# Semantic Web Technologies and Applications

Dr P Sreenivasa Kumar

Professor, CS&E Department

I I T Madras, Chennai

INDIA

# Web and Semantic Web

- Web of Documents
  - HTML, Hyperlinks
  - Search engines
    - Keyword based access
  - Meant for Humans
- Web of Data (aka the Semantic Web)
  - RDF, RDFS, OWL, Labeled-Links
  - Querying, Inferences
  - Meant for Software Agents

# Web and Semantic Web

- Web of Documents
  - Manual navigation
  - Manual information assimilation
  - Tedious to perform web-based tasks
- Web of Data (aka the Semantic Web)
  - Agent programs find resources
    - Process information from resources
    - Perform tasks on behalf of humans
  - A new way of data integration / smart processing

# Travel Arrangements

Want to travel to Mumbai for a meeting at 11am at IIT Bombay on Jan 21st. Return after Lunch

No suitable flight tickets on 21st. Shall I book night stay at Mumbai?

OK, try IIT Bombay Guest House

Mail from IITB GH manager
E-tkts to Mumbai (20th evening)
To Chennai (21st afternoon)
Taxi-Cab bookings for transfers

Agent "understands" terms from travel domain

# The Foundations

- XML
  - Extensible Markup Language
    - Framework for creating markup languages
  - Domain dependent tags
    - Unlike HTML's fixed set of tags
  - Data is not just for rendering in browsers
    - Data becomes self-describing
    - Programs can make use of data
  - A step towards adding semantics to data
  - Generalizes relational, object models

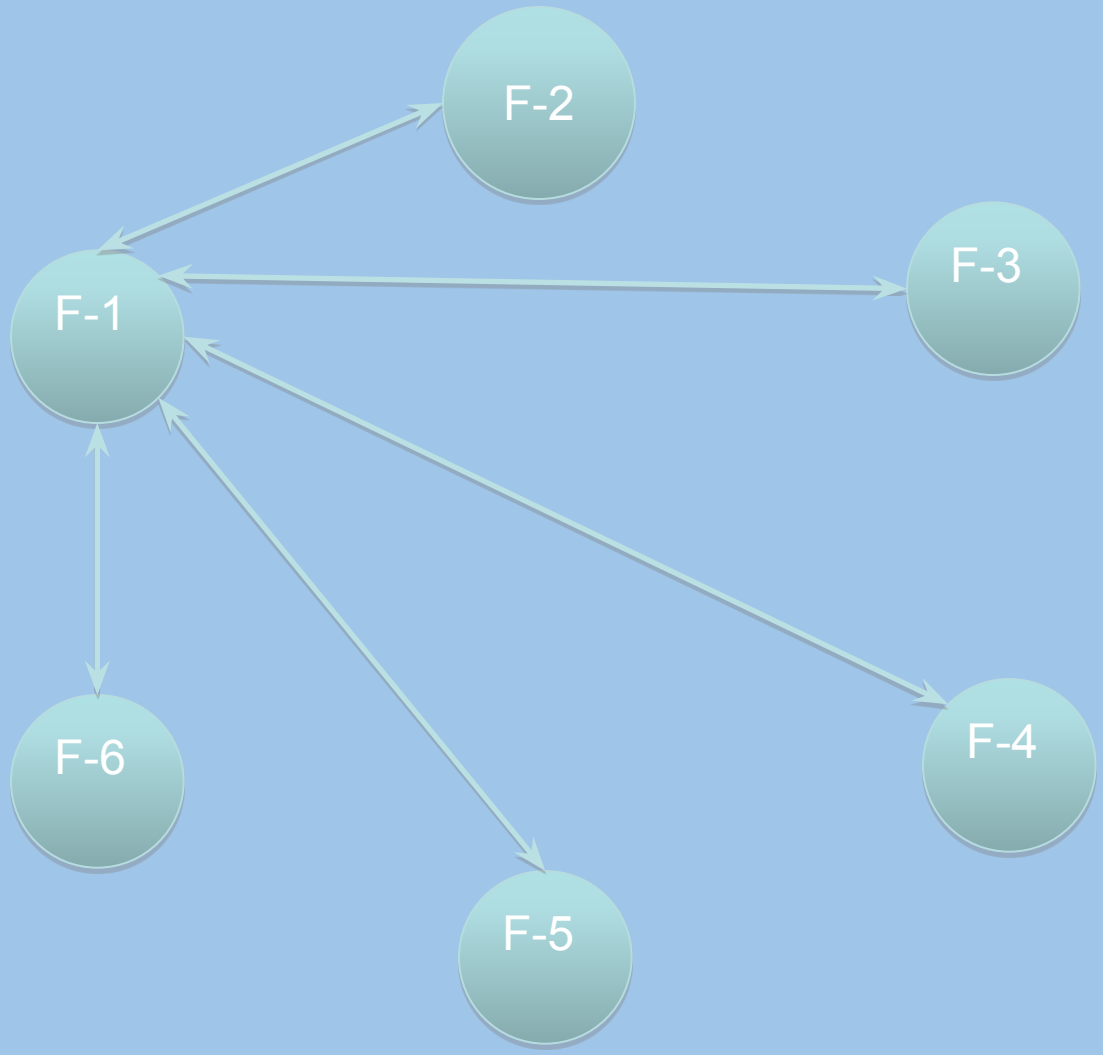
# Typical XML Data

```
<institute>
    <name>Indian Institute of Technology Madras </>
    <department><name>Mathematics</name>
                    <address> HSB321</address>
                <phone>8510</phone>
                <head>M T Nair</head>
    </department>
    <department> <name>Computer Science and Engineering</name>
                <address>  <bldg>CSB</bldg>  <room>101</room>
                    </address>
                 <phone>4350</phone>  <phone>4351</phone>
                 <head><firstName>Krishna</firstName>
                          <lastName>Sivalingam</lastName>
                 </head>
    </department>    …
</institute>
```

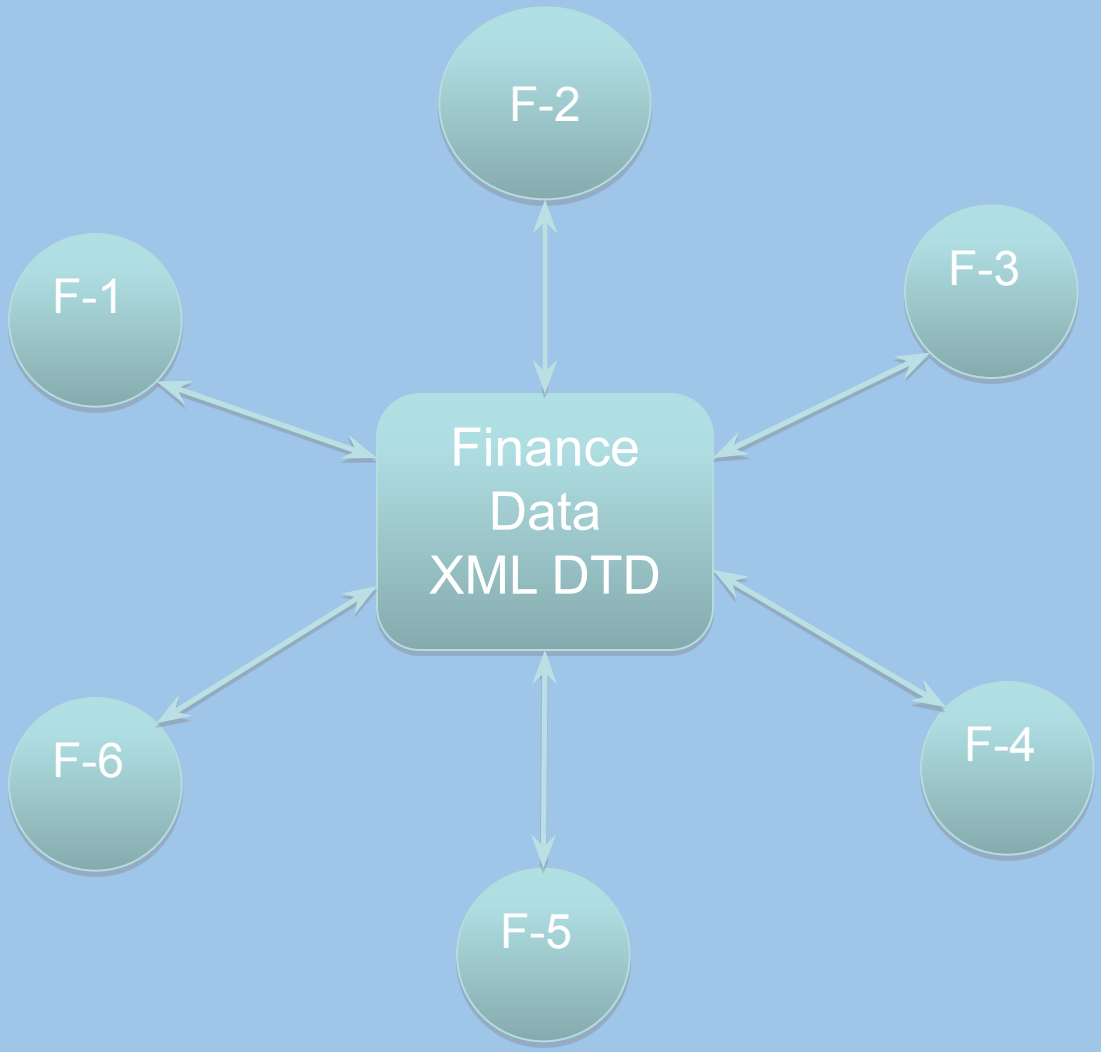
# Benefits of XML

- Creating standard formats for domains
  - Markup languages
  - Easy exchange of data – platform independent
  - Machine process-able and human readable
  - Promotes interoperability
    - Organizations/individuals develop domain models
    - More than a hundred markup languages created
      - XHTML, BizTalk, ebXML, ChemML, MathML, VoiceXML
      - http://en.wikipedia.org/wiki/List_of_XML_markup_languages
  - Web Services
  - Orchestrated Data Exchange

# Data Exchange



# Data Exchange Simplified



# Ambiguous Meaning

```
<department>
   <name>Mathematics</name>
   <institute>Indian Institute of Technology Madras</institute>
   <address>HSB321</address>
   <phone>8510</phone>
   …
</department>
```

Is Department part of Institute ? or

Is Institute part of Department ?

Structure alone can not convey meaning….

# Semantic Web

- Basic items represented
  - Things of interest in the domain
  - Physical as well as abstract entities
    - Persons, Books, Cities, Loans, Reservations
  - Relations between entities
    - Domain-specific: Child-of, Parent-of (Family Domain)
    - Taxonomical: surgeon-doctor; others…
- Globally unique names
  - Use URIs as names for things/relations

# Web of Data Foundations - RDF

- RDF - Resource Description Framework
  - A resource is any entity of interest
    - Individual entities, entity types, binary relations …
    - Represented by URIs
- Description
  - Set of triples - each triple is a statement
    - (subject, predicate, object) - (iit-m, located-In, Chennai)
    - or (resource, property, value) - (iit-m, founded-in, 1959)
  - Conceptually, a graph
    - Nodes – resources, Directed edges – relations
  - Framework for building semantic models

# Typical RDF Triples

…

http://www.iitm.ac.in rdf:type acad:Institute

http://www.cse.iitm.ac.in rdf:type acad:Department

Classes

acad:partOf rdf:type rdf:Property

acad:partOf rdfs:domain acad:AdministrativeUnit

acad:partOf rdfs:range acad:AdministrativeUnit

Individuals

http://www.cse.iitm.ac.in acad:partOf http://www.iitm.ac.in

http://www.iitm.TGH.ac.in acad:partOf http://www.iitm.ac.in

acad:isHeadOf rdf:type rdf:Property

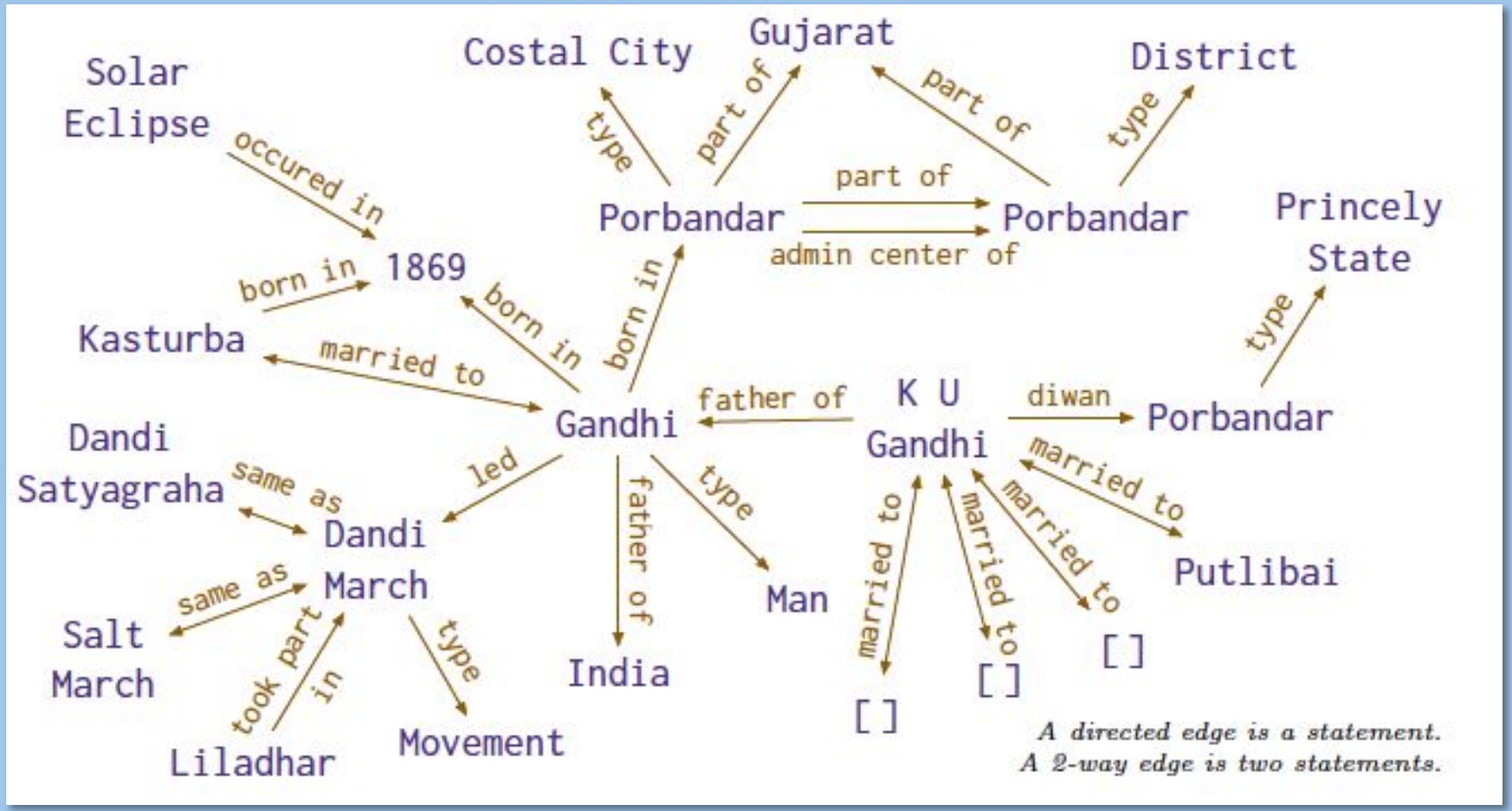
acad:isHeadOf rdfs:domain acad:Professor

acad:hasHead rdfs:inverseOf acad:isHeadOf

…

# An example RDF graph..

Baskaran 2017



# Web of Data Foundations

- RDFS (resource description framework schema)
  - *rdfs:subClassOf, rdfs:subPropertyOf*
  - *rdfs:domain, rdfs:range*
- OWL (web ontology language)
  - *owl:equivalentClass owl:sameAs*
  - *owl:inverseOf owl:symmetricProperty*
  - *owl:allValuesFrom*
  - *owl:someValuesFrom*
  - *owl:functionalProperty*
  - *etc..*

# Modeling Issues

- Old wine in a new bottle?
- Classes, properties etc
  - OO modeling, programming
- Important differences
  - Closed vs. Open world assumption
  - Set of properties
    - Does *not* determine membership in a class
    - Does *not* characterize a class
  - Properties are first-class citizens
    - Can be defined *independent* of a class

# Closed vs Open World Assumption

- Databases – CWA
  - Only explicitly mentioned statements are true
  - All other statements are false
- ‘Web of Data’ context – OWA
  - Explicitly mentioned statements are true
  - Other statements are unknown
    - Can not be taken as false
    - New information can always come up

# Classes or Concepts

- Class or Concept
  - Set of individual resources
    - Faculty, AcademicStaffMember, AdministrativeUnit
  - A class can be subset of another class
    - Faculty *subClassOf* AcademicStaffMember
- Class membership
  - Has to be explicitly stated
  - Can be inferred due to
    - *subClassOf* relations or
    - Domain / Range of properties

# Properties or Relations (or predicates)

- Properties are first-class citizens
  - Defined *independent* of classes
  - Essentially binary relations
  - "having a name" is a generic property
    - persons, streets, pets, mountains, buses
  - A property *can* have one or more domain/range
- A property can be a sub-property of another
    - `isWifeOf` is a sub property of `isSpouseOf`
    - `isBrotherOf` is a sub property `isSiblingOf`

# Semantic Model

Also known as a Domain Ontology

- Set of classes/concepts
  - Class hierarchy
- Set of properties and property hierarchy
  - Object properties - *partOf*
  - Datatype properties - *hasPhone*
- Additional information about properties
  - What properties are Symmetric? Transitive? Functional?, Inverses of properties, etc
- Individuals and Relationships among them
  - Data triples – (*cse.iitm.ac.in*, *partOf*, *iitm.ac.in*)

# Academic Institute Model

- Concepts
  - Student, Faculty, Course, Grade, etc
  - Faculty ( a subclass of AcademicStaffMember)
  - AsstProfessors, AssocProfessors, Professors
    ( all *disjoint* sub-classes of Faculty) etc
- Properties
  - teaches, taughtBy, enrolls, supervises
- Individuals – URI's are used
- Written in OWL, machine process-able

# Description Logic(s)

- Underlying mathematical basis for OWL-DL
- Decidable subsets of First Order Logic
- Variable free, compact notation for
  - Concepts, properties (binary relations),
  - Property restrictions
    - E.g., Parent – A person having at least one child
  - Number restrictions
    - E.g., Any person can have at most two parents
  - Quantification
    - E.g., Persons with only girl children
  - And many more …

# Example: Family Ontology

Given:

Concepts: Person, Man, Woman; Property: hasChild

$Person \equiv Man \sqcup Woman$; $Man \sqcap Woman \equiv \bot$

---

$Parent \equiv Person \sqcap \exists hasChild.Person$

$Mother \equiv Parent \sqcap Woman$; $Father \equiv Parent \sqcap Man$

$GrandFather \equiv Man \sqcap \exists hasChild.Parent$

$GrandMother \equiv Woman \sqcap \exists hasChild.Parent$

$GrandParent \equiv GrandFather \sqcup GrandFather$

$PersonWithOnlyDaugters \equiv Parent \sqcap \forall hasChild.Woman$

$Person \sqsubseteq =2hasParent.Person$

…

# IIT Ontology – a snippet

E V Vinu, 2016

**TBox**

$$
\begin{aligned}
\text{IITStudent} &\equiv \text{Student} \sqcap \forall \text{hasAdvisor.TeachingStaff} \sqcap \\
&\quad \exists \text{hasAdvisor.Professor} \sqcap \exists \text{enrolledIn.IITProgramme} \\
\text{IIT\_MS\_Student} &\equiv \text{IITStudent} \sqcap \leq 1\,\text{hasAdvisor.TeachingStaff} \\
\text{IITPhdStudent} &\equiv \text{IITStudent} \sqcap \geq 2\,\text{hasAdvisor.TeachingStaff} \sqcap \\
&\quad \leq 1\,\text{hasAdvisor.Professor} \\
\text{Professor} &\sqsubseteq \text{TeachingStaff} \\
\text{AssistantProf} &\sqsubseteq \text{TeachingStaff} \\
\bot &\equiv \text{Professor} \sqcap \text{AssistantProf}
\end{aligned}
$$

**ABox**

IITStudent(tom)
IIT_MS_Student(tom)
hasAdvisor(tom, bob)
IITPhdStudent(sam)
hasAdvisor(sam, alice)
hasAdvisor(sam, roy)
AssistantProf(alice)

In DL notation.
Can also be written in OWL.

What is “roy” ?

# Some DL Operators

| Construct | Syntax | Interpretation |
|---|---|---|
| Concept intersection | C ⊓ D | $(C \sqcap D)^{\mathcal{I}} = (C^{\mathcal{I}} \cap D^{\mathcal{I}})$ |
| Concept union | C ⊔ D | $(C \sqcup D)^{\mathcal{I}} = (C^{\mathcal{I}} \cup D^{\mathcal{I}})$ |
| Concept negation | ¬C | $(\neg C)^{\mathcal{I}} = (\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}})$ |
| Universal restriction | ∀R.C | $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b, (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$ |
| Existential restriction | ∃R.C | $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b, (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$ |
| Minimum cardinality | ≥nR | $(\geq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$ |
| Maximum cardinality | ≤nR | $(\leq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$ |
| Concept inclusion | C ⊑ D | $(\mathrm{C} \sqsubseteq D)^{\mathcal{I}} = (\mathcal{C}^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}})$ |
| Concept equivalence | C ≡ D | $(\mathrm{C} \equiv D)^{\mathcal{I}} = (\mathcal{C}^{\mathcal{I}} = \mathcal{D}^{\mathcal{I}})$ |

# Namespaces

- Namespace - Represented by a URI
  - Terms – concept names, property names etc
  - Used in a specific context
- Same term in different contexts
  - different meanings  e.g, title: books, persons
  - different namespaces
- RDF/RDFS/OWL namespaces

– Terms – *class*, *type*, *domain, range, subclass,…*
– Used to define terms in domain ontologies
– Defined domain-terms – separate namespace

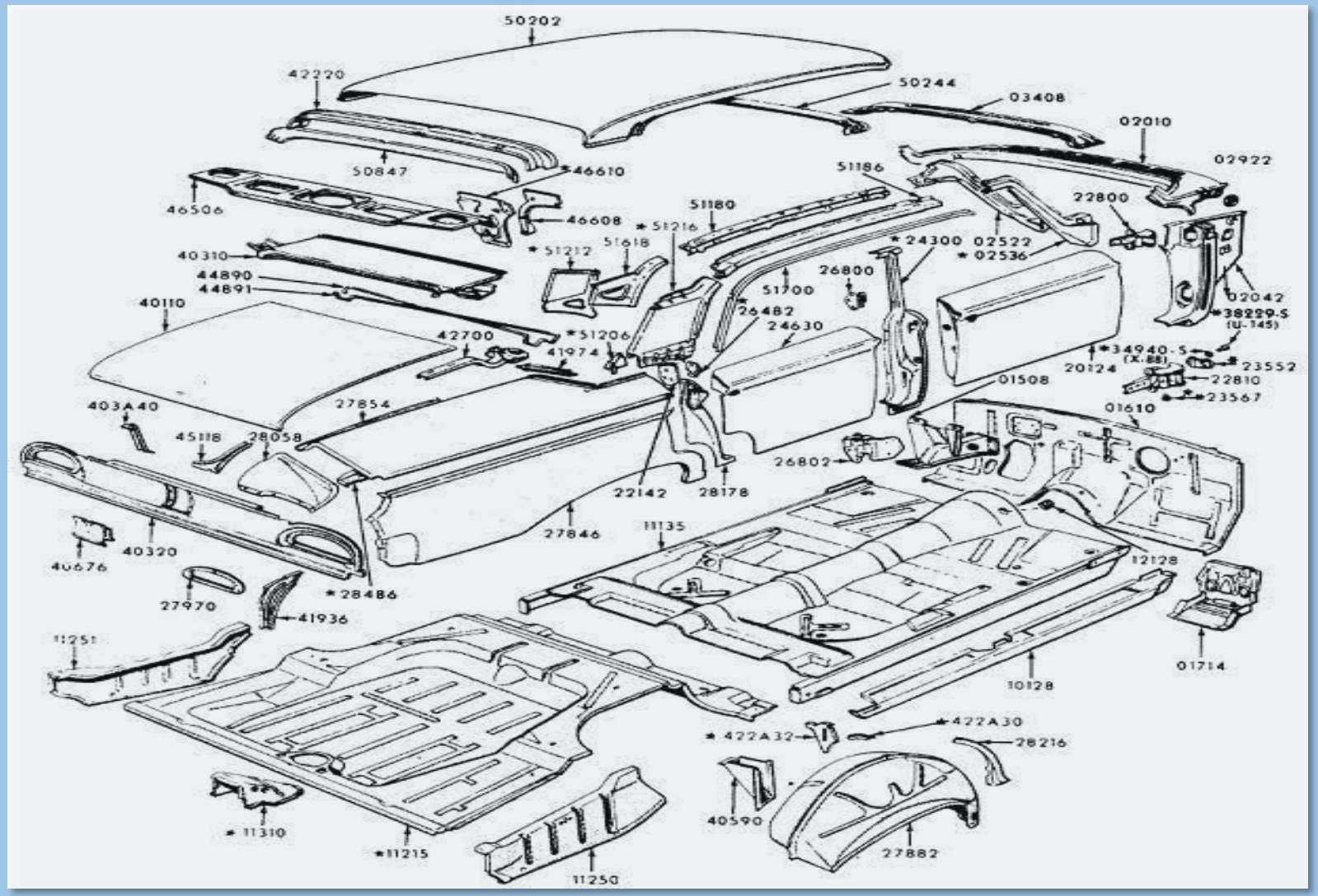
# Where exactly is the semantics?

- Semantics
  - Meaning of terms
  - Every medical surgeon is also a doctor
  - med:Surgeon rdfs:subClassOf med:Doctor
- Meanings get fixed
  - By standardizing the meanings of terms in the *ontology* frameworks
  - RDF, RDFS, OWL – *ontology* frameworks
  - Applications can interpret them in only one way

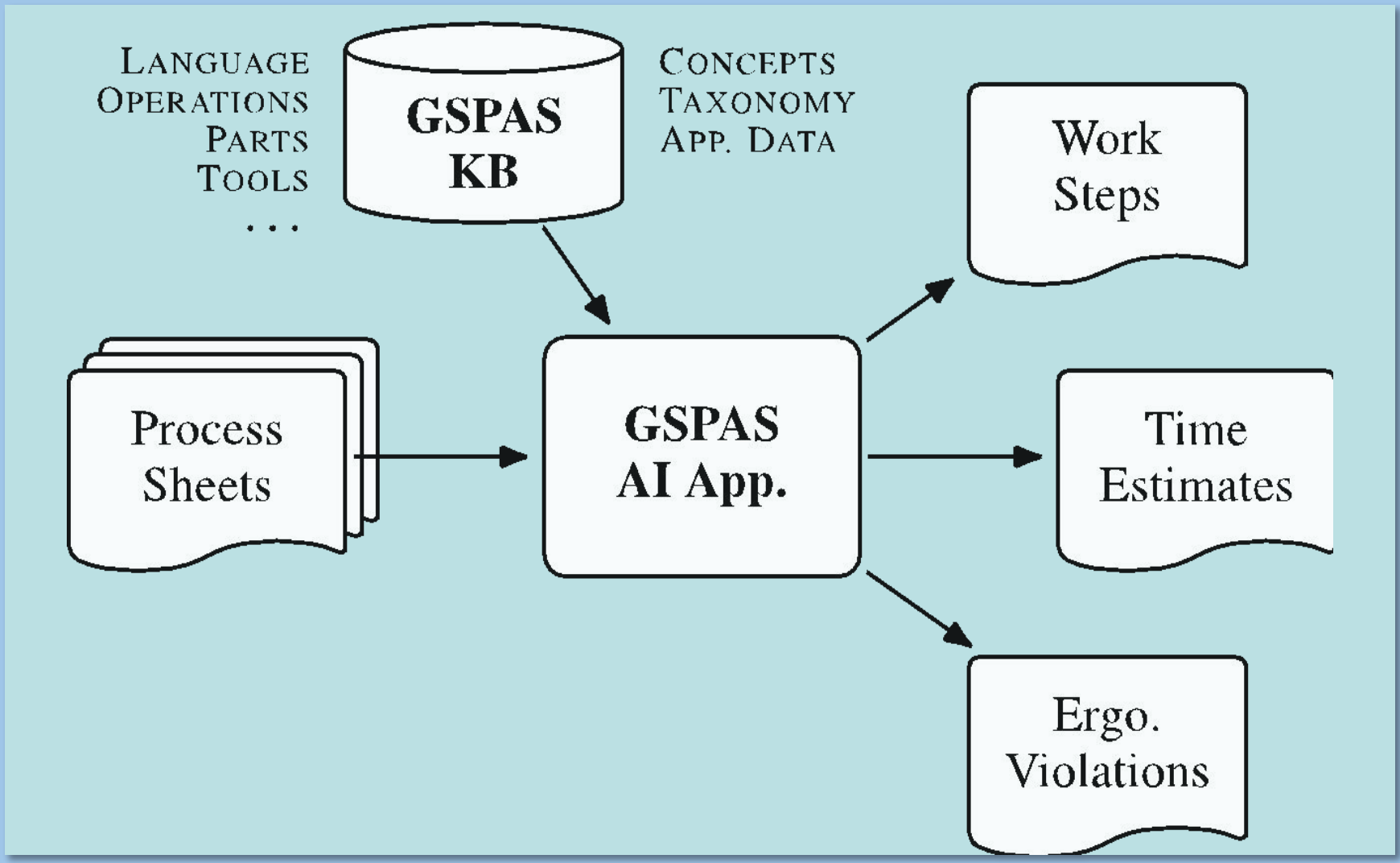
# Ontologies in Manufacturing

- GSPAS AI system of Ford Motor Company
  - **G**lobal **S**tate **P**rocess **A**llocation **S**ystem
  - Input is a set of Process Sheets
  - Process Sheet
    - High-level instructions for assembling a part
    - Written in Ford's own "*Standard Language*"
  - Employs KL-One based
    - Manufacturing Ontology
    - Lexical Ontology
  - Re-engineered into RDF/OWL by us at IITM

# Car Assembly



# GSPAS System



# FORD's GSPAS Ontology

- KL-one based ontology
  - 10,000+ concepts
  - Models knowledge about:
    - Ford's Standard Language, Vehicle Assembly
- Multiple hierarchies
  - Verbs, nouns – lexical-related
  - Operations, Parts, tools – Vehicle-related
- One single namespace for all terms
  - Homonym problem – eg 'hammer' – tool; operation
  - 'power hammer' is a 'hammer' and hence an operation!!

# An Example Process Sheet

```
TITLE: ASSEMBLE IMMERSION HEATER TO ENGINE

10 OBTAIN ENGINE BLOCK HEATER ASSEMBLY FROM STOCK
20 LOOSEN HEATER ASSEMBLY TURNSCREW USING POWER TOOL
30 APPLY GREASE TO RUBBER O-RING AND CORE OPENING
40 INSERT HEATER ASSEMBLY INTO RIGHT REAR CORE PLUG HOSE
50 ALIGN SCREW HEAD TO TOP OF HEATER

TOOL 20 1 P AAPTCA TSEQ RT ANGLE NUTRUNNER
TOOL 30 1 C COMM TSEQ GREASE BRUSH
```

- Translated into low-level, detailed shop-floor operator instructions
  - Ontology definitions of operations, parts are used

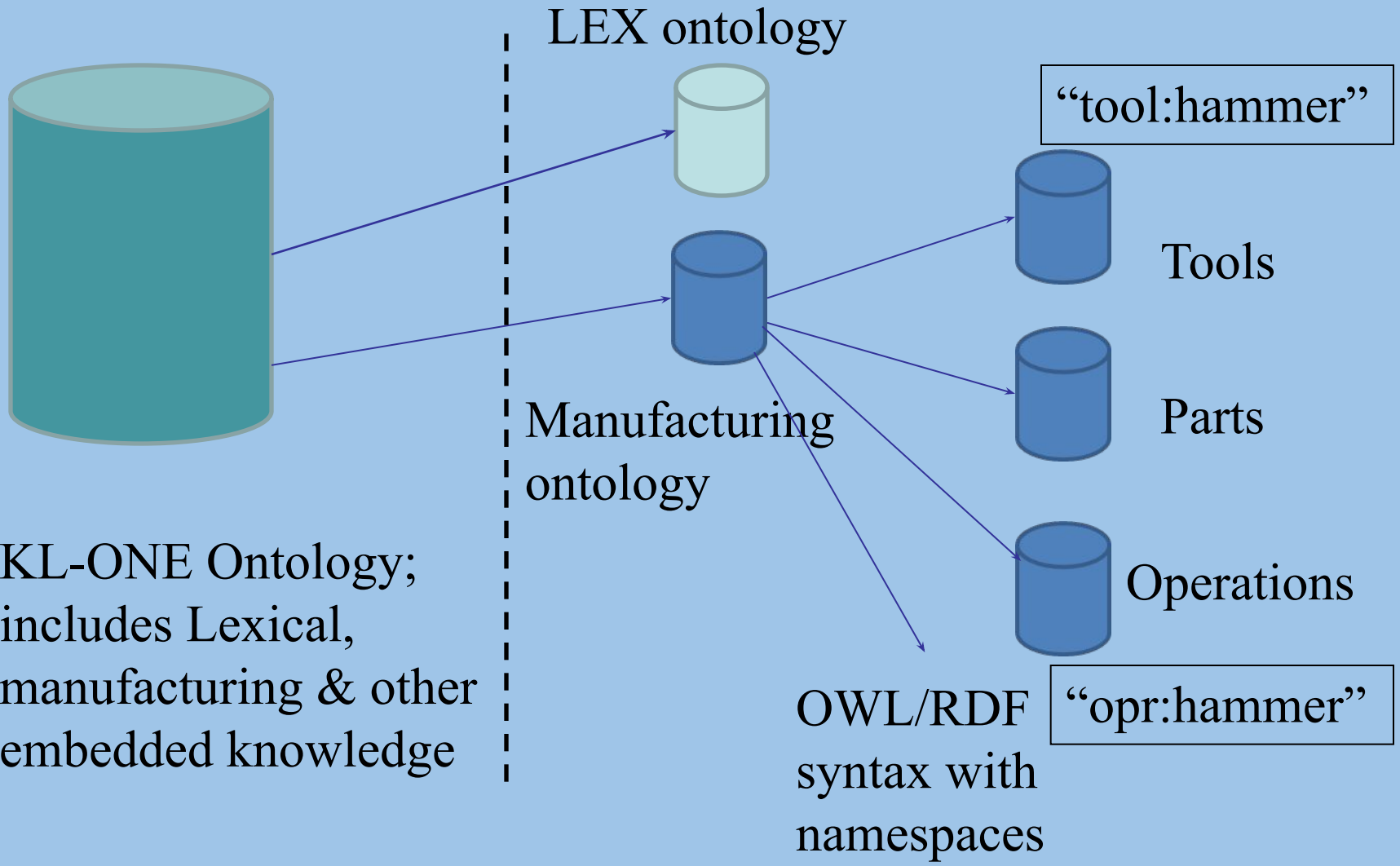
# Ergonomic Analysis

- Knowledge-based ergonomic analysis
- Ergonomic violations in a process sheet
  - Frequency violations
    - "Standard Language" verb/operation usage
    - Indicates a certain operation frequency
      - More than a certain threshold – humanly impossible
  - Heavy-part violations
    - A heavy part usage
    - Without appropriate lift/hoist mechanism
  - Inferences from verb/part name to be done

# Issues with current system

- KL-One – a very early KR framework
  - Custom-built KL-One editors are used by Ford
- KB Update Module
  - GUI based, Enables updates & error-checking
  - Difficult to maintain as system migrated …
- KB and Application : tightly coupled
  - Difficult to extract knowledge for use in other applications
- All terms in a single namespace

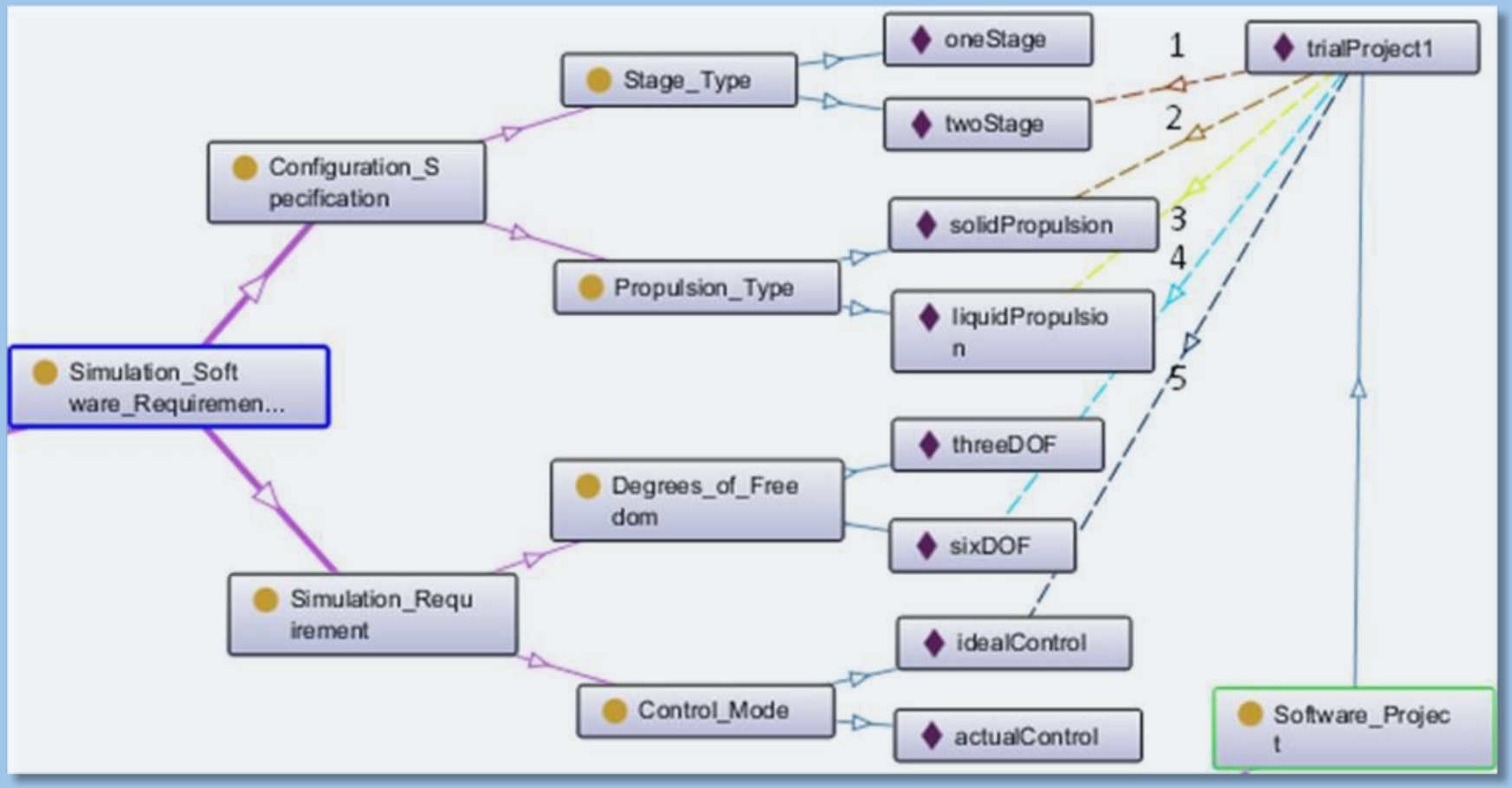
# Re-Engineered Ontology

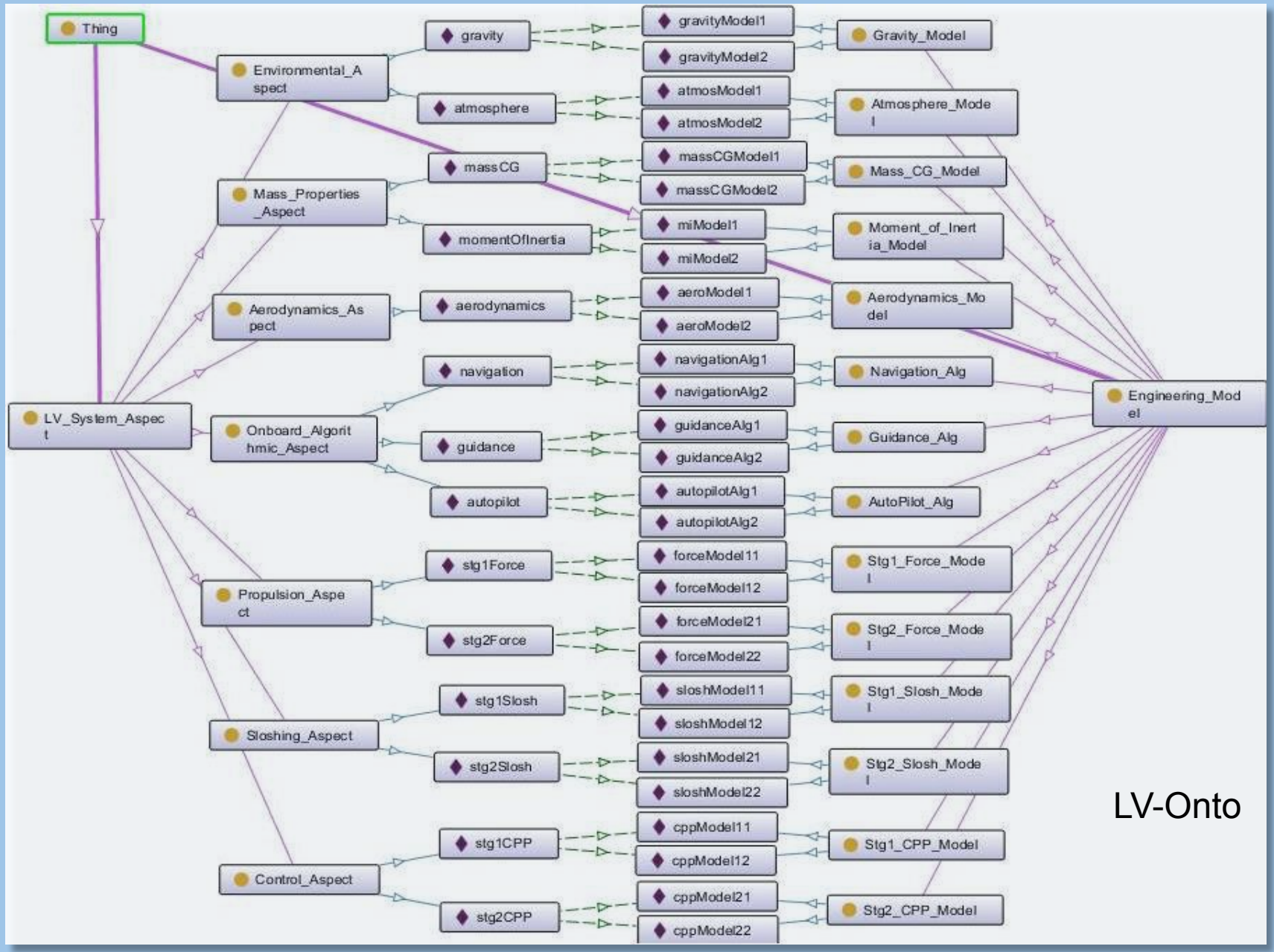


# Ontologies in S/W Engineering

- Automatic software synthesis from modules
- Simulation software for satellite launch vehicles
- Launch vehicle ontology
  - Subsystems and their inter-relationships
  - Simulation requirements
    - Gross / Fine / h/w-in-loop etc
  - Incompatible choices
  - Subsystem – applicable math models
- Agent does the synthesis
  - Given the end-user specs, produces a custom-assembled simulation s/w that meets the specs.

# Ontology LV-Onto Segment

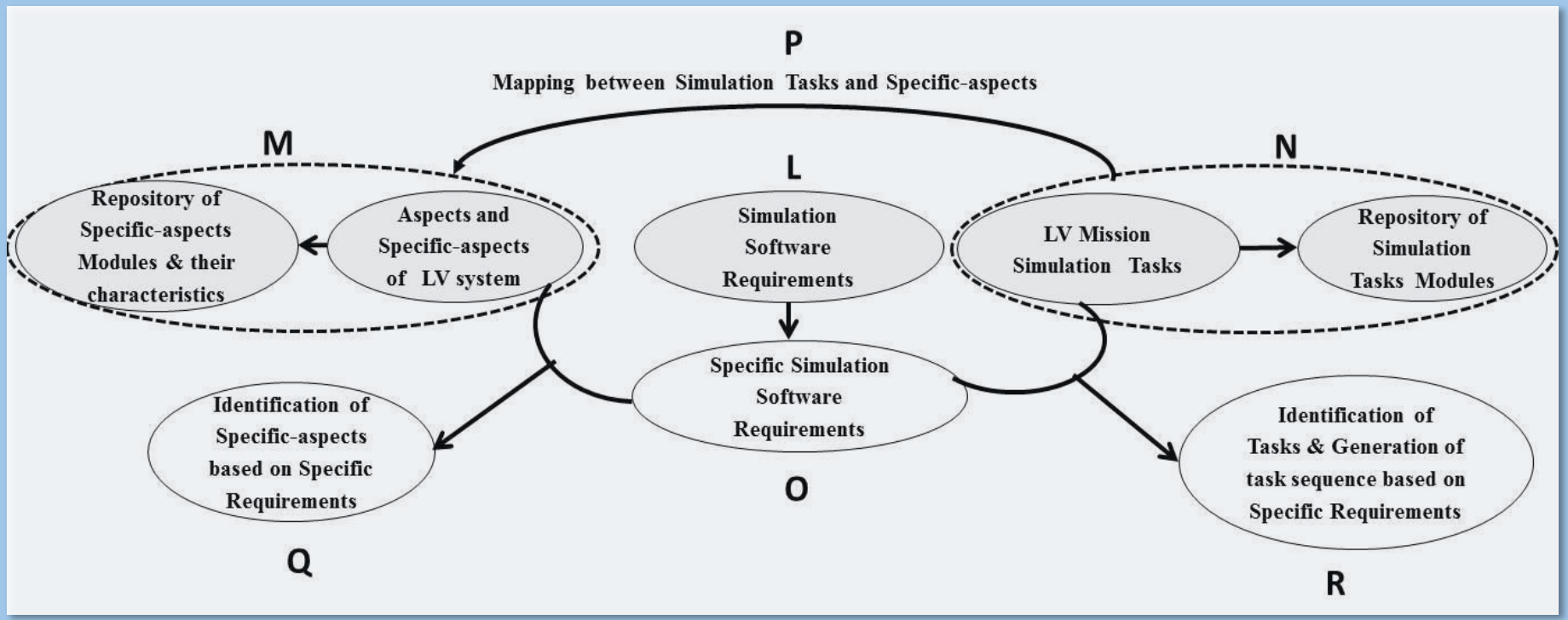



Thing
Environmental_Aspect
Mass_Properties_Aspect
Aerodynamics_Aspect
LV_System_Aspect
Onboard_Algorithmic_Aspect
Propulsion_Aspect
Sloshing_Aspect
Control_Aspect
gravity
atmosphere
massCG
momentOfInertia
aerodynamics
navigation
guidance
autopilot
stg1Force
stg2Force
stg1Slosh
stg2Slosh
stg1CPP
stg2CPP
gravityModel1
gravityModel2
atmosModel1
atmosModel2
massCGModel1
massCGModel2
miModel1
miModel2
aeroModel1
aeroModel2
navigationAlg1
navigationAlg2
guidanceAlg1
guidanceAlg2
autopilotAlg1
autopilotAlg2
forceModel11
forceModel12
forceModel21
forceModel22
sloshModel11
sloshModel12
sloshModel21
sloshModel22
cppModel11
cppModel12
cppModel21
cppModel22
Gravity_Model
Atmosphere_Model
Mass_CG_Model
Moment_of_Inertia_Model
Aerodynamics_Model
Navigation_Alg
Guidance_Alg
AutoPilot_Alg
Stg1_Force_Model
Stg2_Force_Model
Stg1_Slosh_Model
Stg2_Slosh_Model
Stg1_CPP_Model
Stg2_CPP_Model
Engineering_Model
LV-Onto

# Example Domain Knowledge

1. Whatever may be the requirements for a project, all environmental aspects need to be simulated.
2. Taking care of *sixDOF* requires more *specific-aspects* to be simulated compared to *threeDOF*.
3. Whenever a rocket stage uses *liquid propulsion*, the sloshing of the liquid stage needs to be simulated.
4. Projects only with *actual control* and *sixDOF* will require the *aspect* of Control Power Plant (CPP) for simulation.

# Knowledge Organization



# LV-Onto SWRL Rules (1/2)

R1. *Software_Project(?x), Environmental_Aspect(?y) → hasSpecificAspect(?x, ?y)*

R2. *Software_Project(?x) → hasSpecificAspect(?x, massCG)*

R3. *Software_Project(?x), Aerodynamics_Aspect(?y) → hasSpecificAspect(?x, ?y)*

R4. *Software_Project(?x) → hasSpecificAspect(?x, stg1Force)*

R5. *Software_Project(?x), hasNumberOfDOF(?x, sixDOF) → hasSpecificAspect(?x, momentOfInertia)*

R6. *Software_Project(?x), hasNumberOfDOF(?x, sixDOF), Onboard_Algorithmic_Aspect(?y) → hasSpecificAspect(?x, ?y)*

R7. *Software_Project(?x), hasStage1Type(?x, liquidPropulsion) → hasSpecificAspect(?x, stg1Slosh)*

# SWRL Rules (2/2)

R8. *Software_Project(?x), hasNumberOfDOF(?x, sixDOF), hasControlMode(?x, actualControl) → hasSpecificAspect(?x, stg1CPP)*

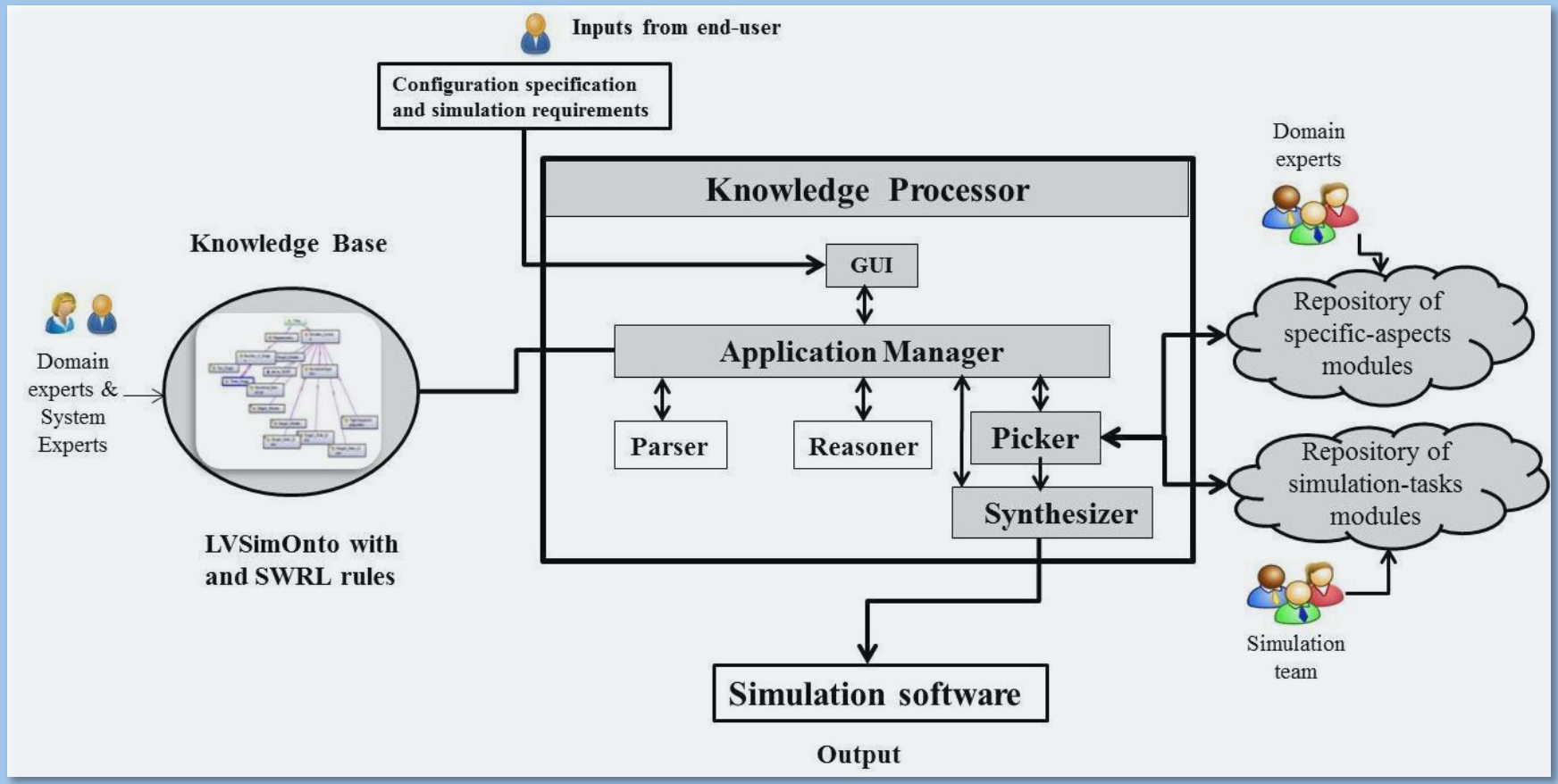
R9. *Software_Project(?x), hasNumberOfStages(?x, twoStage) → hasSpecificAspect(?x, stg2Force)*

R10. *Software_Project(?x), hasNumberOfStages(?x, twoStage), hasStage2Type(?x, liquidPropulsion) → hasSpecificAspect(?x, stg2Slosh)*

R11. *Software_Project(?x), hasNumberOfStages(?x, twoStage), hasNumberOfDOF(?x, sixDOF), hasControlMode(?x, actualControl) → hasSpecificAspect(?x, stg2CPP)*

R12. *Software_Project(?x), hasNumberOfDOF(?x, threeDOF), hasControlMode(?x, actualControl) → hasIncompatiableRequirement(?x, actualControl)*

# Overall System Architecture



# Reasoning and Querying

- Data Sources
  - Publish/Provide RDF triples
- Reasoners / Extractors
  - Derive new information
  - Establish new connections
- Query languages
  - Retrieve information - SPARQL
- Applications
  - Carry out tasks for us

# Travel Arrangements Example

- The Agent
  - Consults travel and accommodation semantic model
  - Sends booking requests to Travel Agent
  - Consults IIT-Bombay semantic model
  - Sends email request to GH in-charge
  - Consults call taxi semantic model
  - Sends booking requests for cars
  - Sends me an email about what to do!!

# References

- Semantic Web Primer (2nd edition),
  - Grigoris Antoniou and Frank van Harmelen
  - MIT Press, 2008.
- Foundations of Semantic Web Technologies
  - Markus Kroetzsch, Pascal Hitzler, and Sebastian Rudolph, CRC Press, 2010.
- http://www.w3.org/
- Protégé – Ontology editor from Stanford University

# References


- S. S. Uma Sankari, P. Sreenivasa Kumar, C. Geethaikrishnan, R. Vikraman Nair: Ontology for Launch Vehicle Mission Simulation. J. Aerosp. Inf. Syst. 14(3): 198-202 (2017)

- Nestor Rychtyckyj, Venkatesh Raman, Baskaran Sankaranarayanan, P. Sreenivasa Kumar, Deepak Khemani: Ontology Re-Engineering: A Case Study from the Automotive Industry. AI Magazine 38(1): 49-60 (2017)