AN HYBRYD APPROACH FOR MODELS COMPARISON

Samia Benabdellah Chaouni ENSIAS, Mohammed V Souissi University, Rabat, Morocco
bcsamia@gmail.com

Mounia Fredj ENSIAS, Mohammed V Souissi University, Rabat, Morocco
fredj@ensias.ma

Salma Mouline FSR, Faculty of Sciences, Rabat, Morocco
mouline@fsr.ac.ma

ABSTRACT
Model integration problem occurs during the integration of enterprise information systems. Models comparison is an essential step of the integration, and has been discussed in several domains and various models. However, previous approaches have not correctly handled the semantic comparison. In the current paper, we develop a comparison hybrid approach which takes into account the syntactic, semantic and structural comparison aspects. We provide a rule-based system for models comparison. For this purpose, we use a domain ontology as well as other resources such as dictionaries.

KEYWORDS
UML models comparison, Ontology, rules, metamodel

1. INTRODUCTION
The information systems domain has changed dramatically in recent years under the influence of organizational evolution. This evolution can be of internal origin, generated by the restructuring of organizations, creation of new subsidiaries or new geographic or changes in business activity. Result of these factors, new information systems with their business models are created, the need to integrate existing models to make them communicate and cooperate. This evolution may also be of external origin, explained by the evolution of two organizations with the same activity domain who want merge. In this case, it must merge their information
AN HYBRYD APPROACH FOR MODELS COMPARISON

systems and more specifically their models. The goal is to integrate these models easily and efficiently.

Integration has been treated by several authors, for several models in different fields and contexts: the schemas database integration (Spaccapietra and al., 94) and (Navathe and al., 86); integration of meta-models independent models (UML, database schema, ...) (Haddar, 02) and (Pottinger and al., 03); views models integration (Anwar and al., 07) and (Rubin and al., 08); partial UML class diagrams integration (Borona et al., 06), aspect-oriented UML models integration (Ferut, 06), (Quintian, 04), (Reddy and al., 03), (Lahire and al., 06), (Olivier and al., 07) and (Fleurey and al., 07); and finally, ontology integration, which has been treated in (Falquet et al., 04), (Ouagne et al., 05) (Dorion et al., 07) and (Bouras et al., 07). We are interested in our case in the UML models integration and more specifically the UML class diagrams (OMG UML, 09). After the analysis of models integration existing work, we found that semantic integration is a crucial problem. So far, this problem is still not properly treated. In this paper, we focus on models’ comparison (the first stage of the integration process). We propose an hybrid approach which compares models syntactically, semantically and structurally. For that, we use domain ontology and other resources.

This article is organized as follows: Section 2 is an introduction to the general approach of integration of models. We mention in section 3 related work and their limitations. Our ontology-based proposal is developed in section 4. Some research perspectives are finally developed in the conclusion section.

2. MODELS INTEGRATION

The integration is defined as the combination of components in such a way as to form a new set constituting a unit for creating synergy (Weston, 1993). Existing research (Batini and al., 86) (Pottinger and al, 03) has shown that models integration process involves two steps: 1) the comparison step is based on a set of rules called correspondence rules, also called comparison rules, mapping rules or matching rules which identify the correspondence between elements of models (correspondences created during this step are stored in a separate model called correspondence model or mapping model); 2) the integration step integrates models mapped in the previous step. The integration strategy relies on rules that define which and how elements will appear in the result model. These rules are (1) rules for merging the matching elements (merging rules), and (2) rules for incorporating elements that do not belong to the mapping model (rules of integration).

3. RELATED WORK

Several studies have proposed models comparison. The authors (Manning, 99), (Haddar and al., 02) and (Oliveira, 2009) provided a comparison of meta-model independent models. Databases comparison has been treated in (Madhavan and al, 01) and (Reddy and al., 06). The authors provided a comparison of UML class diagrams oriented aspects. In (Anwar and al., 07), a comparison of views models is proposed. (Uhlig et al., 2008) develop a method to
compare UML class diagrams. The specification of UML 2.1 (OMG UML, 09) defines the comparison of packages.

We found different approaches of models comparison:
- **Syntactic approaches:** they compare the letters of strings of models elements.
- **Semantic approaches:** they compare the meaning associated with the compared items.
- **Local structural approaches:** they compare the components of the elements. For example, the comparison of local structure of two classes corresponds to the comparison of their attributes and operations.
- **Global structural approaches:** they compare elements in relation with the elements to compare. For example, the comparison of global structure of two relations corresponds to the comparison of the two classes that they connect.
- **Hybrid approaches:** they combine two, three or four types of comparison (syntactic, semantic, global structure and local structure).

The table below displays a synthesis of these works. References of the approaches are shown on the first column. The existing types of comparison are provided on the other columns. Crosses (X) show which type of comparison is used by the approach.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Syntactic</th>
<th>Semantic</th>
<th>Structural</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identity</td>
<td>Abbreviation</td>
<td>Acronymy</td>
</tr>
<tr>
<td>(Manning, 99)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Madhavan and al., 01)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(Haddar and al., 02)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Reddy and al., 06)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Anwar and al., 07)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Uhrig et al., 2008)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(OMG UML, 09)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Oliveira, 2009)</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Let M1 and M2 be two models to compare. Most approaches compare syntactically models elements. However, they only test **identity** of elements. (Madhavan and al., 01) also detects other correspondences such as **abbreviation** (e.g. “Qty” in M1 and “Quantity” in M2) and the **acronym** (e.g. "UOM" in M1 and "UnitOfMeasure" in M2). Moreover, most approaches structurally (local and global structure) compare the models elements. Finally, all these works do not take into account the semantic aspect and are limited to detection of **synonyms** (e.g. "Book" in M1 and "Work" in M2) and **homonyms** (e.g. two classes "Family" (products) and "Family" (people)).

Our review showed on the one hand that existing works do not detect semantic mappings such as **disjunction** (e.g. two boolean attributes "Single" and "Married") and **reverse** (e.g. the relation "Buy" is the inverse of "BoughtBy" relation). Syntactic correspondences such as **inclusion** syntactic (e.g. “Student” and “Students”) and **multilingual** (e.g. “Nom” (In French) and “Name” (In English)) are not detected either. Any approach is incomplete. One may also
emphasize that approaches are complementary, even though their union does not cover all
types of comparison and does not detect all matches (correspondences).

On the other hand, syntactic approaches are limited because they do not detect elements
that are syntactically identical but do not have the same meaning (case of homonyms) and
elements which are syntactically different but which have the same meaning (case of
synonyms). In addition, non-semantic approaches are limited because they do not detect
elements that are syntactically different but semantically identical. Non-local structural
approaches are also limited because they do not detect elements which are syntactically
identical but different in local structure (e.g. two classes having the same name and no
attribute in common). Finally, non-global structural approaches are limited because they
cannot detect elements that are syntactically different and equivalent in global structure (e.g.
two relations that are syntactically different but connect two equivalent classes).

Therefore, our goal is to provide an hybrid approach incorporating syntactic, structural and
semantic aspects in order to detect any mapping or correspondence.

4. PROPOSITION

Our proposal is based on ontological techniques. We therefore briefly introduce ontology
concepts, before developing our approach.

4.1 Ontology

Ontologies are introduced as an”explicit specification of a conceptualization” (Gruber, 93). Domain
ontologies are ontologies which are built on a particular knowledge domain. Many
domain ontologies exist such as MENELAS (medical domain) (Zweigenbaum and al., 94) and
TOVE (business management domain) (Gruber, 95). The domain ontology is a semantically
rich model (it can express equivalence, inverse, disjunction, symmetry, transitivity, etc.), and
is defined as an exhaustive list of concepts and relations between these concepts describing a
particular field (Medicine, Business, Library, Restaurants, etc.).

We use an OWL ontology (Ontology Web language) because it is a W3C recommendation
(Smith and al., 2004), and the meta-model OWL was defined by Ontology Definition
Metamodel specification (ODM, 08) of OMG¹. An ontology comprises the notion of
"concept", also called class, corresponding to the abstractions of the relevant field. It has a
name and is characterized by data properties. "Data property" allows to represent the
relationship that connects the concept to a data type (integer, boolean, etc.). It is equivalent to
an attribute of class. Relationship between concepts, called "Object property", reflects the
interaction between concepts, it has a name and connects a source concept called "Domain" to
a target concept called "Range". “Subsumption relations”, links a specific class to a more
generally class.

¹ www.omg.org
4.2 Comparison Approach

Our goal is to provide a semantic comparison approach integrating syntactic and structural aspects as well (Figure 1). We propose a system called COM²Model (Complete Comparison of Models) that takes two models as input and gives correspondence models as output. COM²Model is syntactic, semantic and structural rules-based. It detects mappings between models elements. We used strategies based on semantic properties to take into account the semantic aspect. Therefore, our system refers to a domain ontology that will enable to provide semantic relevant information and decision-making during the comparison. Our system is also based on other resources to complete syntactic comparison. We use a multilingual dictionary (translation) as EuroWordNet², an acronym dictionary³, an abbreviation dictionary⁴, and a dictionary of synonyms as WordNet⁵. In our approach, we consider that we have at our disposal the domain ontology and the other resources. We provide a system for decision support. Our system allows the user to validate or delete mappings automatically created.

![Figure 1. COM²Models architecture](image)

Our comparison process starts with the comparison of syntactical and semantical elements (first classes, second attributes, third operations and fourth relations). It next compares elements (in the same order as just described) in global structures and in local structures.

4.3 Comparison Rules

We provided a first version of rules comparison in informal (natural) language in (Benabdellah et al., 10a) and an improved version applied to a case study in (Benabdellah et al., 10b). To specify the language for expressing these rules, we propose a meta-model.

4.3.1 MDE

Model-driven engineering (MDE) is a software development approach that has the potential to address the identified challenges of software engineering. It offers an environment that ensures the systematic and disciplined use of models throughout the development process of software systems. The essential idea of MDE is to shift the attention form program code to models. This way models become the primary development artifacts that are used in a formal and precise way.

² http://www.illc.uva.nl/EuroWordNet/
³ http://acronymes.info/
⁴ http://theleme.enc.sorbonne.fr/dico.php
⁵ http://wordnet.princeton.edu/
The MDE approach identifies tools and materials necessary for the implementation of its paradigm. We find among others model, metamodel, language.

The most comprehensive definition of model is given by (Bézivin et al., 01): "A model is a simplification of a system built with an intended goal in mind. The model should be able to answer questions in place of the actual system." According to (MOF, 02), "A metamodel is a model that defines the language for expressing a model".

In our case, the model is the comparison rules. We define a metamodel that defines the language for expressing these rules.

4.3.2 Models Comparison Rules Metamodel

We modeled our metamodel in UML language. The rule is characterized by a name, a boolean result (i.e. true or false) and the type (commutative or not). The rule can be syntactic, semantic, global structure or local structure. It is composed of parameters that have a name. These parameters belong to a set of elements. A rule can call one or more other rules.

4.3.3 Comparison Rules

We first established the syntactic comparison rules: rule of identity, rule of inclusion, rule of equivalence multilingual, rule of acronym, rule of abbreviation and rule of syntactic equivalence. Then the comparison semantic rules: rule of synonymy of classes, rule of equivalence of classes (as an ontology), rule of semantic equivalence of classes, rule of hyponymy of classes, rule of synonymy of attributes, rule of disjunction of attributes, rule of semantic equivalence of attributes, rule of operations synonymy, rule of semantic equivalence of operations, rule of synonymy of relations, rule of inverse relation, rule of equivalence of relations (as an ontology), and rule of semantic equivalence of relations. Then the rules for comparing global structure elements (classes, attributes, operations, relations and generalization relation). And finally, rules for comparing local structure elements (classes, attributes, operations and relations).
Some representative rules in accordance to the comparison rules metamodel are described below.

- **Rule of syntactic inclusion of two elements** \(elt_i\) and \(elt_j\)

This is a syntactic rule, called “Syntactic_inclusion”, compares two elements (parameters) called \(D_ielt_i\) and \(D_2elt_j\). The first element belongs to the set of elements of the first diagram called \(D_1E\) and the second element belongs to the set of elements of the second diagram called \(D_2E\). This commutative rule returns 1 (true) if the first elements are included syntactically in the second, and else returns 0 (false).

\[
\text{Syntactic_inclusion : } D_1E \times D_2E \rightarrow \{0,1\}
\]

\[
\text{Syntactic_inclusion} (D_1elt_i, D_2elt_j) = \begin{cases} 
1 
& \text{if } \exists m \in \mathbb{N} \text{ such that } D_1elt_i.numeq = p + D_1elt_i.numeq + s \text{ or } D_2elt_j.numeq = p + D_2elt_j.numeq + s \\
0 & \text{else}
\end{cases}
\]

**Rule explanation:** A first element is syntactically included in a second element if the name of the first element appended to a prefix and (or) a suffix gives the name of the second element.

- **Rule of syntactic equivalence of two elements** \(elt_i\) and \(elt_j\)

This is a syntactic rule, called “Equivalence_syntactic_elements”, compares two elements (parameters) called \(D_ielt_i\) and \(D_2elt_j\). The first element belongs to the set of elements of the first diagram called \(D_1E\) and the second element belongs to the set of elements of the second diagram called \(D_2E\). This rule called other rules called “Syntactic_inclusion”, “Syntactic_Identity”, “Acronyms_Equivalence”, “Abbreviation_Equivalence” and “Multilingual_Equivalence”. This commutative rule returns 1 (true) if the two elements are syntactically equivalent, and else returns 0 (false).

\[
\text{Equivalence_syntactic_elements : } D_1E \times D_2E \rightarrow \{0,1\}
\]

\[
\text{Equivalence_syntactic_elements} (D_1elt_i, D_2elt_j) = \begin{cases} 
1 & \text{if } \text{Syntactic_inclusion} (D_1elt_i, D_2elt_j) = 1 \text{ or } \text{Syntactic_Identity} (D_1elt_i, D_2elt_j) = 1 \\
\text{or } \text{Acronyms_Equivalence} (D_1elt_i, D_2elt_j) = 1 \text{ or } \text{Abbreviation_Equivalence} (D_1elt_i, D_2elt_j) = 1 \\
\text{or } \text{Multilingual_Equivalence} (D_1elt_i, D_2elt_j) = 1 \\
0 & \text{else}
\end{cases}
\]

**Rule explanation:** Two elements are syntactically equivalent, if one of the following conditions is performed: syntactically identical, a first element is syntactically included in a second element or a second element is syntactically included in a first element, a first element is acronym of a second element or a second element is acronym of a first element, a first element is an abbreviation of a second element or a second element is an abbreviation of a first element, or a first element is the translation of a second element or a second element is the translation of a first element. 

- **Rule of semantic equivalence of two relations** \(R_i\) and \(R_j\)

This is a semantic rule, called “Equivalence_semantic_relations”, compares two elements (parameters) called \(D_iR_i\) and \(D_2R_j\). The first element belongs to the set of relations of the first
diagram called $D_jR$ and the second element belongs to the set of relations of the second diagram called $D_2R$. This rule called other rules called “Synonymy_elements”, “Inverse_relations” and “Equivalence_Ontology_relations. This commutative rule returns 1 (true) if the two elements are semantically equivalent, and else returns 0 (false).

**Equivalence_semantic_relations:** $D_1R \times D_2R \rightarrow \{0,1\}$

\[
\begin{aligned}
&1. \text{if Synonymy_elements}(D_1R_1, D_2R_1) = 1 \text{ or Inverse_relations}(D_1R_1, D_2R_1) = 1, \\
&\text{or Equivalence_Ontology_relations}(D_1R_1, D_2R_1) = 1, \\
&\text{else,}
\end{aligned}
\]

**Rule explanation:** Two relations are semantically equivalent if they are synonyms, equivalent (in reference to ontology) or reverse.

**Rule of semantic equivalence of two classes $C_i$ and $C_j$**

This is a semantic rule, called “Equivalence_semantic_classes”, compares two elements (parameters) called $D_1C_i$ and $D_2C_j$. The first element belongs to the set of classes of the first diagram called $D_1C$ and the second element belongs to the set of classes of the second diagram called $D_2C$. This rule called other rules called “Synonymy_elements” and “Equivalence_Ontology_classes. This commutative rule returns 1 (true) if the two elements are semantically equivalent, and else returns 0 (false).

**Equivalence_semantic_classes:** $D_1C \times D_2C \rightarrow \{0,1\}$

\[
\begin{aligned}
&1. \text{if Synonymy_elements}(D_1C_i, D_2C_j) = 1 \text{ or Equivalence_Ontology_classes}(D_1C_i, D_2C_j) = 1, \\
&\text{else,}
\end{aligned}
\]

**Rule explanation:** Two classes are semantically equivalent if they are synonyms or they are equivalent (referring to an ontology).

**Rule for comparing global structure of two relations $R_i$ and $R_j$**

This is a global structural rule, called “Equivalence_structure_global_relations”, compares two elements (parameters) called $D_1R_i$ and $D_2R_j$. The first element belongs to the set of relations of the first diagram called $D_1R$ and the second element belongs to the set of relations of the second diagram called $D_2R$. This rule calls other rules. This commutative rule returns 1 (true) if the two elements are equivalent in global structure, and else returns 0 (false).

**Equivalence_structure_global_relations:** $D_1R \times D_2R \rightarrow \{0,1\}$

\[
\begin{aligned}
&1. \text{if } (\text{Equivalence_semantic_relations}(D_1R_1, D_2R_1) = 1 \text{ or Equivalence_synactic_elements}(D_1C_1, D_2C_1)) \\
&\text{and (Equivalence_semantic_relations}(D_1C_1, D_2C_1) = 1 \text{ or Equivalence_synactic_elements}(D_1C_2, D_2C_2)) \\
&\text{or (Equivalence_semantic_relations}(D_1C_1, D_2C_1) = 1 \text{ or Equivalence_synactic_elements}(D_1C_2, D_2C_2)) \\
&\text{and (Equivalence_synactic_relations}(D_1R_1, D_2R_1) = 1 \text{ or Equivalence_synactic_elements}(D_1C_1, D_2R_1)) \\
&\text{and (Equivalence_synactic_relations}(D_1R_1, D_2R_1) = 1 \text{ and (Equivalence_synactic_relations}(D_1R_2, D_2R_2)) = 1 \\
&\text{or Equivalence_synactic_elements}(D_1C_2, D_2C_2)) \text{ and (Equivalence_synactic_relations}(D_1R_2, D_2R_2)) = 1 \\
&\text{or Equivalence_synactic_elements}(D_1C_2, D_2C_2)))) \\
&\text{else,}
\end{aligned}
\]
Rule explanation: Two relations \(D_1R_i\) and \(D_2R_i\) are equivalent in global structure if: (There is two classes \(D_1C_k\) and \(D_1C_m\) such as \(D_1R_i\) links them and there is two classes \(D_2C_l\), \(D_2C_n\) such as \(D_2R_i\) links them and \(D_1C_k\) and \(D_2C_l\) are syntactically or semantically equivalent) Or (There is two classes \(D_1C_k\) and \(D_1C_m\) and there is \(D_1C_o\) class such as \(D_1R_i\) links \(D_1C_o\) and \(D_1C_m\) and there is two classes \(D_2C_l\), \(D_2C_n\) such as \(D_2R_i\) links them and \(D_1C_k\) and \(D_2C_l\) are syntactically or semantically equivalent and \(D_1C_m\) and \(D_2C_n\) are syntactically or semantically equivalent).

Rule for comparing global structure of two classes \(C_i\) and \(C_j\)

This is a global structural rule, called “Equivalence_structure_global_classes”, compares two elements (parameters) called \(D_1C_i\) and \(D_2C_j\). The first element belongs to the set of classes of the first diagram called \(D_1C\) and the second element belongs to the set of classes of the second diagram called \(D_2C\). This rule calls other rules. This commutative rule returns 1 (true) if the two elements are equivalent in global structure, and else returns 0 (false).

Explanation: Two classes \(D_1C_i\) and \(D_2C_j\) are equivalent in global structure if: [(There is \(D_1R_k\) relation and \(D_1C_m\) class such as \(D_1R_k\) links \(D_1C_i\) and \(D_1C_m\) and there is \(D_2R_l\) relation and \(D_2C_o\) class as \(D_2R_l\) links \(D_2C_j\) and \(D_2C_o\) and \(D_2R_l\) and \(D_2R_k\) are syntactically or semantically equivalent) Or [(There is \(D_1R_k\) relation and \(D_1C_m\) class such as \(D_1R_k\) links \(D_1C_i\) and \(D_1C_m\) and there is \(D_2R_l\) relation and \(D_2C_o\) class such as \(D_2R_l\) links \(D_2C_j\) and \(D_2C_o\) and \(D_2R_l\) and \(D_2R_k\) are syntactically or semantically equivalent)] or [(There is \(D_1R_k\) relation and \(D_1C_m\) class such as \(D_1R_k\) links \(D_1C_i\) and \(D_1C_m\) and there is \(D_2R_l\) relation and \(D_2C_o\) class such as \(D_2R_l\) links \(D_2C_j\) and \(D_2C_o\) and \(D_2R_l\) and \(D_2R_k\) are syntactically or semantically equivalent) And \(D_1C_m\) and \(D_2C_o\) are syntactically or semantically equivalent)] or [(There is \(D_1R_k\) relation and \(D_1C_m\) class such as \(D_1R_k\) links \(D_1C_i\) and \(D_1C_m\) and there is \(D_2R_l\) relation and \(D_2C_o\) class such as \(D_2R_l\) links \(D_2C_j\) and \(D_2C_o\) and \(D_2R_l\) and \(D_2R_k\) are syntactically or semantically equivalent) And \(D_1C_m\) and \(D_2C_o\) are syntactically or semantically equivalent).]

Rule for comparing local structure of two classes \(C_i\) and \(C_j\)

This is a local structural rule, called “Equivalence_structure_local_classes”, compares two elements (parameters) called \(D_1C_i\) and \(D_2C_j\). The first element belongs to the set of classes of the first diagram called \(D_1C\) and the second element belongs to the set of classes of the second diagram called \(D_2C\). This rule calls other rules. This commutative rule returns 1 (true) if the two elements are equivalent in local structure, and else returns 0 (false).
AN HYBRYD APPROACH FOR MODELS COMPARISON

Equivalence_structure_local_classes: \( D_1 \times D_2 \rightarrow \{0,1\} \)

Equivalence_structure_local_classes(D_1, D_2):

\[
\begin{cases}
1 & \text{if } \{ D_1(C_1, C_2, C_3, C_4), D_2(C_1, C_2, C_3, C_4) \} \text{ is equivalent in semantics attributes: } \{ D_1(T_1, T_2), D_2(T_1, T_2) \} \equiv 2 \\
or \text{Equivalence_synthetic_elements}(D_1(T_1, T_2), D_2(T_1, T_2)) & \text{or } \text{Equivalence_synthetic_elements}(D_1(T_1, T_2), D_2(T_1, T_2))
\end{cases}
\]

Rule explanation: Two classes are equivalent in local structure if their attributes and operations are syntactically or semantically equivalent.

Rule for comparing local structure of two attributes T_i and T_j

This is a local structural rule, called “Equivalence_structure_local_attributes”, compares two elements (parameters) called D_1T_i and D_2T_j. The first element belongs to the set of attributes of the first diagram called D_1(T) and the second element belongs to the set of attributes of the second diagram called D_2(T). This commutative rule returns 1 (true) if the two elements are equivalent in local structure, and else returns 0 (false).

Equivalence_structure_local_attributes(D_1T_i, D_2T_j): \( D_1T \times D_2T \rightarrow \{0,1\} \)

Equivalence_structure_local_attributes(D_1T_i, D_2T_j) = \{ 1 \text{ if } D_1T_i, \text{type} = D_2T_j, \text{type} \}, \text{else} 0

Rule explanation: Two attributes are equivalent in local structure if they have the same type.

Rule for comparing local structure of two relations R_i and R_j

This is a local structural rule, called “Equivalence_structure_local_relations”, compares two elements (parameters) called D_1R_i and D_2R_j. The first element belongs to the set of relations of the first diagram called D_1(R) and the second element belongs to the set of relations of the second diagram called D_2(R). This commutative rule returns 1 (true) if the two elements are equivalent in local structure, and else returns 0 (false).

Equivalence_structure_local_relations(D_1R_i, D_2R_j): \( D_1R \times D_2R \rightarrow \{0,1\} \)

Equivalence_structure_local_relations(D_1R_i, D_2R_j) = \{ 1 \text{ if } R_i, \text{type} = R_j, \text{type} \}, \text{else} 0

Rule explanation: Two relations are equivalent in local structure if they have the same type, the same multiplicity and the same navigability.
5. CONCLUSION

Any approach of model comparison must take into account syntactic, semantic and structural aspects. The semantic integration of models is a complex task because it requires understanding the semantics of linking concepts. The main contribution of this paper is to compare syntactic, semantic and structural aspects of two models. The development of our application is done in Java because this language allows the use of several APIs for manipulating OWL ontologies as Jena (http://jena.sourceforge.net/) and Sesame (http://jena.sourceforge.net/). Other resources (dictionaries) are managed in tables. We are currently achieving an interactive user interface of our system. In fact, the user validates or delete mappings created. Validated correspondences will be stored in a MySQL database. The integration can be applied to "n" models \( M_i = \{M_i \mid i=1..n \} \). In this case, we can integrate \( M_1 \) and \( M_2 \), then integrate their result model \( MR_{1.2} \) and \( M_3 \), etc..., until the \( M_n \) model. Our research will be a further study on the definition of rules of integration and fusion, which will thus enable to realize the whole process of model integration.

REFERENCES


Benabdellah, C. S. and Fredj, M. 2010. Vers une contribution à l’intégration sémantique des modèles UML. ERATSI, INFORSID, France (Benabdellah and al., 2010b)

Benabdellah, C. S., Fredj, M., Mouline, S., 2010. Un système à base de règles d’aide à la décision pour la comparaison sémantique des modèles”. SITA 2010, Rabat (Benabdellah and al., 2010a)


Ferut, T., June 2006. Définition d'un mécanisme de composition appliqué aux modèles métiers, Stage du Master PLMT, Laboratoire IJS, équipe OCL.


AN HYBRYD APPROACH FOR MODELS COMPARISON


