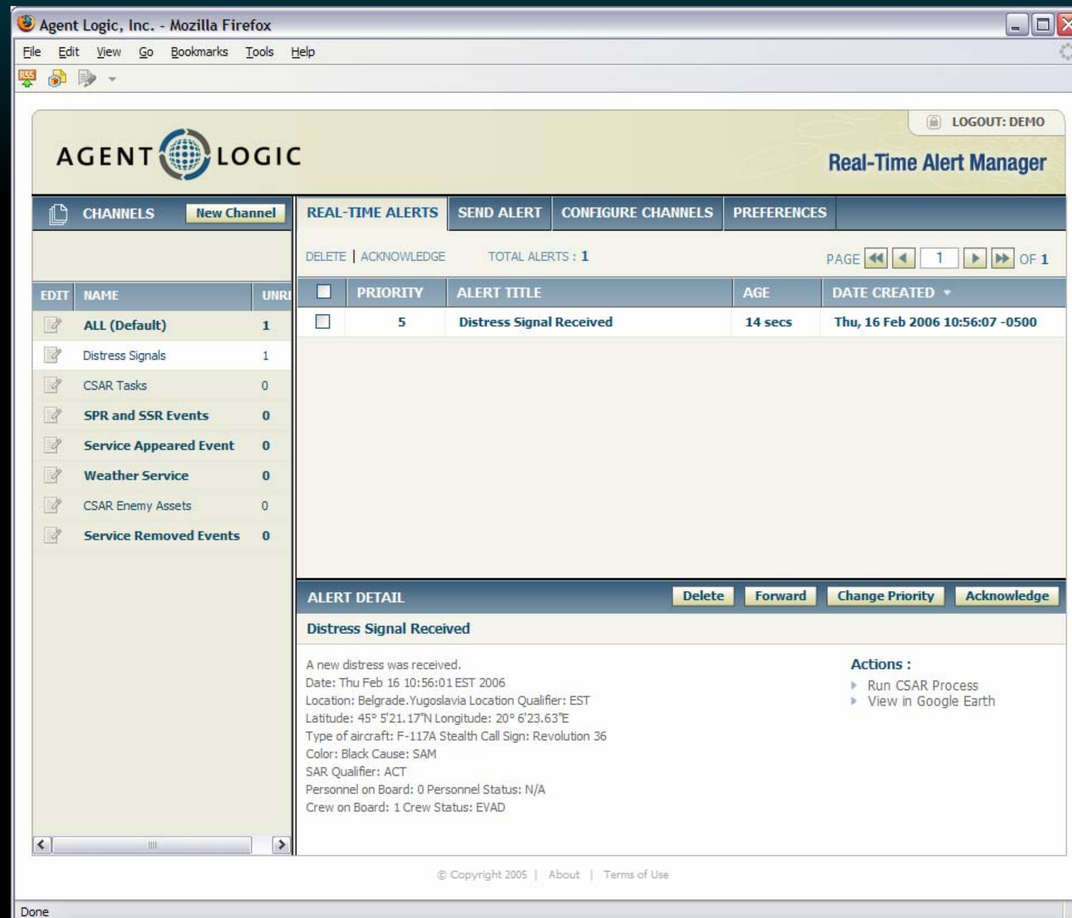
Subscription Management

- Subscription management capability allows end users to define notification rules based on the attributes of incoming events
- Subscriptions can be tied to business services, i.e.:
  - Notify if the Distress Signal Service generates an event that occurred within a specified geo-bounding box
  - Notify if an entity extraction service generates output containing entity “John Doe”



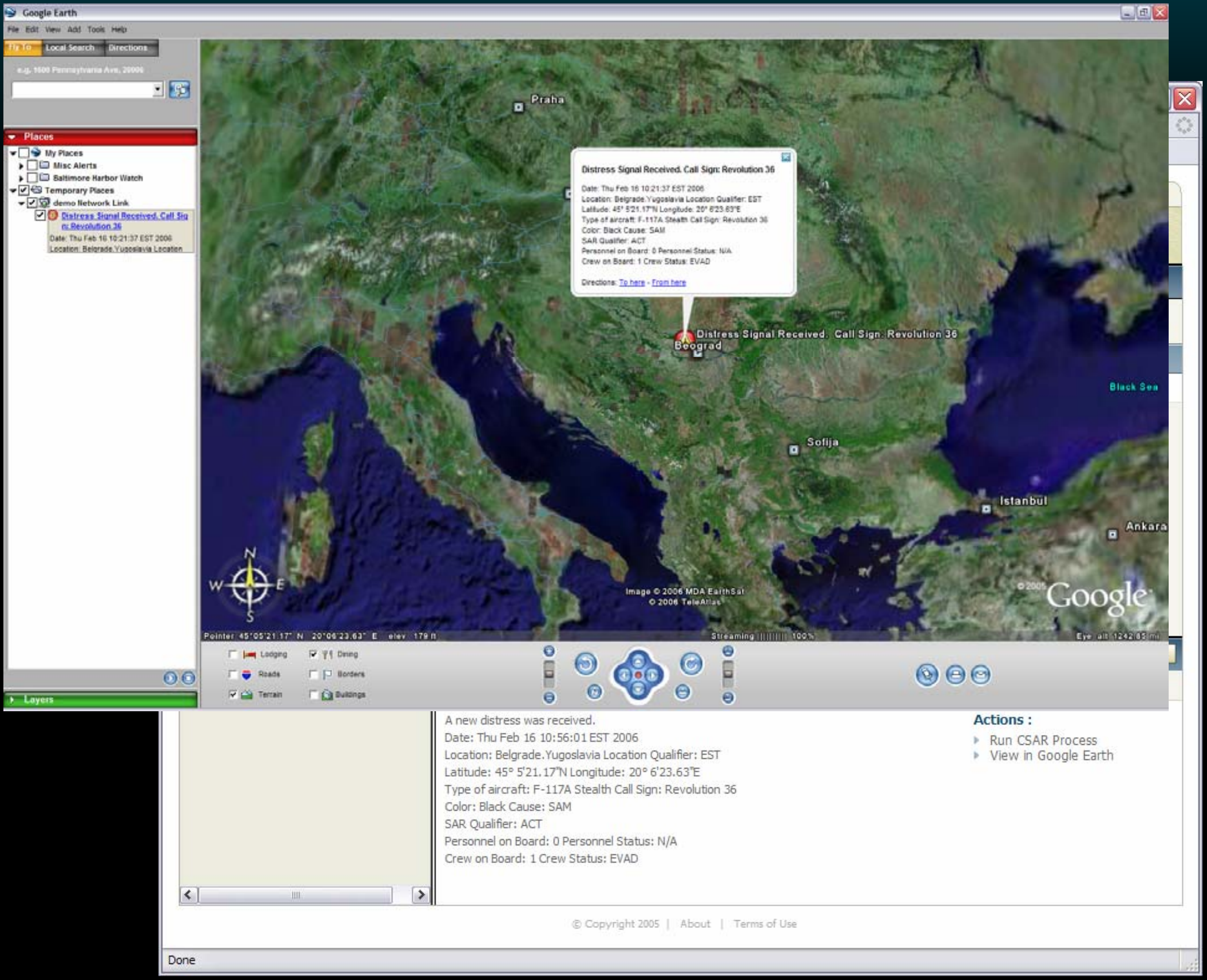
Agent Logic: Real Time Alert Manager™

- When distress signal event is received, notifications are delivered to subscribed end users
- Alert is displayed in Real Time Alert Manager and sorted into the appropriate channel
- Alert contains data associated with the triggering event (location, time, crew status, etc.)
- User is provided contextual actions
  - View in Google Earth
  - Launch the CSAR collection process



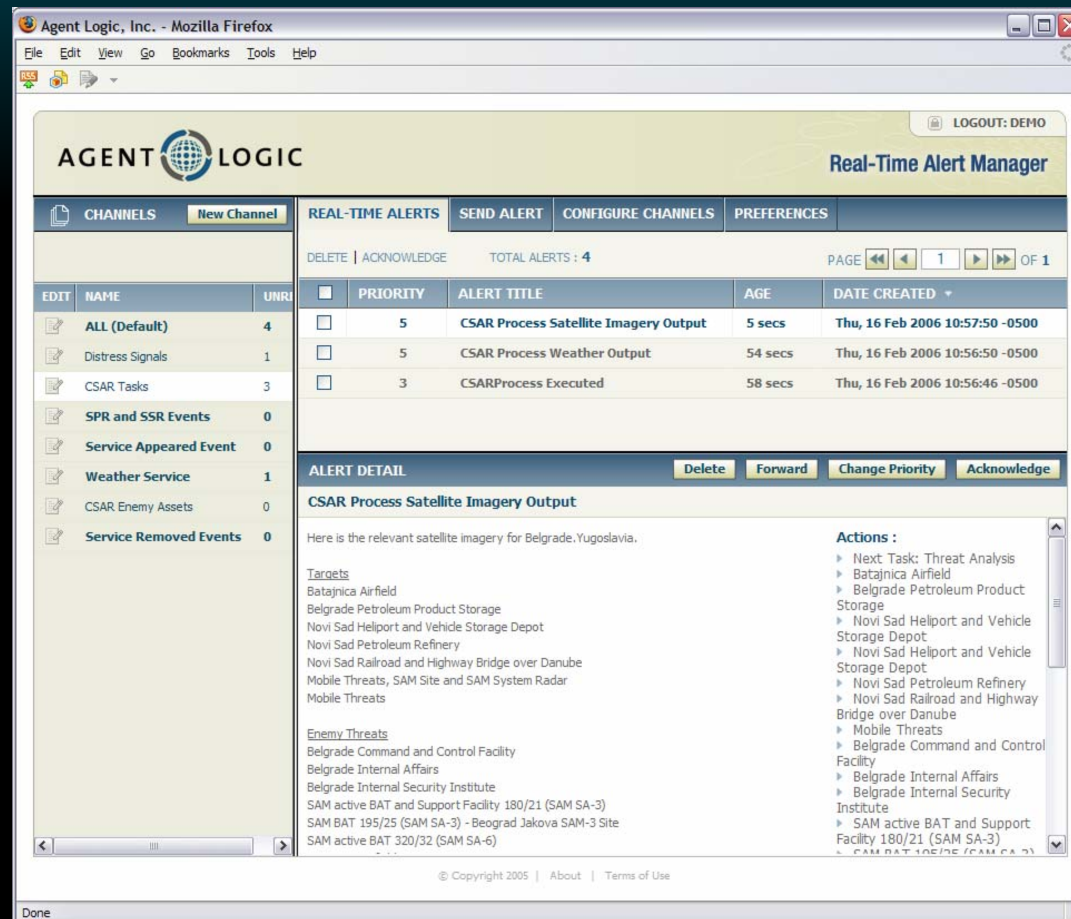
Google Earth Integration

- Alerts are also routed to Google Earth by writing to network accessible ‘placemark’ files
- Provides an alternative real time alerting interface



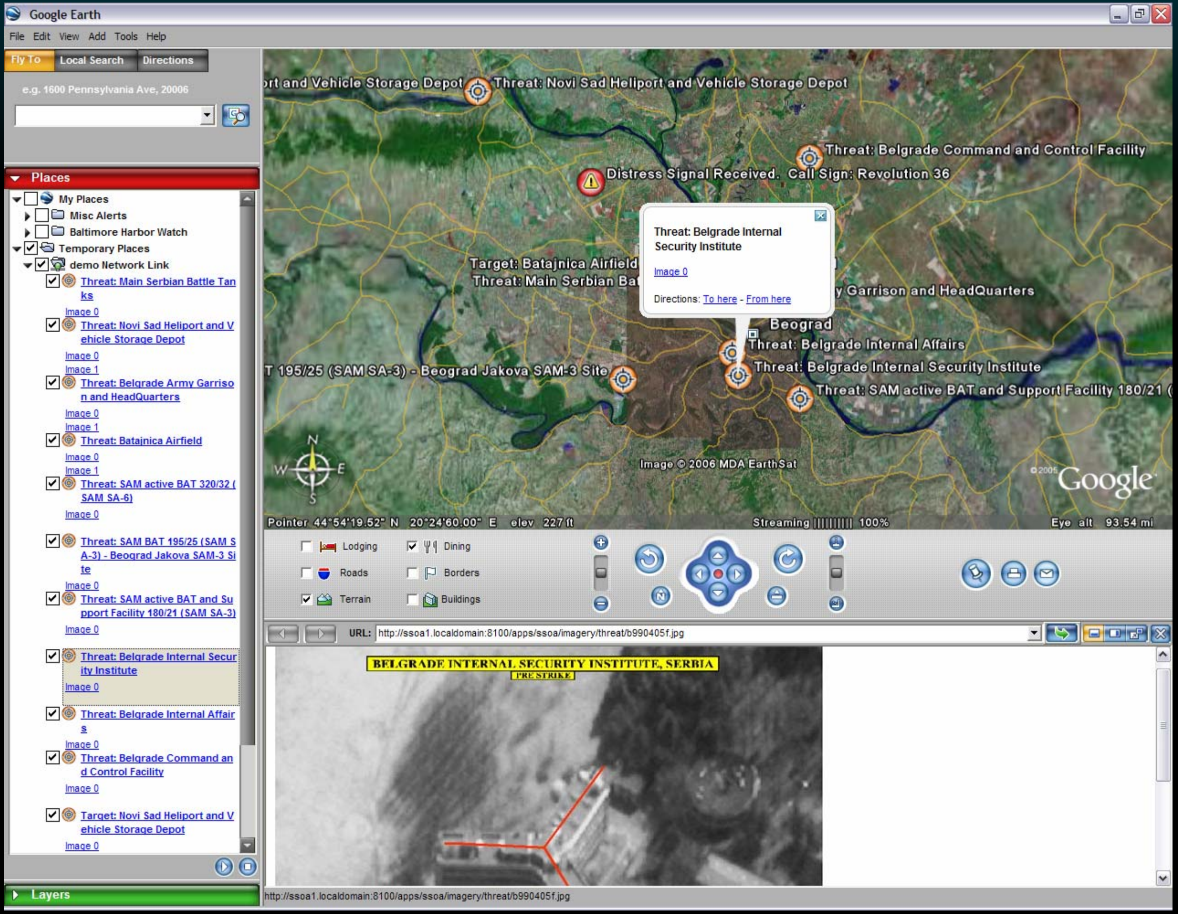
Automated Data Gathering / Contextual Actions

- Agents retrieve data from other business services, using attributes of the original event to formulate queries (i.e. LAT/LONG)
- These data points are processed and presented to subscribed users in real time as RTAM alerts



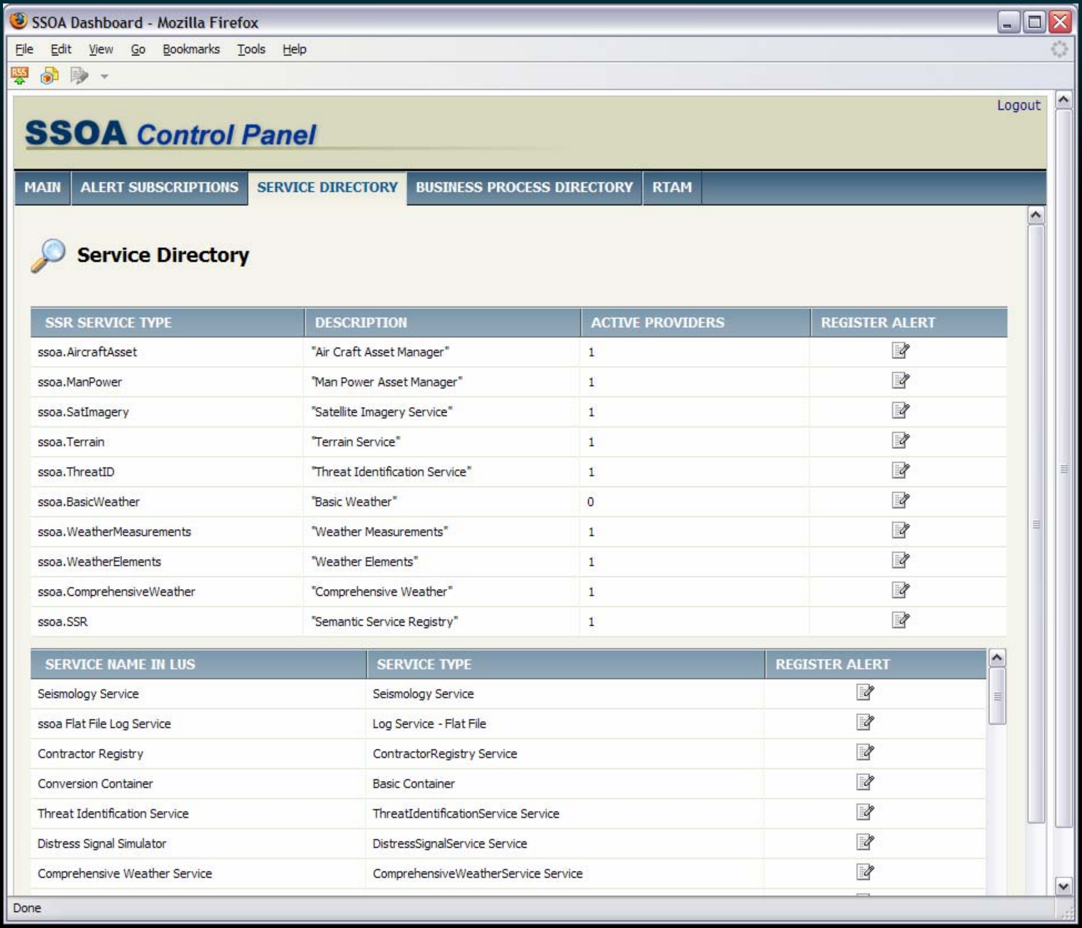
Geospatial Threat Map

- As the agent gathers additional intelligence from other services, a ‘threat map’ begins to take shape…
- Mission commanders utilize this event-driven, automated process to assess mission risks
- Friendly force information is added to the threat map to allow mission commanders to identify which resources can be tasked for a rescue mission



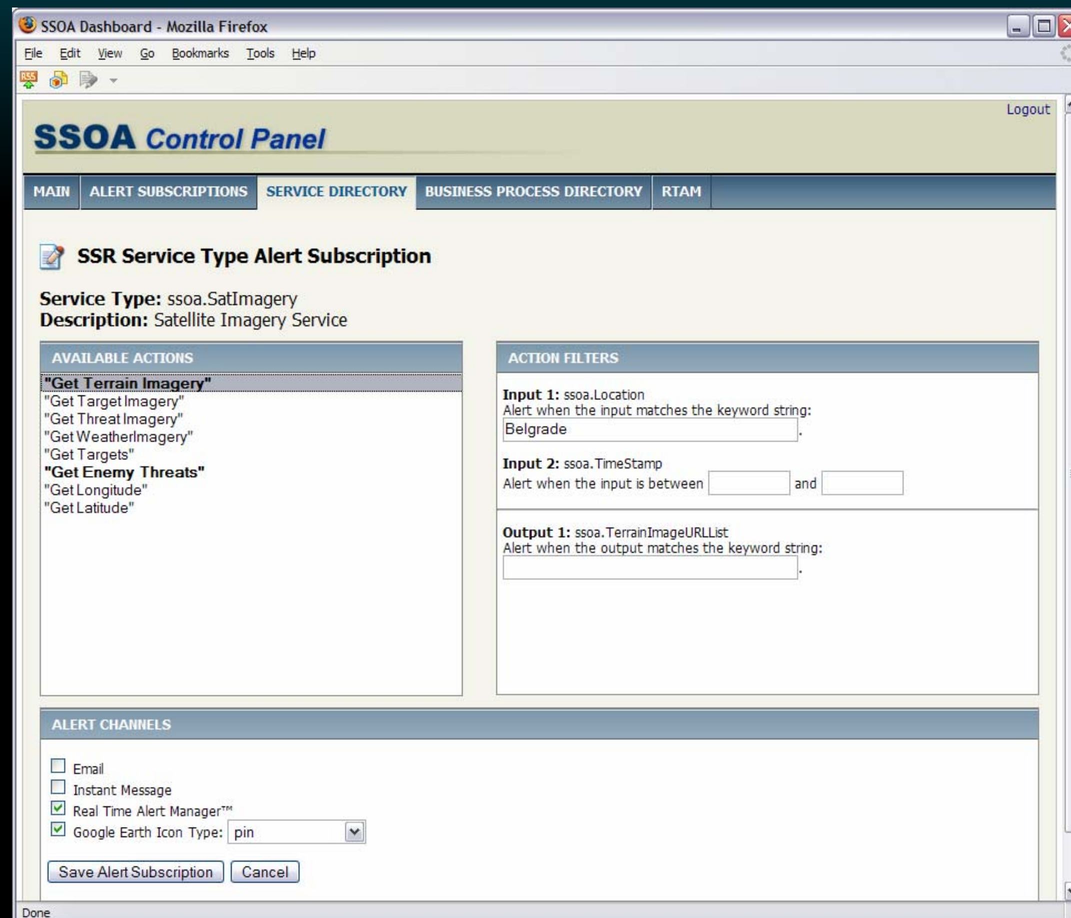
Service Discovery and Management

- EAS registers with a lookup service to receive notifications of service status
- These notifications are used to alert interested users (i.e. Preferred Entity Extraction Service A is down), or to all agents to failover to backup services during process execution

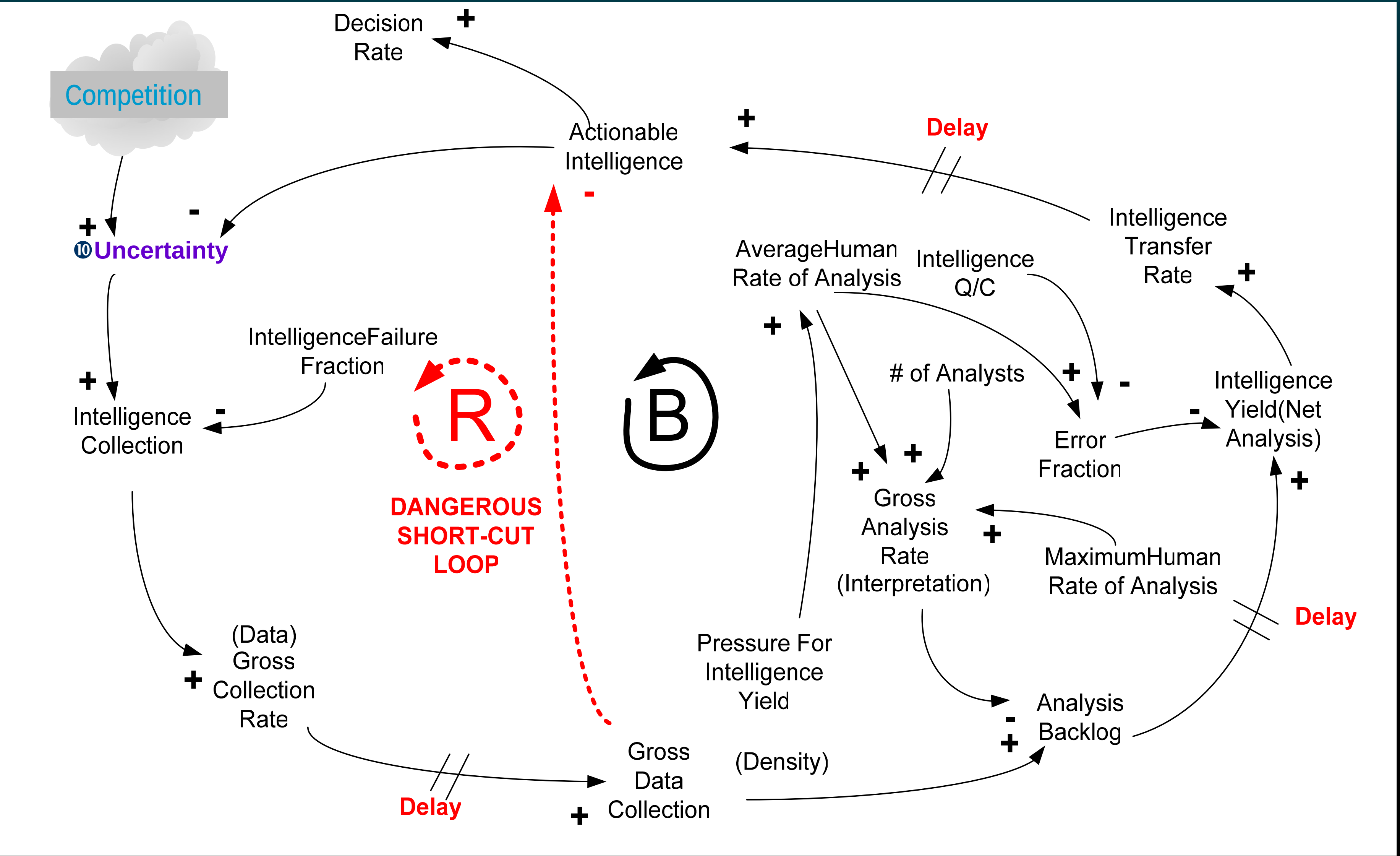


Event Subscription

- Service descriptions are used to display dynamically generated subscription screens
- Available Actions are based on the particular service the user has selected for subscription setup
- Channel selection includes email, instant messenger, RTAM, and Google Earth for any services that provide LAT/LONG information



Conclusions



Cognitive Assistant that Learns & Organizes

- DARPA IPTO funded program
- Personal office assistant, tasked with:
  - **Noticing** things in the cyber and physical environments
  - **Aggregating** what it notices, thinks, and does
  - **Executing, adding/deleting, suspending/resuming** tasks
  - **Planning** to achieve abstract objectives
  - **Anticipating** things it may be called upon to do or respond to
  - **Interacting** with the user
  - **Adapting** its behavior in response to past experience, user guidance
- 22 participating organizations

CALO & InferenceWeb Slides courtesy Dr. Deborah L. McGuinness, Stanford Knowledge Systems, AI Laboratories

Working with a Cognitive Assistant

- CALO users need to
  - **Understand** system behavior and responses
  - **Trust** system reasoning and actions
- To believe and act on recommendations from CALO, users need ways of exploring how and why the system acted, responded, recommended, and reasoned the way it did.
- Additional wrinkle: CALO knowledge, behavior, and assumptions are constantly changing through several forms of machine learning.

A unified framework for explaining behavior and reasoning is essential for users to trust and adopt cognitive assistants.

Motivating Scenario: buying a laptop

1. GetQuotes
   - Process requires 3 quotes from 3 different sources
2. GetApproval
   - Precondition: 3 valid quotes already obtained
   - Completion: approval form signed by an authorized approval representative
3. SendOrderToPurchasing
   - Precondition: signed approval form
   - Completion: order send to purchasing

Getting an Explanation

Initial request and answer strategy

- **<user>:** Why are you doing <subtask>?
- **<system>:** I am trying to do <high-level-task> and <subtask> is one subgoal in the process.

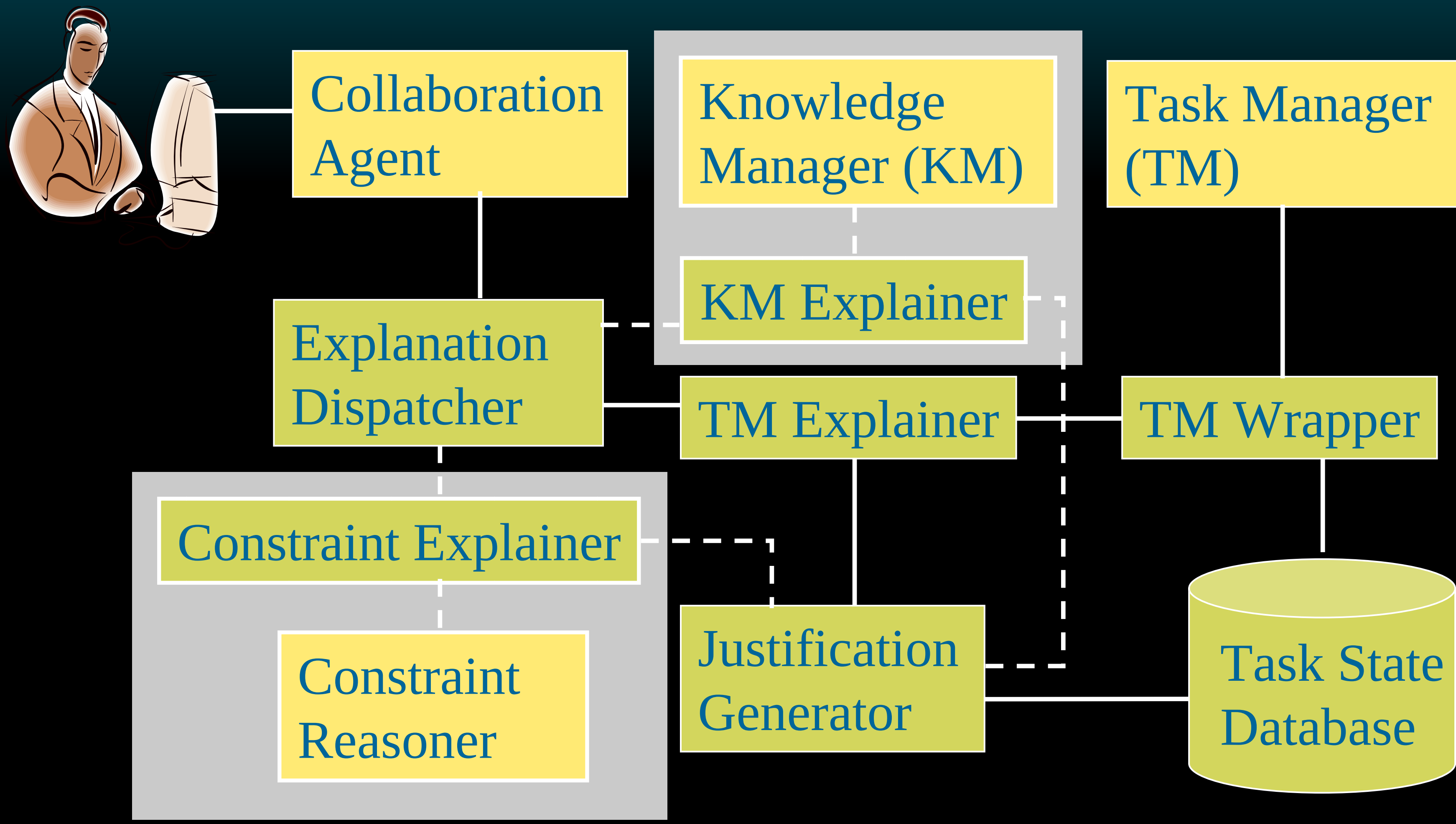
Follow-up questions for mixed initiative dialogue

- **<user>:** Why are you doing <high-level-task>?
- **<user>:** Why haven't you completed <subtask> yet?
- **<user>:** Why is <subtask> a subgoal of <high-level-task>?
- **<user>:** When will you finish <subtask>?
- **<user>:** What sources did you use to do <subtask>?

The Integrated Cognitive Explanation Environment (ICEE): System Goals

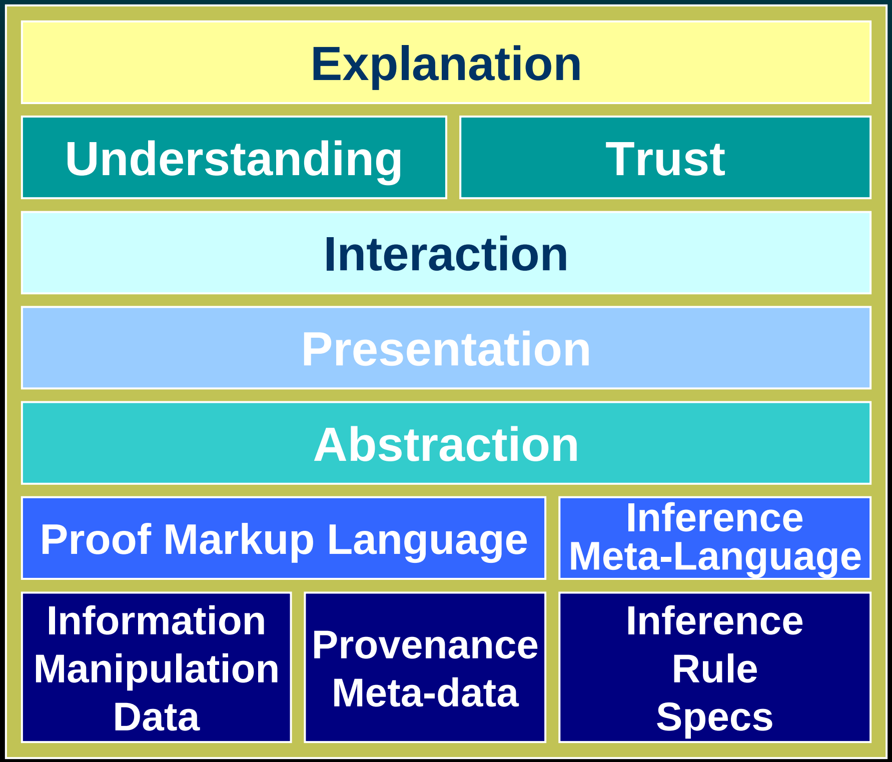
- Unified framework for explaining logical and task reasoning.
- Applicable to multiple task execution systems.
- Leverage existing InferenceWeb work for generating formal justifications.
- Underlying task reasoning useful beyond explanation.
- Provide sample implementation of end-to-end system.

ICEE Architecture



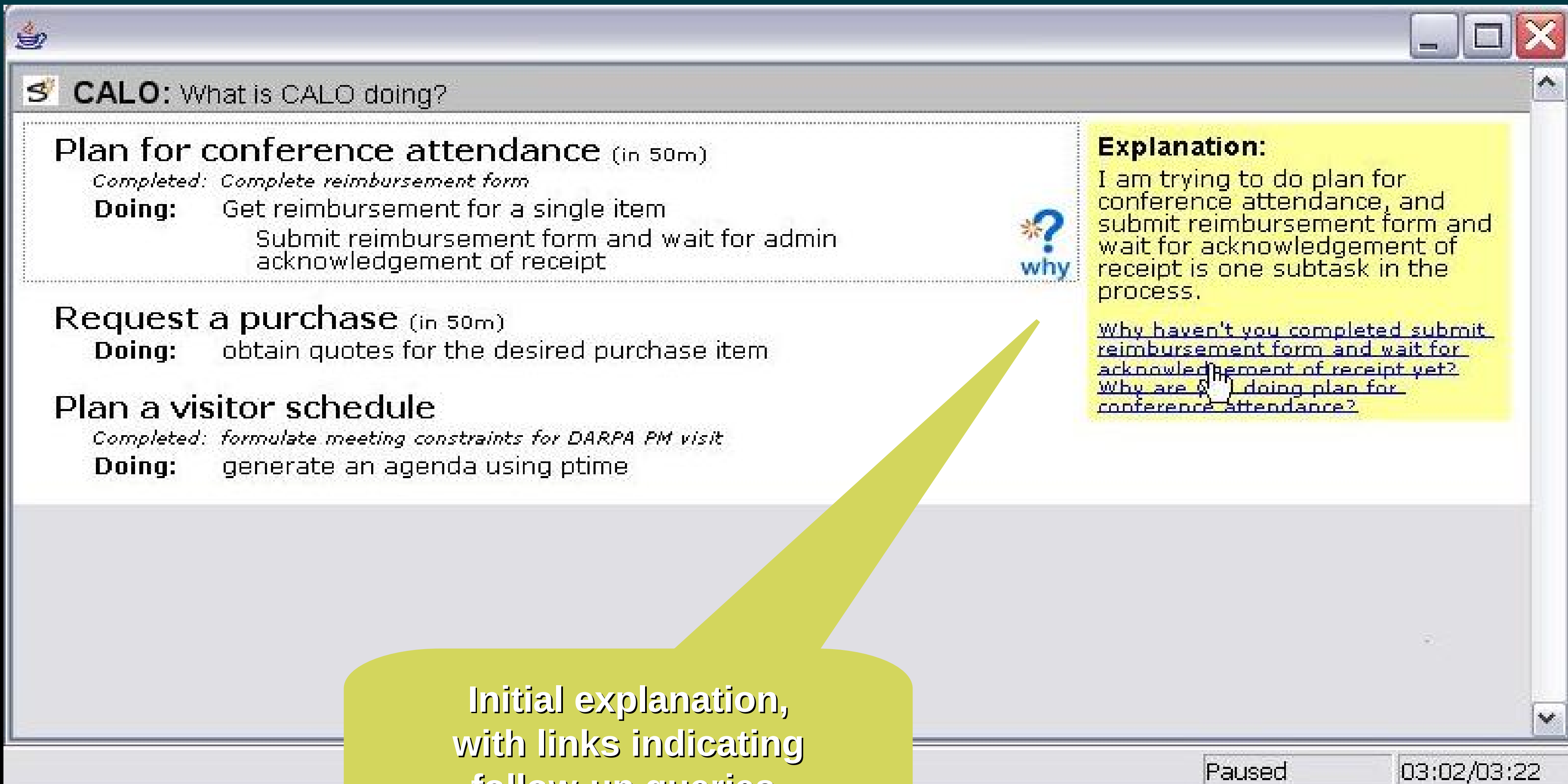
An InferenceWeb Primer

1. Registry and service support for knowledge provenance.
2. Language for encoding hybrid, distributed proof fragments (both formal and informal).
3. Declarative inference rule representation for checking proofs.
4. Multiple strategies for proof abstraction, presentation, and interaction.



Framework for *explaining* reasoning and execution tasks by abstracting, storing, exchanging, combining, annotating, filtering, comparing, and rendering justifications from varied cognitive reasoners.

Sample Interface Linked to ICEE



Initial explanation,
with links indicating
follow-up queries
and alternate strategies.

Advantages to ICEE Approach

- Unified framework for explaining task execution and deductive reasoning exploiting semantic web technologies.
- Architecture for reuse among many task execution systems.
- Introspective predicates and software wrapper that extract explanation-relevant information from task reasoner.
- Reusable action schema for representing task reasoning.
- A version of InferenceWeb for generating formal justifications.

Trust & Understanding

- If users (humans and agents) are to use, reuse, and integrate system answers, they must trust them.
- System transparency supports understanding and trust.
- Even simple "lookup" systems benefit from providing information about their sources.
- Systems that manipulate information (with sound deduction or potentially unsound heuristics) benefit from providing information about their manipulations.

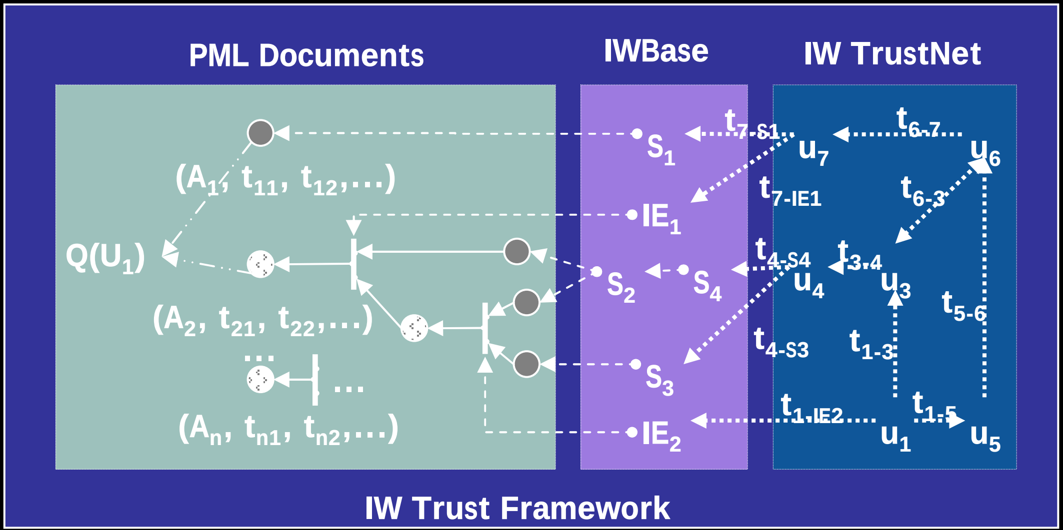
Goal: Provide interoperable infrastructure that supports explanations of sources, assumptions, and answers as an enabler for trust.

Explanations, Proof Analysis

- Framework for explaining question answering tasks
  - Stores and manages meta-information about proofs and explanations through a distributed repository (IWBase)
  - Uses the Proof Markup Language (PML) for proof interchange (OWL-based)
  - Provides registry services for proof generation and checking
- Services include proof and explanation analysis, comparison, annotation, abstraction, filtering, rendering and other capabilities
- Integrated browsing and display of PML documents from diverse sources
- Rewriting capabilities for improved understanding
- Multi-modal dialogue options including alternative strategies for presenting explanations, visualizations, and summaries

Traceability, Provenance & Management

- Inference Web uses PML documents to provide justifications
- **IWBase** contains information about sources and question answering components.
- **IWTrust** extends the Inference Web to support trust computation
  - **IW TrustNet** is a social network of source recommenders
  - A trust component implementing an algorithm to compute trust values for answers
- Trust values are used to rank answers and answer justifications
- User U_1 trusts U_3 to a degree $t_{1\text{-}3}$



Questions & Discussion

Acronym Soup

- **AD PTF** – OMG Analysis & Design Task Force
- **BMI DTF** – OMG Business Modeling & Integration Domain Task Force
- **BPEL** – Business Process Execution Language (OASIS), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- **BPEL4WS** – Business Process Execution Language for Web Services
- **CL** – ISO 24707 Common Logic: a family of first order logic languages, including Conceptual Graphs & Common Logic Interchange Format – a successor to the Knowledge Interchange Format (KIF), http://cl.tamu.edu/
- **DAML** – DARPA Agent Mark-up Language, one of the primary languages leading to the development of OWL, http://www.daml.org/
- **DAML-S** – Services ontology for DAML, http://www.daml.org/services/
- **DARPA** – Defense Advanced Research Projects Agency, http://www.darpa.mil/
- **DL** – Description Logics: a subset of first order logic, for which tractable & complete reasoning systems are available
- **ER** – Entity Relationship modeling
- **IMM** - Information Management Metamodel (a.k.a CWM2)
- **MDA** – Model-Driven Architecture, http://www.omg.org/mda/
- **MMF** - Metamodel Management Framework (ISO 19763)
- **ODM** – Ontology Definition Metamodel

More Acronym Soup

- **OWL** – W3C Web Ontology Language, a formal W3C Recommendation as of 10 February 2004, http://www.w3.org/TR/owl-semantics/
- **OWL DL** – the normative description logics dialect of OWL
- **OWL Full** – the normative OWL dialect that has increased expressivity over OWL DL, but does not conform to DL reasoning requirements
- **OWL-S** – a set of OWL ontology components that extend the W3C OWL specifications to support Semantic Web Services, http://www.daml.org/services/
- **PRR** – Production Rules Representation
- **QVT** – MOF Query / View / Transformations Specification, http://www.omg.org/docs/ptc/05-11-01.pdf
- **RIF** – Rule Interchange Format, http://www.w3.org/2005/rules/wg
- **RDF** – Resource Description Framework, http://www.w3.org/TR/rdf-concepts/
- **SBVR** – Semantics for Business Vocabularies and Rules
- **SOA** – Service Oriented Architecture
- **SOAP** – Simple Object Access Protocol, http://www.w3.org/TR/soap/
- **SWSF** – Semantic Web Services Framework, http://www.w3.org/Submission/SWSF/
- **TM** – ISO 13520 Topic Maps, http://www.isotopicmaps.org/sam/sam-model/
- **WSDL** – Web Services Description Language