SEMANTIC ANNOTATIONS FOR SYSTEMS INTEROPERABILITY IN A PLM ENVIRONMENT (ANNOTATIONS SÉMANTIQUES POUR L’INTEROPÉRABILITÉ DES SYSTÈMES DANS UN ENVIRONNEMENT PLM)

Thesis Defence
14/11/2013

Ph.D. Student: Yongxin LIAO

Thesis Directors: M. Hervé PANETTO
M. Nacer BOUDJILIDA

Thesis Supervisor: M. Mario LEZOCHE

A Quick Overview of the Research Problem

- An Example of without or with explicit semantics during the collaboration

Without Explicit Semantics

- Customers: We need a cup, which is not heavy!
- Analysts: Light = not heavy, use plastic.
- Designers: Light = not heavy, use plastic.
- Customers: Yes!

With Explicit Semantics

- Customers: We need a cup, which is not heavy!
- Analysts: Light = not heavy, suggest to use plastic.
- Designers: Light = not heavy, use plastic.
- Customers: Yes!

Explicit Semantics:
Light = not heavy

A light plastic cup
A Quick Overview of the Research Problem

Outline

1. Research Context and Background
   - Research Context
   - Research Problems
   - Background Investigation
   - Postulates and Research Focus

2. Formalization of Semantic Annotation
   - Overview of the Method
   - Meta-model of a semantic annotation
   - Semantic Blocks
   - Formal Definitions
   - Reasoning Mechanisms
   - Semantic Annotation Framework

3. SAP-KM
   - Overview of SAP-KM
   - Design and Implementation of SAP-KM

4. Case Study
   - Overview of Application Scenario
   - Preparation
   - Annotation
   - Reasoning

5. Conclusion and Discussion
   - Contributions
   - Limitations and Perspectives
Based on this research context, this work is proposed to deal with the following main research problem:

What are the essential elements of a semantic annotation and how can formal semantic annotations contribute to the semantic interoperability in a PLM environment?
Background Investigation

Target Knowledge Representation (TKR)  
(Liao et al. 2013)

Model  
(Miller & Mukerji 2003)

Knowledge Representation  
(Davis et al. 1993)

Ontology  
(Gruber, T.R., 1993)

PLC-related Ontology  
(Liao et al. 2013)

Ontology-based Knowledge Representation (OKR)  
(Liao et al. 2013)

"Drawings"  
(Miller & Mukerji 2003)

"Text"  
(Miller & Mukerji 2003)

Structure Semantics  
(Liao et al. 2013)

Domain Semantics  
(Liao et al. 2013)

Meta-model Ontology  
(Liao et al. 2013)

MSDL Ontology (OWL)

CMMI ontology (User-defined)

OntoSTEP ontology (OWL)

SCOM-Full ontology (OWL)

Onto-PDM (Class Diagram)

Bunge-Wand-Weber (Theory)

General Process Ontology (OWL)

Petri net Ontology (OWL)

Activity diagram ontology (User-defined)

BPMN Ontology (OWL)

MSDL Ontology (OWL)

CMMI ontology (User-defined)

OntoSTEP ontology (OWL)

SCOM-Full ontology (OWL)

Onto-PDM (Class Diagram)

Bunge-Wand-Weber (Theory)

General Process Ontology (OWL)

Petri net Ontology (OWL)

Activity diagram ontology (User-defined)

BPMN Ontology (OWL)

(1) Machine readable and processable
(2) Formal and shared terms

Semantic Annotation for Web Services  
(Talantikite et al. 2009), (Patil et al. 2004), (W3C 2007)

Semantic Annotation for Texts  
(Vargas-Vera et al. 2002), (Patil et al. 2004), (Ma et al. 2013)

Semantic Annotation for Models  
(Boudjlida et al. 2006), (Song et al. 2012), (Bergamaschi et al. 2011), (Ji et al. 2012), (Attene et al. 2009), (Lin 2008), (Di Francia e omarino 2011)
**Background Investigation**

- **Existing Drawbacks**
  1. Lack of a formalization of semantic annotations that is able to be used for the semantic enrichments of different kinds of models;
  2. Lack of a mechanism to combine both structure and domain semantics together to contribute in the inference process.
  3. Lack of a mechanism to assist the detection of inconsistencies between semantic annotations and the identification of conflicts between annotated objects.

**Postulates and Research Focus**

- **Postulates**
  - H1 All the knowledge that is needed for the semantic enrichment of models has already been captured, represented and formalized.
  - H2 The corresponding interconnections among all the used ontologies have already been prepared.
  - H3 The semantic similarity between two objects can be compared.

- **Research Focus**
  Formalization of Semantic Annotations for Assisting Semantic Interoperability in the model level
Overview of the Method

- Step 1. Preparation of OKRs
  - Hypothesis 1
  - Hypothesis 2
- Step 2. Creation of Semantic Annotations Schema
  - Meta-model of a Semantic Annotation
  - Formal Definitions
- Step 3. Creation of Reasoning Rules
  - Semantic Annotation Suggestion
  - Inconsistency Detection
  - Conflict Detection
- Step 4. Explicitation of Structure/Domain Semantics
  - Semantic Annotation Schema (from step 2)
  - Semantic Block for Semantic Description
  - Hypothesis 3
- Step 5. Configuration of Reasoning Parameters
  - Semantic Block for Semantics Substitution
  - Hypothesis 3
  - Formal Semantic Annotations (from step 4)
  - Reasoning Rules (from step 3)
  - Reasoning Parameters (from step 5)
- Step 6. Execution of Reasoning

Meta-model of a semantic annotation

- TKR, OKR and SASM
  - TKR: Target Knowledge Representation
  - OKR: Ontology-based Knowledge Representation
  - SASM: Semantic Annotation Structure Model

The Meta-model of a Semantic Annotation

- Target Knowledge Representation
- Ontology-based Knowledge Representation
- Semantic Annotation Structure Model
- Formal Semantic Annotations
- Reasoning Rules
- Reasoning Parameters

Formal Definitions

\[ E \text{ is a set of elements from a "TKR"} \]

**"Domain Semantic" of a "TKR"**

\[ SA = (E, P, SB, MME, MR) \]

**"Structure Semantic" of a "TKR"**

What is \( P \)?

1. Let \( o \) be an ontology
   - Let \( \text{all}_o \) be composed of a set of ontology element \( \text{oc}_o \) (Concept or Relationship).
   - \( o = (C_o, R_o, A_o) \)
   - \( \text{all}_o = (\text{oc}_o | \text{oc}_o \in C_o \cup R_o) \)

2. Let \( PO \) be a set of PLC-related ontologies

\[ PO = (p_1, p_2, \ldots, p_n) \]

(3) Let \( \bigcup_{o \in PO} \text{all}_o \) be the arbitrary union of elements in \( PO \)

\[ \bigcup_{o \in PO} \text{all}_o = (\text{oc} | \text{oc} \in \bigcup_{o \in PO} \text{oc}_o) \]

(4) Then \( P \) is one subset of \( \bigcup_{o \in PO} \text{all}_o \)

\[ P = \{p_1, p_2, \ldots, p_n\} \]

\[ P \subseteq \bigcup_{o \in PO} \text{all}_o \]
Formal Definitions

E is a set of elements from a "TKR"

\[ SA = \langle E, P, SB, MME, MR \rangle \]

"Domain Semantic" of a "TKR"

What is MME?

(1) - Let \( a_0 \) be an ontology
  - Let \( o_{a_0} \) be composed of a set of ontology
    element \( o_{a_0} \) (Concept or Relationship).
  \[ a_1 = (C_{a_0}, R_{a_0}, A_{a_0}) \]
  \[ o_{a_0} = (o_{a_0} \cup \{(a, c) \} \cup C_{a_0} \cup R_{a_0}) \]

(2) - Let \( mme_{a_0} \) be a meta-model ontology
  - Then \( MME \) is composed of a set of ontology
    element \( mme_{a_0} \) (Concept or Relationship).
  \[ mme_{a_0} = (C_{mme_{a_0}}, R_{mme_{a_0}}, A_{mme_{a_0}}) \]
  \[ MME = (mme_{a_0}, mme_{a_0} \cup C_{mme_{a_0}} \cup R_{mme_{a_0}}) \]

What is MR?

- MR is a set of binary relations that describe the semantic relationship from \( E \) to MME. Given \( e_i \in E \) and \( mme_i \in MME \)
  \[ mr_{rel} = \{ (e_i, mme_i) \} \]
  \( e_i \) is a instance of \( mme_i \)

Semantic Blocks

\[ SB_{R_{a_0}} = (A_{R_{a_0}}, B_{R_{a_0}}) \]

Semantic Blocks for Semantic Description

(1) - Let \( a_0 \) be the selected main entity of \( SB_{R_{a_0}} \).
  - Then \( R_{a_0} = \{ b_{a_0}, b_{a_0}, b_{a_0} \} \) be the selected relations.
  - The procedure to get \( A_{R_{a_0}} \) is as follows:
    \[ \begin{align*}
    (1) & A_{R_{a_0}} = \{ a_1 \} \\
    (2) & A_{R_{a_0}} = \{ a_2, a_2 \} \\
    (3) & A_{R_{a_0}} = \{ a_3 \}
    \end{align*} \]

Semantic Blocks for Semantic Substitution

(1) - Let \( a_0 \) be a concept or relationship.
  - Then \( A_{R_{a_0}} \) is a set of ontology elements.
  - Given \( a_0 \) is a concept or relationship.
  \[ \begin{align*}
    (1) & A_{R_{a_0}} = \{ a_1 \} \\
    (2) & A_{R_{a_0}} = \{ a_2, a_2 \} \\
    (3) & A_{R_{a_0}} = \{ a_3 \}
    \end{align*} \]

In practice, the procedure is as follows:

- Let \( a_0 \) be the selected main entity of \( SB_{R_{a_0}} \).
- Then \( R_{a_0} = \{ b_{a_0}, b_{a_0}, b_{a_0} \} \) be the selected relations.
- The procedure to get \( A_{R_{a_0}} \) is as follows:
  \[ \begin{align*}
    (1) & A_{R_{a_0}} = \{ a_1 \} \\
    (2) & A_{R_{a_0}} = \{ a_2, a_2 \} \\
    (3) & A_{R_{a_0}} = \{ a_3 \}
    \end{align*} \]
Reasoning Mechanisms - Semantic Annotation Suggestion

Semantic Annotation

Initial Semantic Annotation

(Boudjlida & Panetto, 2007)

Inferred Semantic Annotation

(Liao et al., 2013)

Suggestion

An Example

Domain Semantics of “Turning” and “Lathe” from Wordnet

Turning: the activity of shaping something on a lathe

Lathe: the machine tool for shaping metal or wood

Semantic Blocks

Semantic Blocks for

Semantic Substitution

(Liao et al., 2013)

Semantic Blocks for

Semantic Description

(Liao et al., 2013)

(1) Relation Explicitation Rule

\[ SBR_{A,B} := (A_{A,B} \cup B_{A,B}) \]

where

- \( A \) and \( B \) are instances of DataObject
- \( \alpha \) and \( \beta \) are instances of SequenceFlow
- \( \xi \) and \( \eta \) are instances of Task
- \( \alpha, \beta, \xi, \eta \) are instances of Association

(2) Delimitation Rule

Select “DataObject” and “Task”

\[ SBR_{A,B} := (A_{A,B} \cup B_{A,B}) \]

\[ SBR_{A,B} := (A_{A,B} \cup B_{A,B}) \]
Research Context and Background | Formalization of Semantic Annotation | SAP-KM | Case Study | Conclusion and Discussion

Reasoning Mechanisms - Inconsistency Detection

Let the $c_i$ be annotated by $p_x$ and $p_y$

1. Given the semantic relationship between $p_x$ and $p_y$

2. Detection results

(a) $c_{x1}$ and $c_{y1}$ are consistent with each other
(b) $c_{x2}$ and $c_{y2}$ are possible consistent with each other
(c) there is an inconsistency between $c_{x3}$ and $c_{y3}$

An example of Inconsistency Detection

<table>
<thead>
<tr>
<th>$p_x$</th>
<th>$c_{x1}$</th>
<th>$c_{x2}$</th>
<th>$c_{x3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_y$</td>
<td>$c_{y1}$</td>
<td>$c_{y2}$</td>
<td>$c_{y3}$</td>
</tr>
</tbody>
</table>

Inconsistency Detection Table

<table>
<thead>
<tr>
<th>$p_x$</th>
<th>$p_y$</th>
<th>$c_{x1}$</th>
<th>$c_{x2}$</th>
<th>$c_{x3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{x1}$</td>
<td>$p_{y1}$</td>
<td>$c_{x11}$</td>
<td>$c_{x21}$</td>
<td>$c_{x31}$</td>
</tr>
<tr>
<td>$p_{x2}$</td>
<td>$p_{y2}$</td>
<td>$c_{x12}$</td>
<td>$c_{x22}$</td>
<td>$c_{x32}$</td>
</tr>
<tr>
<td>$p_{x3}$</td>
<td>$p_{y3}$</td>
<td>$c_{x13}$</td>
<td>$c_{x23}$</td>
<td>$c_{x33}$</td>
</tr>
<tr>
<td>$p_{x4}$</td>
<td>$p_{y4}$</td>
<td>$c_{x14}$</td>
<td>$c_{x24}$</td>
<td>$c_{x34}$</td>
</tr>
</tbody>
</table>

Reasoning Mechanisms - Conflict Identification

Let the $c_i$ be annotated by $p_x$ and $p_y$

1. Given the semantic relationship between $p_x$ and $p_y$

2. Detection results

(a) $c_{x1}$ and $c_{y1}$ are consistent with each other
(b) $c_{x2}$ and $c_{y2}$ are possible consistent with each other
(c) there is an inconsistency between $c_{x3}$ and $c_{y3}$

An example of Conflict Identification

<table>
<thead>
<tr>
<th>$p_x$</th>
<th>$c_{x1}$</th>
<th>$c_{x2}$</th>
<th>$c_{x3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_y$</td>
<td>$c_{y1}$</td>
<td>$c_{y2}$</td>
<td>$c_{y3}$</td>
</tr>
</tbody>
</table>

Conflict Identification Table

<table>
<thead>
<tr>
<th>$p_x$</th>
<th>$p_y$</th>
<th>$c_{x1}$</th>
<th>$c_{x2}$</th>
<th>$c_{x3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{x1}$</td>
<td>$p_{y1}$</td>
<td>$c_{x11}$</td>
<td>$c_{x21}$</td>
<td>$c_{x31}$</td>
</tr>
<tr>
<td>$p_{x2}$</td>
<td>$p_{y2}$</td>
<td>$c_{x12}$</td>
<td>$c_{x22}$</td>
<td>$c_{x32}$</td>
</tr>
<tr>
<td>$p_{x3}$</td>
<td>$p_{y3}$</td>
<td>$c_{x13}$</td>
<td>$c_{x23}$</td>
<td>$c_{x33}$</td>
</tr>
<tr>
<td>$p_{x4}$</td>
<td>$p_{y4}$</td>
<td>$c_{x14}$</td>
<td>$c_{x24}$</td>
<td>$c_{x34}$</td>
</tr>
</tbody>
</table>
Semantic Annotation Framework

- The Preparation Phase
  a) TKR (by modeller)
  b) OKRs (by knowledge engineer)
  c) Semantic Annotation Schema (by Plug-in developer)

- The Annotation Phase
  a) Explicitation of Structure Semantics (by annotator)
  b) Explicitation of Domain Semantics (by annotator)

- The Reasoning Phase
  a) Configuration of Reasoning Parameters (by annotator)
  b) Reasoning (by annotator)

Legend
- Process
- Start/End
- Flow
- Parallel
- Line between Phases
Overview of SAP-KM (Semantic Annotation Plug-in for Knowledge Management)

- Development Environment Overview
  - TKR Creation and Management module -> Mega Modelling Environment
  - OKR Creation and Management module -> Protégé OWL Editor
  - Knowledge Cloud module -> Microsoft Windows Folder System
  - Reasoning Engine module -> Jena OWL reasoning engine
  - Semantic Annotation and Processing Agent module -> SAP-KM Plug-in

Design and Implementation of SAP-KM

- The Design and Implementation of Annotation Phase
  - Explicitation of Structure Semantics

1. Acquire the property restrictions on meta-model ontology.
2. Select a meta-model ontology that is relevant to the main concept of a semantic association.
3. Select a PLC-related ontology from the ontology collection.
4. Select a class or an individual in the ontology to be the main concept of a semantic association.
5. Define the properties between the main concept and its dependent elements as the selected properties.
6. Define the properties between the main concept and its dependent elements in the selected properties.
7. Define the properties between the main concept and its dependent elements as the selected properties.
8. Define the properties between the main concept and its dependent elements in the selected properties.
9. Define the properties between the main concept and its dependent elements as the selected properties.
10. Define the properties between the main concept and its dependent elements in the selected properties.
11. Define the properties between the main concept and its dependent elements as the selected properties.
12. Define the properties between the main concept and its dependent elements in the selected properties.
13. Define the properties between the main concept and its dependent elements as the selected properties.
14. Define the properties between the main concept and its dependent elements in the selected properties.
15. Define the properties between the main concept and its dependent elements as the selected properties.
16. Define the properties between the main concept and its dependent elements in the selected properties.
17. Define the properties between the main concept and its dependent elements as the selected properties.
Design and Implementation of SAP-KM

The Design and Implementation of Annotation Phase
- Explicitation of Domain Semantics

1. Select a PLC-related ontology \( \alpha \), from \( \Omega \).
2. Select a class or an individual \( \beta \) in the ontology \( \alpha \), as the main concept of a semantic block.
3. Create an individual \( \gamma \) in \( \partial \) and set the "hasMainConcept" property from \( \gamma \) to \( \beta \).
4. Create a SBRelation individual for that property, and set the "hasSBRelation" property from \( \gamma \) to this individual.
5. Set the "hasSBEntity" property from \( \gamma \) to the object of the select property.
6. Acquire the selected class's or individual's corresponding properties and the objects of those properties.
7. Select a property and its object as the contents of \( \gamma \).
8. Select a sub property of \( \delta \) to define the semantic relationship from \( \gamma \) to \( \delta \).
9. Association Finished?

Yes: Continue to the next step.
No: Go back to step 8.

Suggest semantic annotations based on the associations and the corresponding \( \gamma \).

Design and Implementation of Reasoning Phase
- Properties Association in Annotation suggestion

1. Select an \( \delta \) for defining the property association.
2. Acquire all the interrelation properties of \( \delta \) and annotate \( \delta \) to define the semantic block delimitation.
3. Acquire the existing SBR creation rules, and display them to annotators.
4. Acquire all the properties of the Main Concept in one or more \( \gamma \) that used to annotate \( \delta \).
5. Select a property in \( \gamma \) to associate with a selected interrelation property of \( \delta \).
6. Association Finished?

Yes: Continue to the next step.
No: Go back to step 5.

Suggest semantic associations based on the annotation and the corresponding \( \gamma \).
Design and Implementation of SAP-KM

- The Design and Implementation of Reasoning Phase
  - Similarity Definition between two Domain Semantics

  Acquire all the individuals of the class E, which have two or more semantic annotations.
  Suggest semantic annotations based on the associations and the corresponding π.
  Define the semantic similarities between two domain semantics of the selected individual.

  Identify possible model mistakes based on the mistake identification rules, the inconsistency detection rules.
  Display all the Results to the annotator.
  Make inference request.

Reasoning Results

- Identify possible model mistakes based on the mistake identification rules.
- Display all the Results to the annotator.
  Make inference request.
Application Scenario

- **AIPL (Atelier Inter-Etablissements de Lorraine)**
  - It is a small scale facility for manufacturing products

Outline of the Case Study

- **Preparation Phase**
  - Overview of the process model

- **Annotation Phase**
  - Semantic Annotations from Upstream-System
  - Semantic Annotations in Current-System
  - Semantic Annotations for Downstream-System

- **Reasoning Phase**
  - Annotation Suggestion
  - Inconsistency detection and Conflict Identification
Preparation Phase

- Overview of the process model

Annotation Phase

- Annotation from the Upstream-System
Annotation Phase

- Annotation in the Current-system
  - The Explicitation of Domain Semantics
    - Element Matching
Annotation Phase

- Annotation in the Current-system
  - The Explicitation of Domain Semantics

<table>
<thead>
<tr>
<th>Model Elements</th>
<th>Domain Semantics</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing</td>
<td>TBI-450</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>T01MaxLength</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>T01MaxLength</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>TETime</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>TPTtime</td>
<td></td>
</tr>
</tbody>
</table>

Annotation Phase

- Annotation in the Current-system
  - The Explicitation of Structure Semantics

<table>
<thead>
<tr>
<th>Model Elements</th>
<th>Structure Semantics</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing</td>
<td>Thing</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>BPMNTask</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>BPMNSequenceFlowFlowTargetRefInv</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>BPMNSequenceFlowTargetRefInv</td>
<td></td>
</tr>
<tr>
<td>Thing</td>
<td>BPMNSequenceFlowTargetRefInv</td>
<td></td>
</tr>
</tbody>
</table>
Research Context and Background | Formalization of Semantic Annotation | SAP-KM | Case Study | Conclusion and Discussion

Annotation Phase

- Annotation in for Downstream-system

  Data Query

  SPARQL Query

  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  PREFIX MEGA: http://www.semanticweb.org/ontologies/2013/6/MEGA_BPMN#
  PREFIX AIPL: <http://www.semanticweb.org/ontologies/2013/6/AIPL_Product_Ontology#>
  PREFIX SANS: <http://www.semanticweb.org/ontologies/2013/6/SemanticAnnotations#>

  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  PREFIX MEGA: http://www.semanticweb.org/ontologies/2013/6/MEGA_BPMN#
  PREFIX AIPL: <http://www.semanticweb.org/ontologies/2013/6/AIPL_Product_Ontology#>
  PREFIX SANS: <http://www.semanticweb.org/ontologies/2013/6/SemanticAnnotations#>

  Query Result

  WHERE {
    ?P rdf:type SANS:P.
    ?P SANS:hasMainConcept ?Process.
    ?TP AIPL:hours ?TPTime.
  }

Reasoning Phase

- Semantic Annotation Suggestion (1/2)

  (1) Semantic Block Delimitation (for saturation substitution)
  a) Select concept “Operation” and “DataObject” to create SBR_DataObject_to_Operation
  b) Select concept “Operation” and “DataObject” to create SBR_DataObject_to_Operation
  c) Delimitation rule

  (2) Data Object Suggestion

  (3) Operation Suggestion

  (4) Adding “&SANS;SBR_DataObject_to_Operation” property from T0 (e.g. “TP = AIPL: TPTime = 0.5”).

  (5) Adding “&MEGA;attachesTo” property from T0 (e.g. “TP = AIPL: TPTime = 0.5”).
Research Context and Background | Formalization of Semantic Annotation | SAP KM | Case Study | Conclusion and Discussion

Reasoning Phase

Semantic Annotation Suggestion (2/2)

1. Semantic Block Delimitation (for semantics description)

   a) Properties Association
   - MEGABPMN Ontology
     - SBR_DataObject_to_Operation
     - hasInput
   - MEGABPMN Ontology
     - BPMNOwnerElement
     - BPMNOwnedElement
     - BPMNOwnedElement

2. TInputs

   a) Get the semantic relationships between domain semantics of a common annotation object (manually performed)
   - $\text{Article} \\ \text{Article} \\ \text{Article} \\ \text{Article} \\ \text{Article}$

   b) $\text{Article} \in \text{Article}$

   c) $\text{Article} \in \text{Article}$

   d) $\text{Article} \in \text{Article}$

   e) $\text{Article} \in \text{Article}$

   f) $\text{Article} \in \text{Article}$

   g) $\text{Article} \in \text{Article}$

   h) $\text{Article} \in \text{Article}$

   i) $\text{Article} \in \text{Article}$

   j) $\text{Article} \in \text{Article}$

   k) $\text{Article} \in \text{Article}$

   l) $\text{Article} \in \text{Article}$

   m) $\text{Article} \in \text{Article}$

   n) $\text{Article} \in \text{Article}$

   o) $\text{Article} \in \text{Article}$

   p) $\text{Article} \in \text{Article}$

   q) $\text{Article} \in \text{Article}$

   r) $\text{Article} \in \text{Article}$

   s) $\text{Article} \in \text{Article}$

   t) $\text{Article} \in \text{Article}$

   u) $\text{Article} \in \text{Article}$

   v) $\text{Article} \in \text{Article}$

   w) $\text{Article} \in \text{Article}$

   x) $\text{Article} \in \text{Article}$

   y) $\text{Article} \in \text{Article}$

   z) $\text{Article} \in \text{Article}$
Conclusion

What are the essential elements of a semantic annotation and how can formal semantic annotations contribute to the semantic interoperability in a PLM environment?

1. What are the semantic interoperability problems that exist during the cooperation in a PLM environment?
   - The survey of product lifecycle, knowledge, system interoperability in this context of PLM

2. What kinds of knowledge representation in a PLM environment need semantic enrichment?
   - The survey of models in a PLM environment

3. What kinds of ontology can be used to support the semantic enrichments of those knowledge representations?
   - The survey of PLC-related ontologies and Meta-model ontologies

4. What are the essential elements of a semantic annotation and how to formally represent a semantic annotation in a suitable format?
   - The meta-model and the formal definitions of a semantic annotation

5. How to semantically enrich a knowledge representation and how can these enriched semantics contribute to the semantic interoperability in a PLM environment?
   - Two kinds of semantic blocks, Three reasoning mechanisms, A semantic annotation procedure and framework, A semantic annotation prototype

Discussion

Limitations

- Limitations from the Hypothesis
  - Cost of the creation of new ontologies
  - Cost of the creation of interconnections
  - Difficulties in semantic similarity measurement

- Limitations of the Case Study
  - Lack of automatic assistances
  - Incompleteness of SBR delimitation rules
  - Small scale facility.

Perspectives

- To improve automatic assistances
- To complete SBR delimitation rules
- To apply this method in a larger scale facility
- To develop plug-in for other systems
Thank You For Paying Attention!

Yongxin LIAO