

Using Ontology to bind Web Services to the Data Model of Automation Systems

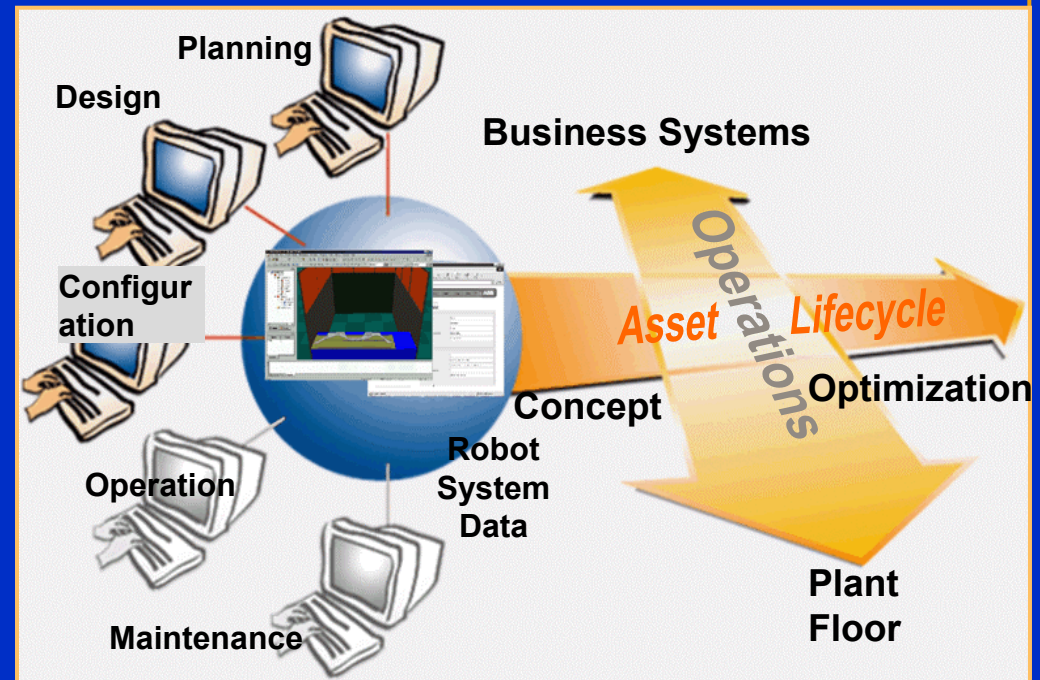
Zaijun Hu
ABB Corporate Research Center
Germany
zaijun.hu@de.abb.com

- Industrial Information System
- New Challenge
- Some Definitions
- Roles of Web Services and Ontology
- Ontology Model
- Web Services and System Architecture
- Example
- Conclusion

Industrial Information System



- Industrial Information Systems
 - Information model integration
 - Automation level
 - Life-cycle
 - Computation integration
 - Standard communication
 - Common interface
 - Component-based
 - Multiple structure
 - Functional
 - Location
 - Product



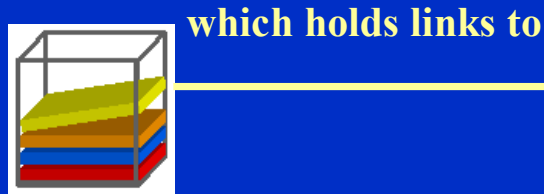
■ Properties

- Close relation among structures
- Clearly defined connection point
- Navigation inside a structure and from one to another
- Each structure has a clearly defined semantic and serves one purpose
- Well-defined designation to identify objects in a structure
- Possibility to associate the computational model

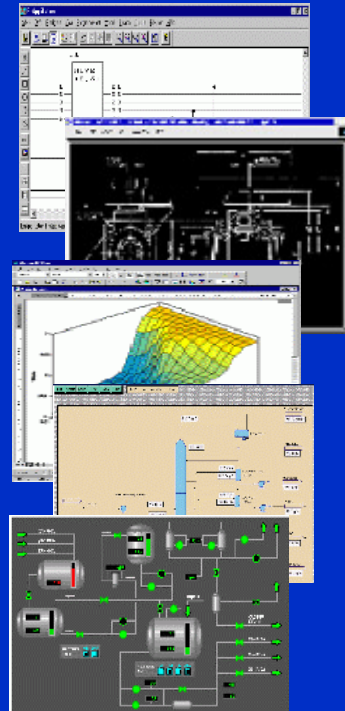
ABB AIP Platform



An Aspect Object



Aspects



Engineering Drawing

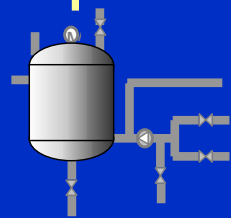
Mechanical Drawing

Quality Report

Simulation

Process Graphic

is modeled by

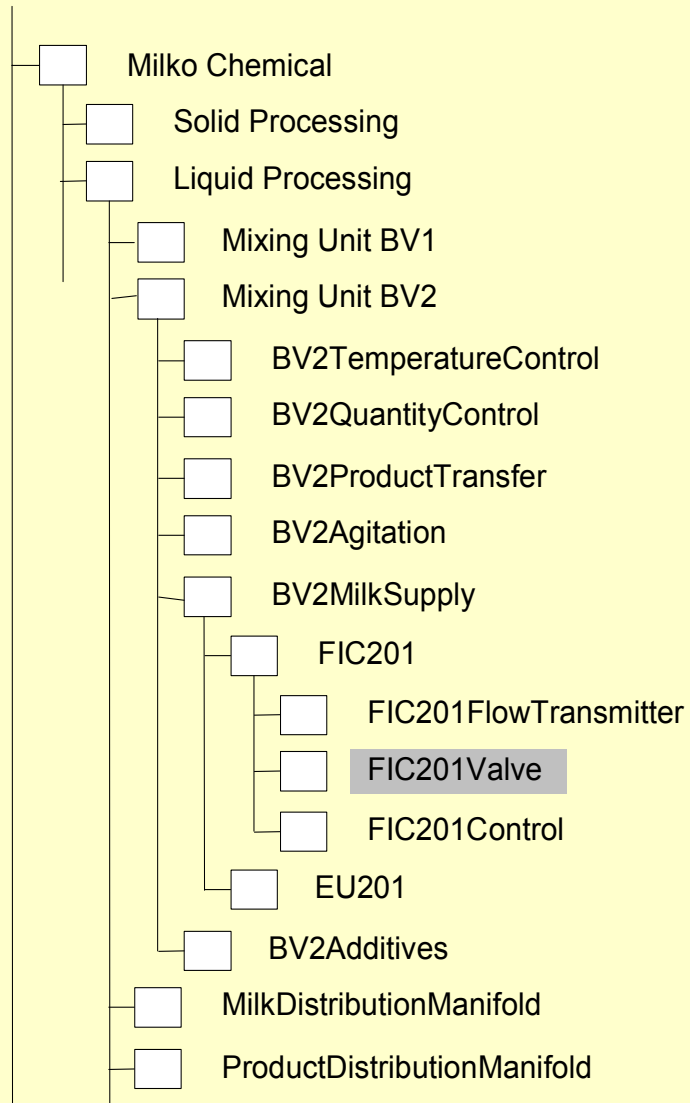


Real Object

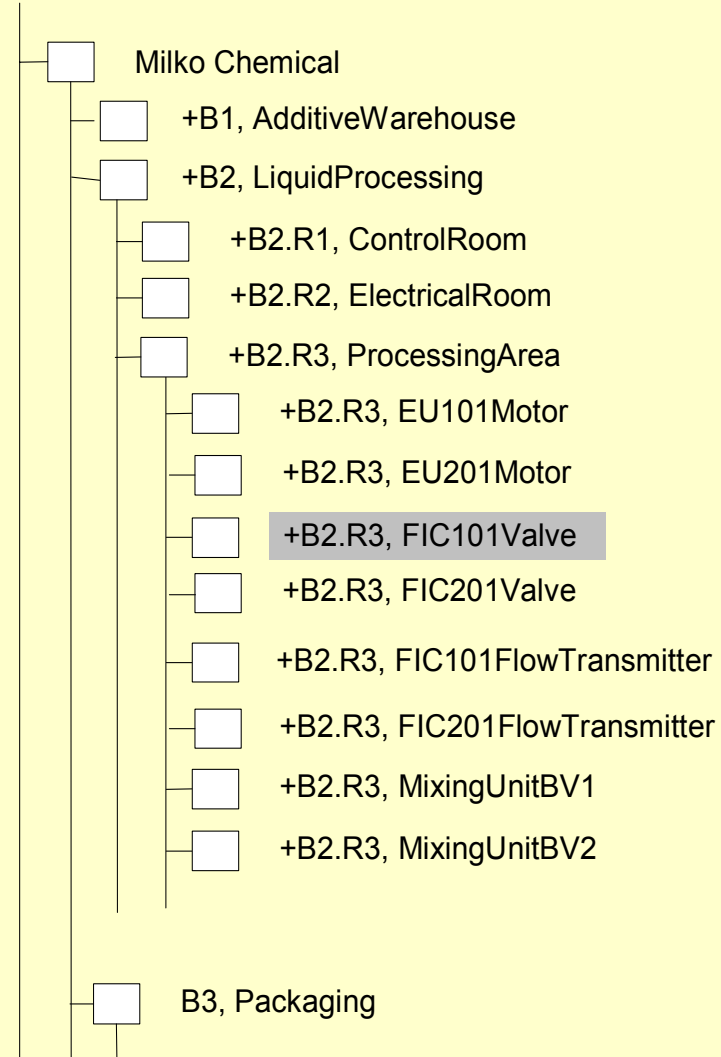
Multiple Structure



Functional Structure



Location Structure



New Challenges

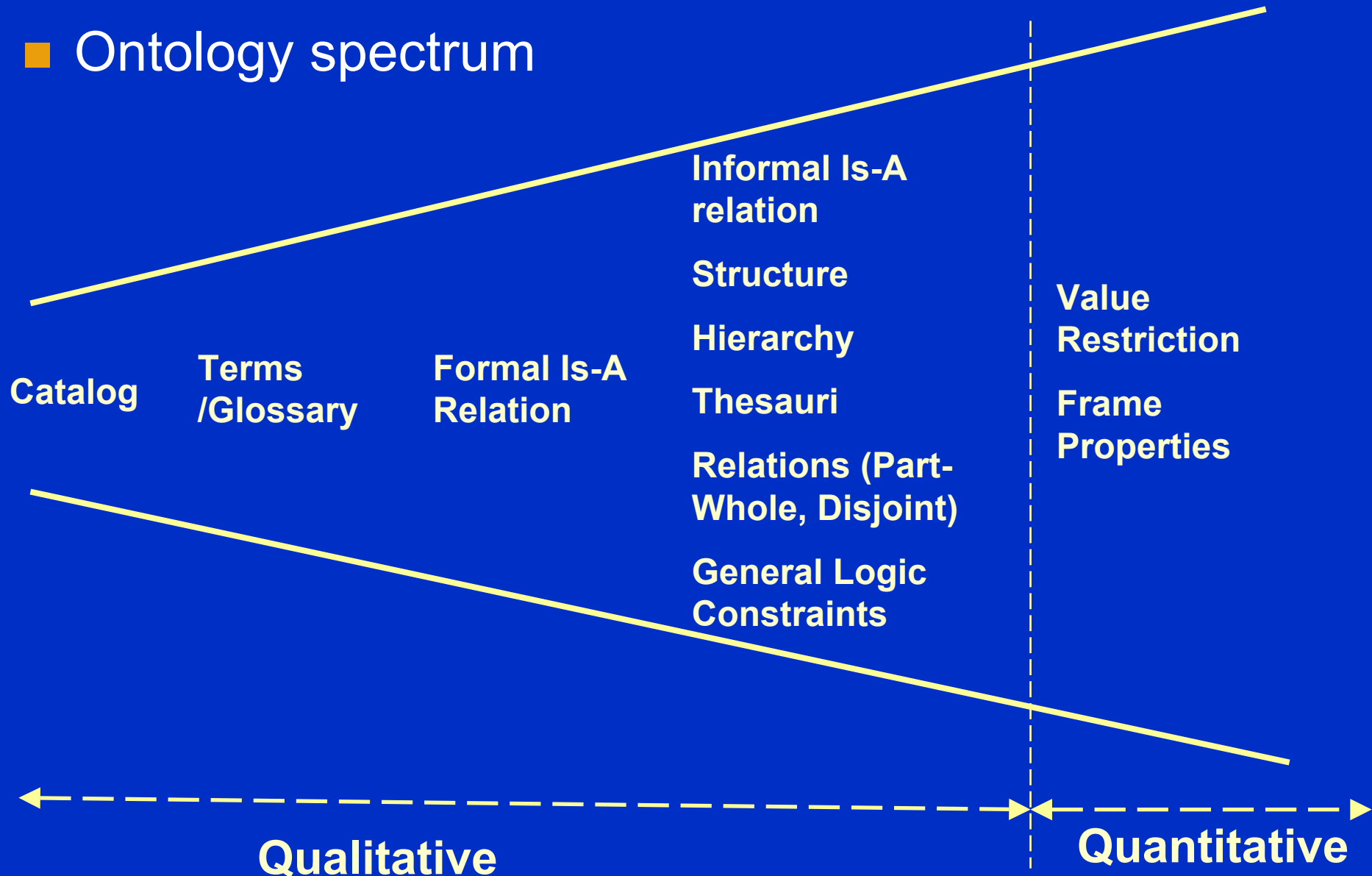


- Diverse data models with the corresponding structure
- Separation of data models from computation models
- Structuring of computation models
- Binding of computation model to data models

Ontology Definition - 1



■ Ontology spectrum



Ontology Definition - 2



- Specification of a conceptualization
 - A set of terms
 - Their relationships
 - The corresponding operations
- Examples
 - Glossary
 - Catalog
 - Taxonomy
 - Object model

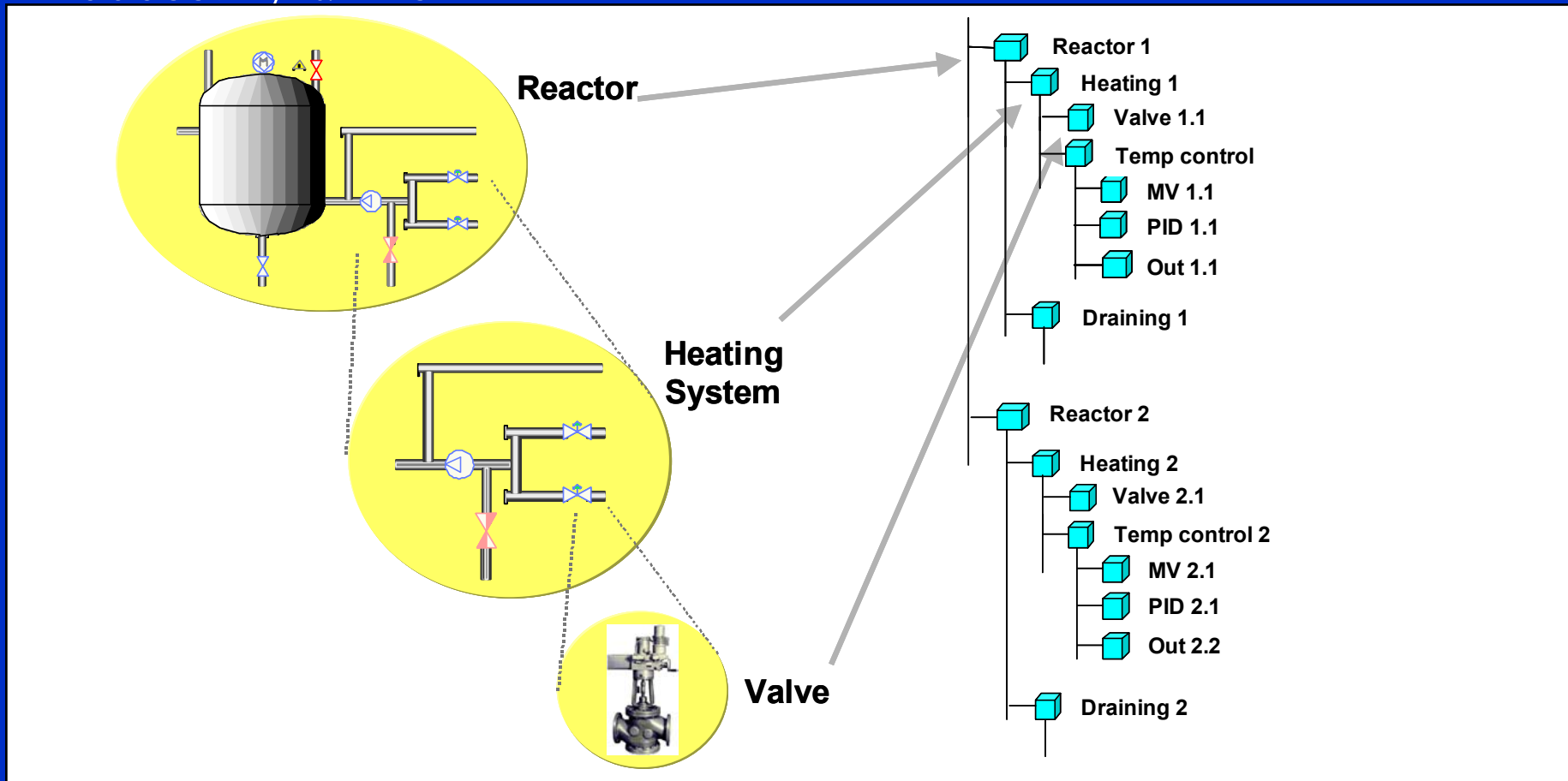
Domain and Operation Ontology



- A domain ontology is an ontology for a certain domain
 - Power generation
 - Medicine
 - Automobile
- An operation ontology is an ontology for data operations
 - Simulation
 - Mechanical drawing
 - Optimization

Data model

A data model in the context of this paper is used to describe and represent an industrial process with the related components and the control of the process with all necessary units



- A computation model defines structure, categorization and classification of data operations
 - A group of related functions
 - In form of software applications or components (web services)
 - Operation ontology can be used for structure, categorization and classification

A binding is a process that associates a computation model containing a group of functions to the corresponding data model. The binding is to find the most suitable functions that can be used to the data of interest or to provide a guideline to users for the association

■ Definition

- Platform with a set of standards
 - Discovery (UDDI)
 - Description (WSDL)
 - Communication (SOAP, HTTP)
- Reusable software components
 - Defined interfaces
 - Flexible in the implementation
 - Use via web

■ Features

- Platform independency in the use
- Neutrality in the implementation

- Role of Web Services in the context of this paper
 - Access to the real-time information
 - Change in the real-time information
 - Navigation among different structures
 - Association of application to information
 - Platform- and language-neutral wrapper for computation units

- Controlled vocabularies
 - Sharable, reusable, reduction of vocabulary
- Description of structure
 - Data model
 - Computation model
- Data model engineering
 - Intelligent structure generation
 - Intelligent structure mapping
- Binding of computation model to data model

- Basic elements
 - Terms
 - Designation of things
 - Set of meanings
- Relations
 - Abstraction hierarchy
 - Composition tree
 - Association
- Operations
- Object model
- XML representation

■ Is-A Relation

■ Expression

$$\phi_{\text{Is-A}} (v_i, v_j)^n$$

■ Abstraction hierarchy

■ Order of Is-A relation

■ Properties:

■ Subset relation

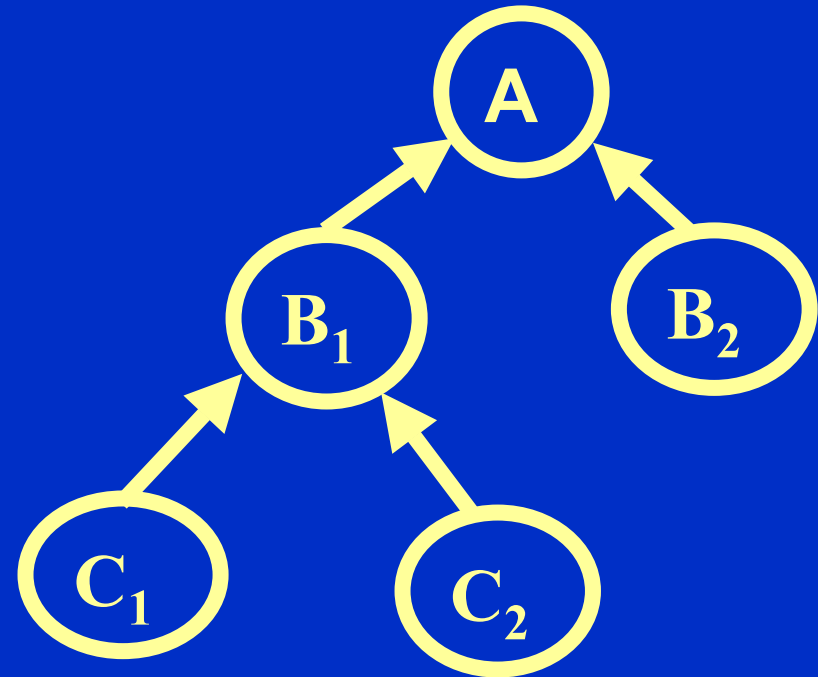
■ All attributes are inherited

$$\mathbf{B_1 \subset A, B_2 \subset A}$$

■ Use:

■ Definition of binding rule

■ Inference



■ Sibling Relation

■ Expression

$$\phi_{\text{sibling}}(v_i, v_j) \mid \exists k \phi_{\text{Is-A}}(v_k, v_i)^1 \wedge \phi_{\text{Is-A}}(v_k, v_j)^1$$

■ Relation between terms with same parent

■ Properties:

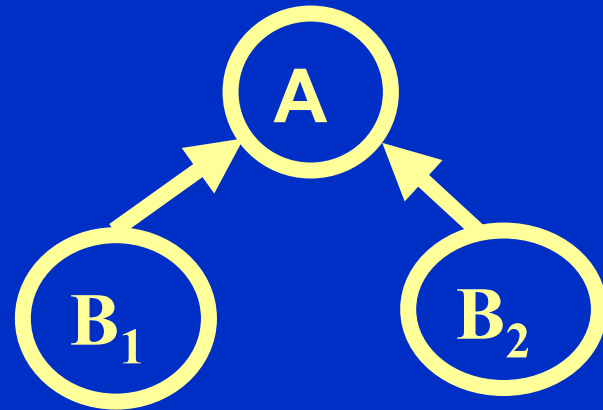
■ Share the same attributes

$$\text{■ } \mathbf{B}_1 \cap \mathbf{B}_2 \subset \mathbf{B}_1$$

$$\text{■ } \mathbf{B}_1 \cap \mathbf{B}_2 \subset \mathbf{B}_2$$

■ Use:

■ Inference of possible binding



- Specialization Relation

- Expression

$$\phi_{\text{spec}} (v_j, v_i)^n$$

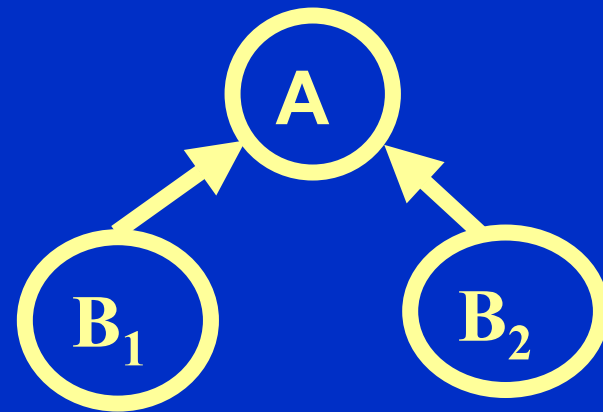
- A specialization in the abstraction level

- Order of specialization relation

- Use:

- Definition of binding rule

- Inference



■ Part-Whole Relation

■ Expression

$$\phi_{\text{part}} (v_i, v_j)^n$$

■ Composition (Building ->Floor->Room)

■ Properties

■ $B_1 \cap B_2 = \emptyset$

■ $B_1 \subset A$

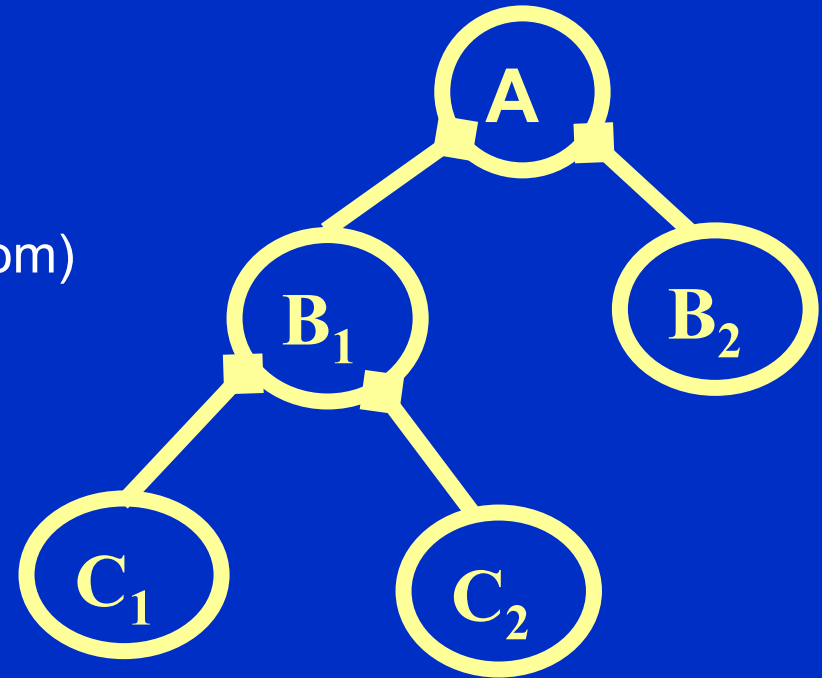
■ Different from Is-A relation

■ Order of part-whole relation

■ Use:

■ Composition-oriented binding

■ Inference of possible composition-oriented binding



■ Buddy Relation

■ Expression

$$\phi_{\text{buddy}}(v_i, v_j) \mid \exists k \phi_{\text{part}}(v_i, v_k) \wedge \phi_{\text{part}}(v_j, v_k)$$

- Same level in the composition relation with same container (Floor->Room1 Floor->Room2)

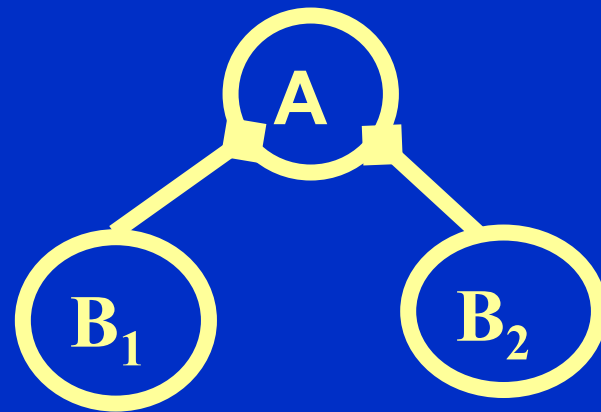
■ Properties:

- $\mathbf{B_1 \cap B_2 = \emptyset}$

- $\mathbf{B_1 \subset A \text{ and } B_2 \subset A}$

■ Use:

- Inference



■ Synonym Relation

■ Expression

$$\phi_{\text{Synonym}}(v_i, v_j)$$

■ Two terms have the exact same meanings

■ Properties

■ Substitutable

■ Use:

■ Synonym-related binding

■ Example

■ Conveyance - Vehicle

- Disjoint relation
 - Expression

$$\phi_{\text{Disjoint}} (v_i, v_j)$$

- Use:
 - Theorem: a data object or data node (D) represented by a term in the domain ontology (TD_1) cannot be bound with an operation (Op) represented by a term in the operation ontology (TO_2) if (TO_2) is bound to (TD_2) and $\phi_{\text{Disjoint}} (TD_1, TD_2)$ holds

- Binding relation

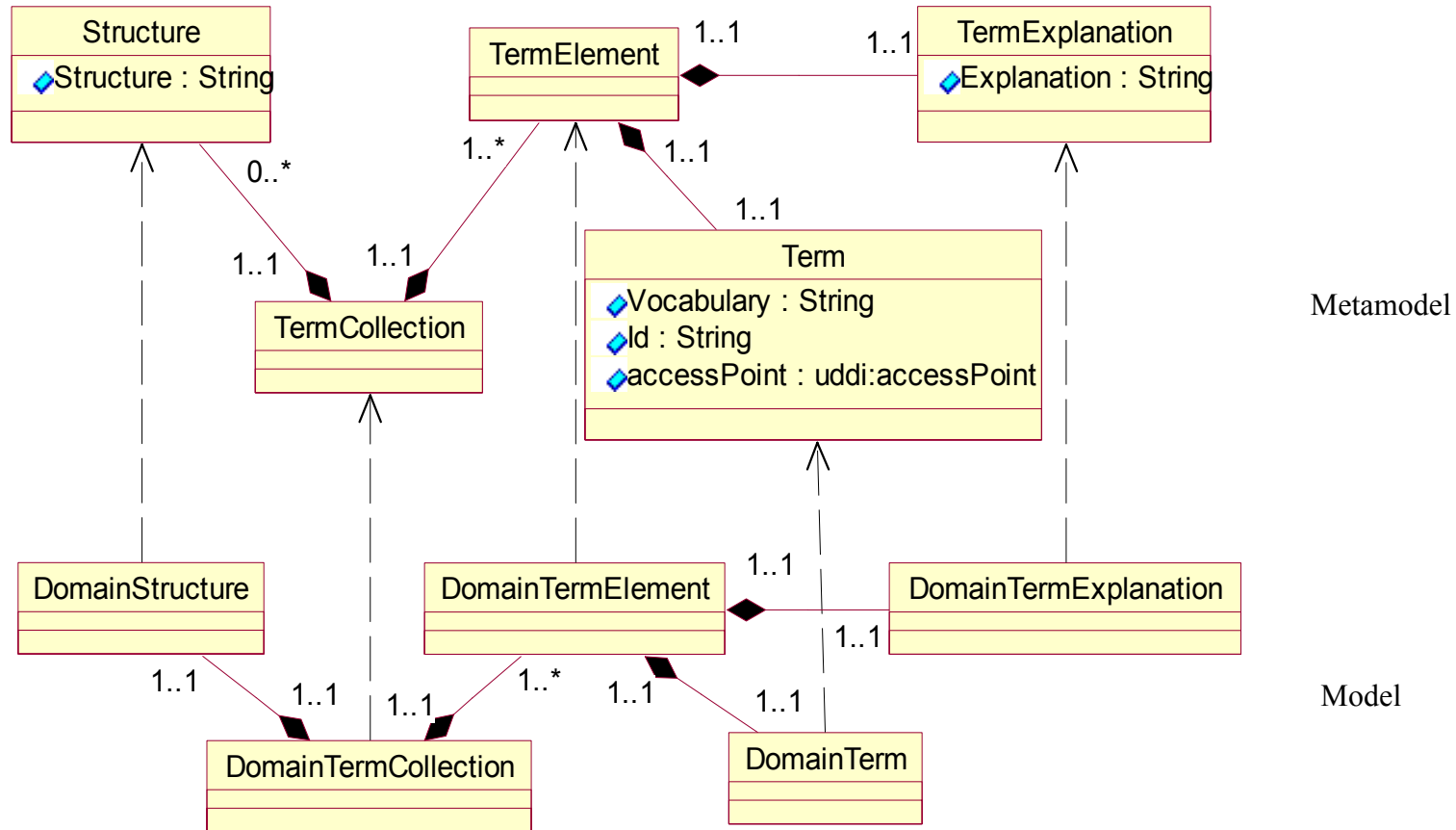
- Expression

$$\phi_{\text{Binding}} (v_i, v_j)$$

- Use:

- Association of operations to data model

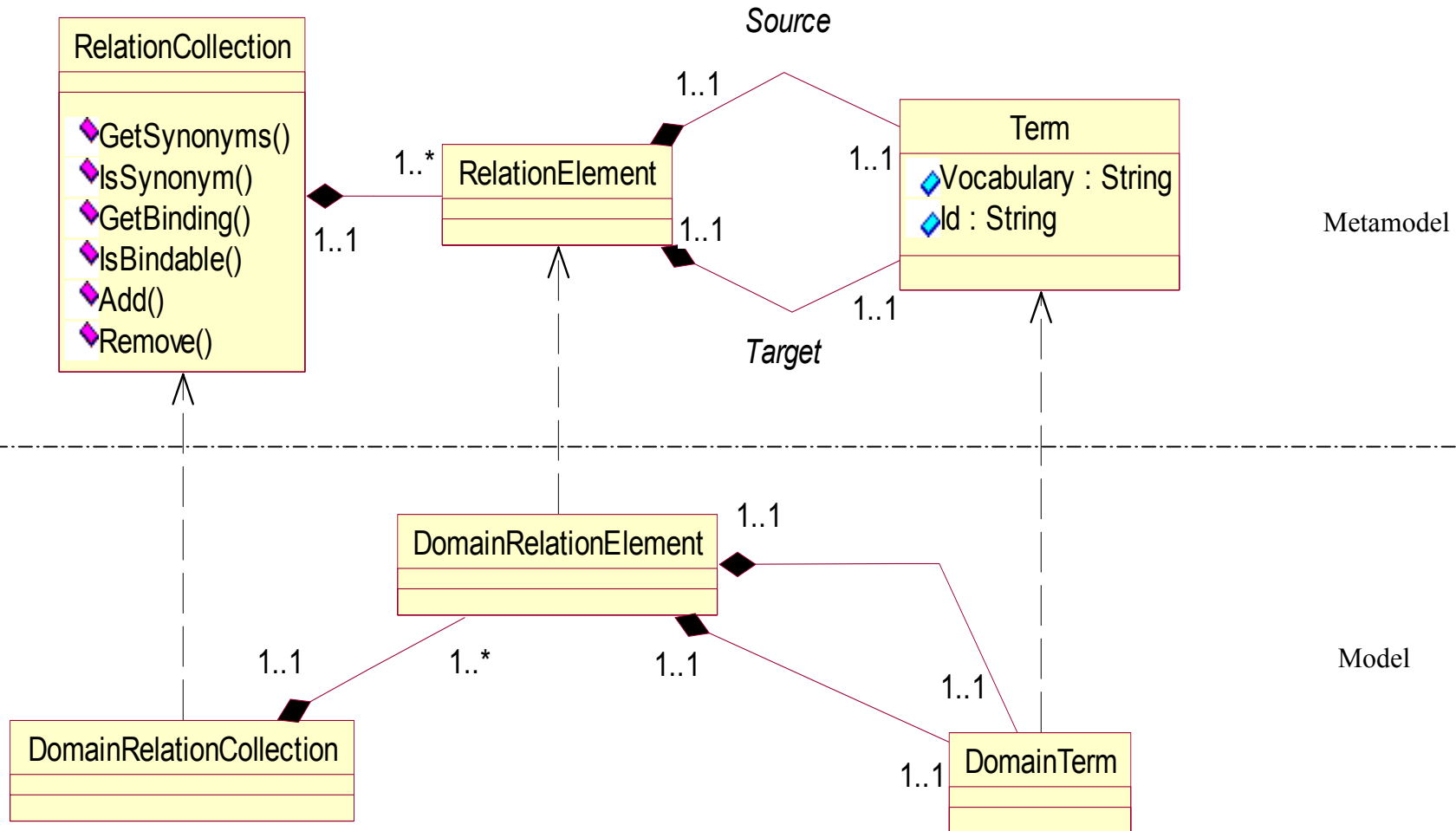
OM – Object Model – Term Model



Metamodel

Model

OM – Object Model – Relation Model



- Reason for XML choice
 - Good exchangeability
 - Ability for machine processing
 - Neutrality of platform and programming language
 - Standard object model for navigation, searching etc.
 - Wide acceptance in industry
- Structure of XML presentation layer
 - Semantic layer - schema
 - Instance layer – XML documents
- Design issue

■ XML Elements

■ Metamodel

- TermCollection
- TermElement
- Structure
- Term , TermExplanation
- RelationCollection, RelationElement

■ Instantiation

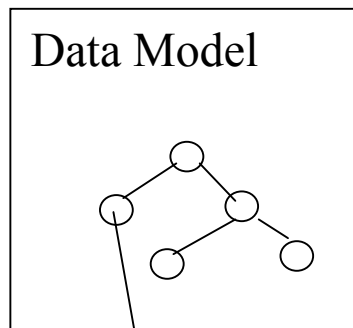
- DomainTermCollection, DomainTermElement, DomainStructure, DomainTerm , DomainTermExplanation
- OperationTermCollection, OperationTermElement, OperationStructure, OperationTerm , OperationTermExplanation
- DomainRelationCollection, DomainRelationElement
- OperationRelationCollection, OperationRelationElement

OM – XML Representation Layer

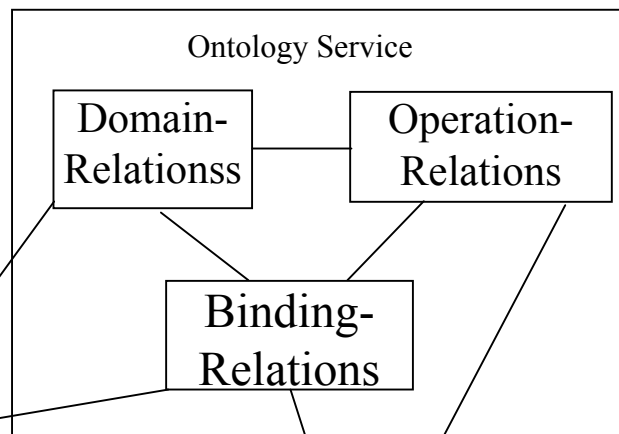


```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="TermCollection" type="TermCollectionType" abstract="true"/>
  <xsd:complexType name="TermCollectionType">
    <xsd:element name="TermElement" type="TermElementType" minOccurs="1" maxOccurs="unbounded" abstract="true"/>
    <xsd:element name="Structure" type="StructureType" minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="accessPoint" type="string" minOccurs="0" maxOccurs="unbounded">
    </xsd:element>
    </xsd:complexType>
  <xsd:complexType name="TermElementType">
    <xsd:element name="Term" type="TermType" minOccurs="1" maxOccurs="1" abstract="true"/>
    <xsd:element name="TermExplanation" type="TermExplanationType" minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>
  <xsd:complexType name="TermType">
    <xsd:attribute name="Vocabulary" type="xsd:string" use="required"/>
    <xsd:attribute name="Id" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="TermExplanationType">
    <xsd:attribute name="Explanation" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:element name="DomainTermCollection" substitutionGroup="TermCollection"/>
  <xsd:element name="DomainTermElement" substitutionGroup="TermElement"/>
  <xsd:element name="DomainTerm" substitutionGroup="Term"/>
  <xsd:element name="OperationTermCollection" substitutionGroup="TermCollection"/>
  <xsd:element name="OperationTermElement" substitutionGroup="TermElement"/>
  <xsd:element name="OperationTerm" substitutionGroup="Term"/>
</xsd:schema>
```

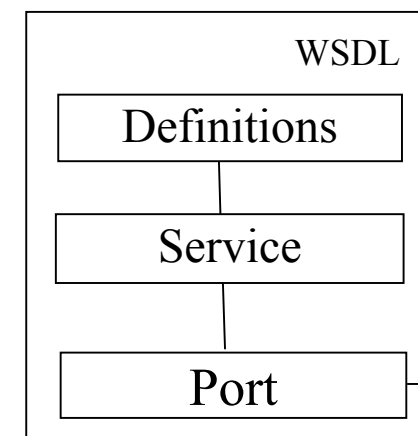
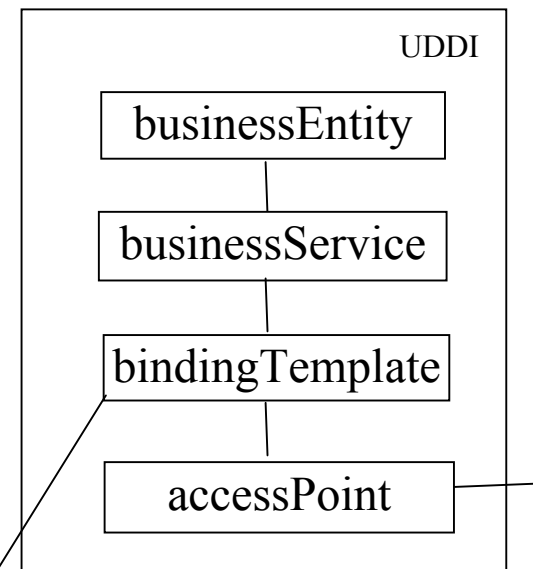
Web Service and System Architecture



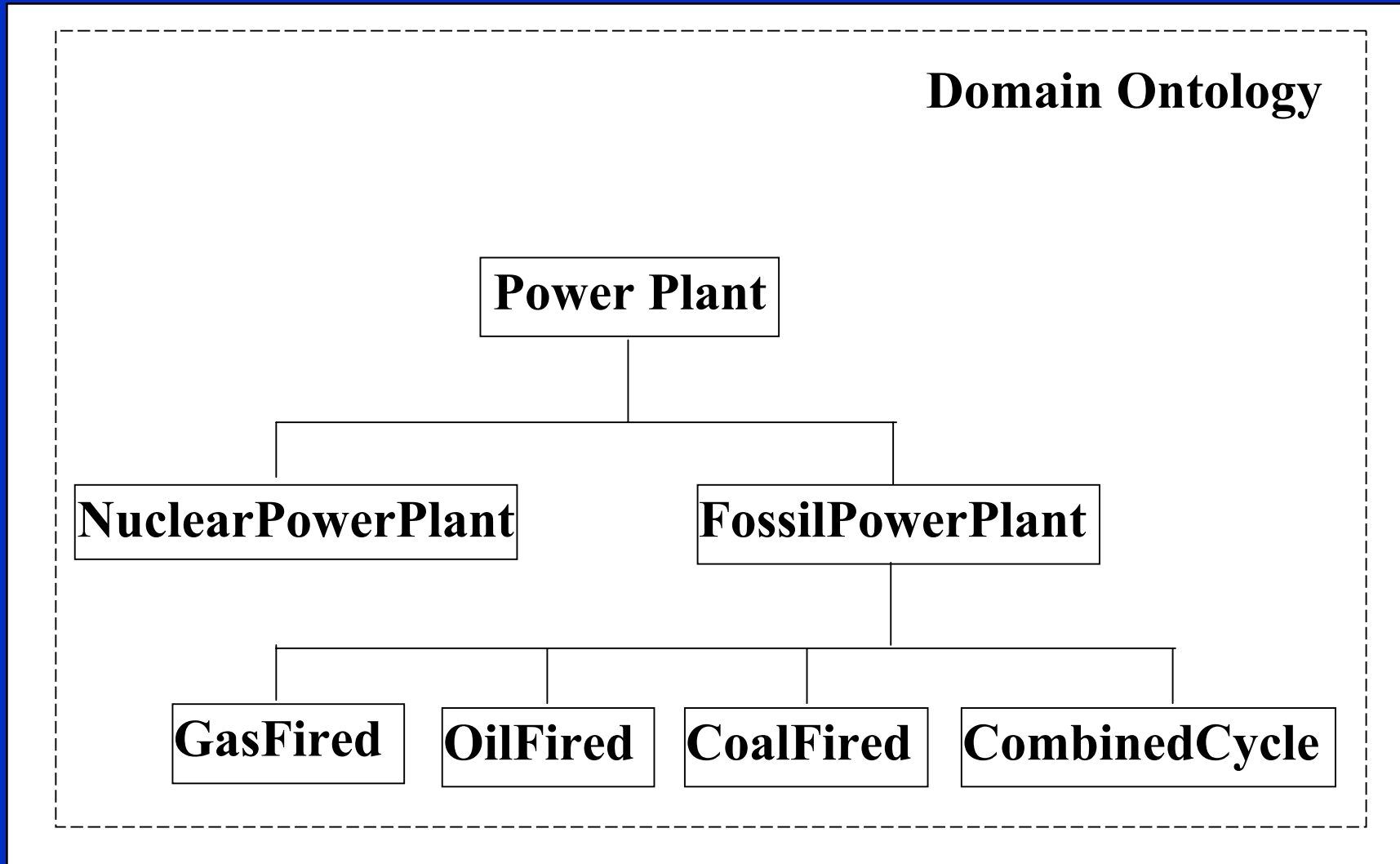
Domain
Ontology



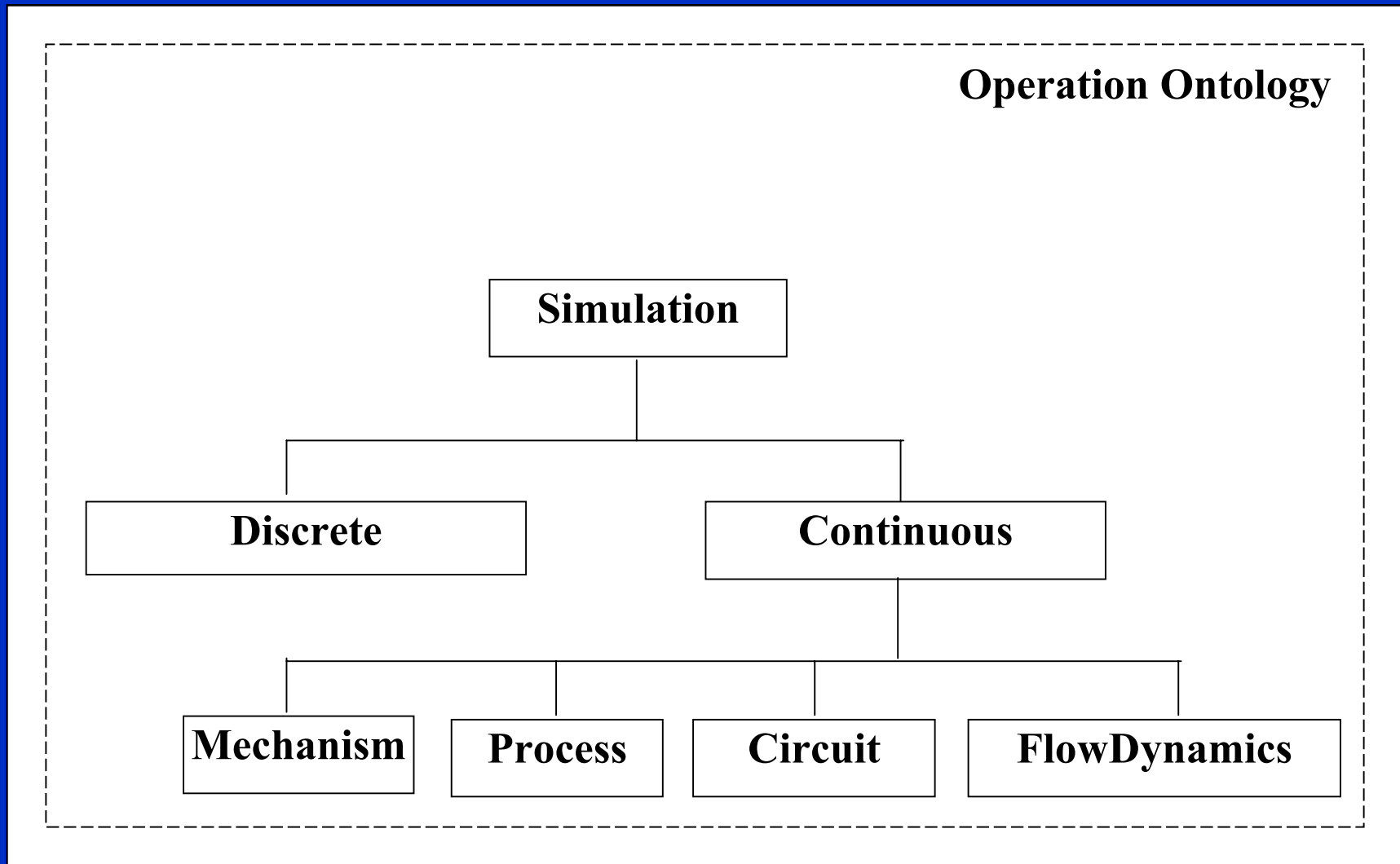
Operation
Ontology



Example - Domain Ontology



Example - Operation Ontology



Example – XML for Domain Ontology



```
<?xml version="1.0" encoding="utf-8" ?>
<DomainTermCollection>
  <DomainTermElement>
    <DomainTerm Vocabulary="PowerPlant" Id="1"/>
  </DomainTermElement>
  <DomainTermElement>
    <DomainTerm Vocabulary="NuclearPowerPlant" Id="2">
  </DomainTermElement>
  <DomainTermElement>
    <DomainTerm Vocabulary="FossilPowerPlant" Id="3">
  </DomainTermElement>
  <DomainTermElement>
    <DomainTerm Vocabulary="CombinedCycle" Id="4"
accessPoint="Plant1">
  </DomainTermElement>
  .....
</DomainTermCollection>
```

Example – XML for Opt. Ontology



```
<?xml version="1.0" encoding="utf-8" ?>
<OperationTermCollection>
  <OperationTermElement>
    <DomainTerm Vocabulary="Simulation" Id="1"/>
  </OperationTermElement>
  <OperationTermElement>
    <OperationTerm Vocabulary="Discrete" Id="100">
  </OperationTermElement>
  <OperationTermElement>
    <OperationTerm Vocabulary="Continuous" Id="101">
  </OperationTermElement>
  <OperationTermElement>
    <OperationTerm Vocabulary="Process" Id="102"
accessPoint="http://128.122.343.23/ps.asmx">
  </OperationTermElement>
  .....
</OperationTermCollection>
```

Example – Domain and Operation Relations in XML

```
<?xml version="1.0" encoding="utf-8" ?>
<DomainRelationCollection>
  <DomainRelation Type="IsA">
    <DomainTerm Vocabulary="FossilPowerPlant"/>
    <DomainTerm Vocabulary="CombinedCycle" />
  </DomainRelation>
</DomainRelationCollection>
```

```
<?xml version="1.0" encoding="utf-8" ?>
<OperationRelationCollection>
  <OperationRelation Type="IsA">
    <OperationTerm Vocabulary="Continuous"/>
    <OperationTerm Vocabulary="Process" />
  </OperationRelation>
</OperationRelationCollection>
```

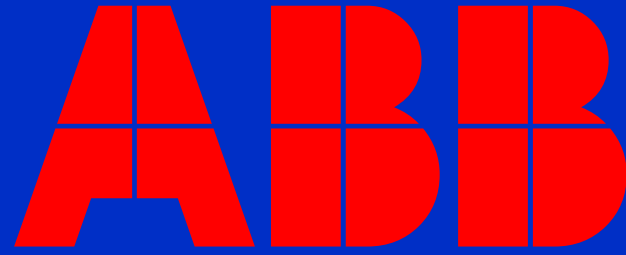
Example – Binding Relations in XML



```
<?xml version="1.0" encoding="utf-8" ?>
<BindRelationCollection>
  <BindingRelation Type="Binding">
    <DomainTerm Vocabulary="CombinedCycle"/>
    <OperationTerm Vocabulary="Process" />
  </BindingRelation>
</BindRelationCollection>
```

Conclusion

Question ?

A large, stylized ABB logo in a vibrant red color. The letters are bold and blocky, with a white horizontal line through the middle of each letter, matching the logo in the header.