ABSTRACT

Today, enterprise application integration is still facing many problems and the crucial one is the semantic heterogeneity. This problem is not adequately addressed by today’s solutions that focus mainly on the technical and syntactical integration. Using the semantic aspect will promote enterprise applications by providing it more consistency and interoperability. The development of ontologies to encapsulate the applications heterogeneity may produce new obstacles for application ontologies integration. This article will propose a mapping process in order to address the semantic integration problem. Ontology mapping is important when working with more than one ontology. Typically similarity considerations are the basis for this. In this paper an approach to integrate various similarity strategies is presented. In brief, we determine also similarity through rules which have been derived by ontology experts.

Keywords: Enterprise Application Integration, Application ontology, Mapping, Similarity Rules.

1. INTRODUCTION

One of the ambitious challenges of the enterprise integration is for enterprise application systems to be able to exchange meaningful information and services with one another in semantically rich and sound ways. This problem is often referred to as the interoperability problem. It is defined by ISO/IEC 2382 as “the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units”. Interoperability can mainly occur at two fundamental levels: the syntactic and the semantic level that resolve respectively syntactic and semantic conflicts [6].

Recently, a new technology and a new industry sector, typically known as EAI (Enterprise Application Integration) technology, have emerged [9]. In essence, EAI technologies provide tools to interconnect multiple and heterogeneous EIS (Enterprise Information Systems) such as ERP (Enterprise Resource Planning) systems, CRM (Customer Relationship Management) systems, SCM (Supply Chain Relationship Management) systems, and legacy systems. This interconnection is difficult to achieve because integrated systems are developed independently [7]. However, these applications do not share the same semantics for the terminology of their applications models, which may lead to inconsistent interpretations.

The Semantic Web community has achieved a good standing within the last years. As more and more people get involved, many individual ontologies are created. Interoperability among different ontologies become essential to gain from the power of the Semantic Web. Thus, mapping and merging of ontologies becomes a core question. As one can easily imagine, this can not be done manually beyond a certain complexity, size, or number of ontologies any longer. Automatic or at least semi-automatic techniques have to be developed to reduce the burden of manual creation and maintenance of mappings.

Some work relating to the automatic and semi-automatic mapping of ontologies [13] was recently published. Authors have focused on the linguistic analysis of ontologies elements information (concept name, properties, taxonomy …) [14], the application of heuristics rules [15], the exploitation of machine learning techniques [16] and Bayesian network [17]. Despite the large number of related work, there are very little approaches on how to combine similarity measurement strategies and rules as we do. The obtained results are more pertinent and more reliable because we use the context notion for the mapping discovery.

We have developed a set of ontologies intended to capture the semantics for applications integration. These ontologies are part of our integration architecture. We propose an ontology mapping process to build the enterprise ontology (mapping ontology). Here, ontology mapping is important because we have more than one ontology. Typically, similarity measurement strategies become necessary. For this purpose, we adopt some similarity methods and mapping rules derived by human experts to find mapping candidates between two or more application ontologies.

The structure of the paper is as follows: Section 2 shows our application integration architecture and gives a description of the core mapper for bridging application ontologies and its main components. Next section explains the four major steps of our mapping process for application ontologies integration. Finally, we conclude our paper and sketch some future work.
concerning ontology evolution should be regarded next steps in our work.

2. ONTOLOGY BASED INTEGRATION ARCHITECTURE

The integration system we propose, aims at offering a support for integrating heterogeneous and distributed applications, accessing multiple ontologies. It includes both type of integration, internal (application to application) and external (business to business) ensuring by applicative level and collaborative one respectively [2].

- Applicative level consists of heterogeneous and distributed applications. Each application has its own local ontology. In our architecture, we aim to overcome the gap between local application ontologies, according to the semantic relations. A special component, named mapper, is invoked to perform its tasks for building the global ontology. The latter can be seen as an enterprise ontology.

- Collaborative level takes place in the business process collaboration with partners. Each company has a mobile agent that is responsible for requesting and providing the services and the negotiation for selecting the best partner. It uses the collaboration scenario for achieving business process (Figure. 1).

Figure 1: Integration system architecture.

Additionally, an overview about the core mapper is given in the following. It is composed of two modules:

*a-Mapping module*

It includes three components:

1. *The identifier* is used to find related concepts or attributes of ontologies and the relations between them. This can be done automatically, semi automatically or manually with the help of domain experts [5]. For instance, it can use lexical ontologies such as WordNet [4] or domain specific thesaurus to define synonym terms, antonym terms and encoding (e.g. miles/kilometers, Fahrenheit/Celsius, hours/minutes).

2. *The adaptor* is used to represent the identified relations of equivalence between ontologies based on the mapping process. It combines many strategies to measure the similarity. It uses a similarity model to compute the concepts similarity at various levels, such as lexical one, properties (roles and attributes), and hierarchical one. It combines mapping rules to obtain better mapping results.

3. *The linker* transforms instances from the source application ontology into instances of the target application ontology by evaluating the equivalence relations defined earlier by the adaptor. Two problems that may arise are that the mappings are incomplete or that the mapped entities differ in the context. The missing mappings can be gained through inference mechanism.

Figure 2: Mapper architecture.

*b-Evolution module*

This module is based on the capacity to manage the changes of the ontology and their effects by creating and maintaining various versions of ontology. Its role consists to identify, differentiate, and modify the versions and to specify relations which make explicit the changes carried out between versions. It contains two components:

1. *The versionnor* is used to detect new relation of mapping between the old version of evolved ontology and the new one using the evolution journal. Then, it composes it (the new mapping) with the existing
mappings between the old version ontology and connected ones.

2. The handler tries to make the relations of mapping consistent by the elimination of the invalid correspondences.

In this paper, we give more attention to the mapping module in the core mapper. The evolution module will be detailed later.

3. MAPPING PROCESS
Application ontology that we have developed concerns the EAI domain. For this purpose, we specify well the concepts relating to this domain and relations between them. These concepts must describe various types of applications, their models, their structures and the domain to which applications belong. These concepts are inspired from EAI domain [9], Web services [24] and middleware technologies [23]. The application ontology consists of a classification of relevant characteristics of applications. We have some pertinent information about the application, such as application-behaviour, application-domain, application-structure …etc.

Our application ontology building process contains four steps: meta-modeling application, formalization, implementation and adaptation [2]. It consists in a classification of relevant applications characteristics whereby each categorized class is linked with associated information. Then, the formalization implies the representation of the ontology in a formal language. The implementation builds computable model and checking consistency.

In order to carry out ontology based semantic integration, we unify the heterogeneous applications by integrating (mapping, merging, aligning …) the associated ontologies. The only really realistic possibility is the mapping approach because in this case the integrated ontologies are not affected. For this purpose, in this work we give more attention to the ontology mapping [13].

Many expressions have been used to define the mapping term. We want to describe our understanding of the term “mapping”. We define mapping as cf. [26]: “Given two ontologies A and B, mapping one ontology with another means that for each concept in ontology A, we try to find a corresponding concept, which has the same or similar semantics, in ontology B and vice versa.”

Ontology mapping is the process whereby semantic relations are defined between two ontologies for transforming source ontology instances into target ontology instances. In this paper, we propose a mapping process which includes the following principal steps: filtration, similarity measurement, bridging and inference achieved by the core mapper (mapping module). The principal task of the latter is to discover the semantic relations between application ontologies concepts. The identifier is used to detect the lexical similarity between the concepts of ontology source and target ontology. With obtaining a semantic relation of synonymy, the adapter deals with the other similarities to calculate the total similarity. Lastly, the connector carries out the bridging and calls upon the mechanism of inference in the absence of the target concept.

3.1FILTRATION STEP
In the most mapping systems, the concept name plays a significant role in the concepts similarity measure. With the use of WordNet, the concept name can belong to one or several sets of the synonyms, called synsets. These categories are large, where two words can belong to one or more synsets, even if in a given context, they do not share the same meaning. It thus becomes necessary to introduce the context notion into ontologies in order to solve this problem.

According to Brézillon [18], the context is seen as "the whole of the conditions and environment influences which make situation, a unique one and make it possible to understand it"

For [19], the usual definition of the context term includes "any internal or external element, relating to the application or the user, or even completely external, which could modify the interaction progress"

In our work, the context notion must contain the knowledge which makes it possible to express the circumstance of a concept, its role in the ontology and its use cases, it is not a question of the inherent characteristics of the concept, but a definition related to the situation where it is used to describe the various semantic aspects. With the use of context notion, the obtained results shall be more pertinent and more reliable.

In general, the use of context notion makes it possible to personalize and adapt the applications to different environments [20].

3.1.1 Semantic enrichment by the context
Generally, concepts have comments expressed in natural language. These comments can for example be definitions taken in a dictionary. We will enrich these
comments by contextual information expressed in natural language.

Example
In the information system of a hospital, there are two ontologies, the first for the bacteriology laboratory, and the second for the human resources service.

The two ontologies contain the common concept “means of access”, which is defined as the tool or the method used to move and change the place. If one wishes to integrate the two ontologies, and to calculate the similarity between these two concepts. The existing techniques will consider them similar even if actually, they are not.

It thus becomes necessary to introduce the context notion into ontologies to avoid this kind of problem. For the concept “means of access” in the first ontology, we add to the comment, the following information: a bacterium can move by blood, by air…, and in the second: a person can used her personal vehicle, the public transport…, to go to his work.

Each context of a concept is regarded as a document, and all the contexts in ontologies are treated as a collection of documents.

3.1.2 Determination of candidate concepts
After enrichment by the context, the next phase consists in comparing the contexts between them to find the candidate concepts. To each concept of the first ontology a list of concepts of the second ontology will correspond.

Let us two ontologies, having N and M concepts respectively, if we use P strategies of similarity measurement, the number of times that we compute the similarity is N*M*P. The use of context constitutes a first filter which will decrease the set of candidate concepts. The number of times that we calculate the similarity becomes equal to N * M + P * ∑_{i=1}^{N} M_i. Knowing that M_i is the number of the concepts of the second ontology most similar to concept i of the first ontology (M_i ≤ M). So, N*M is the number of times that we calculate the similarity in the first step of our process, where P * ∑_{i=1}^{N} M_i represents the number of times of the second step.

To compare the concepts contexts, we use the techniques of the information search. Indeed, each context is seen as a document. To obtain similarity between contexts is to find the most adequate document of the first collection in the second.

A document can represent as a vector according to the vector space model [21]. Each document d in the collection of the documents D is represented by a vector of characteristics. Each coordinate d_w is calculated basing on the frequency occ(w, d) of the word w in d according to the following formula:

\[ d_w = \text{TFIDF}(w, d) = \log(1 + \text{occ}(w, d)) \times \log \left( \frac{N}{N(w)} \right) \]

Where N is the number of documents of the collection, and N(w) the number of documents in which w appears at least once. It thus will allow a weight proportional to the frequency of appearance in the document. All the vectors of characteristics are standardized thereafter to have the same length. The used similarity measurement is:

\[ \text{sim}_{d_1,d_2} = \frac{\text{sim}(V_{d_1w}, V_{d_2w})}{\| V_{d_1w} \| \times \| V_{d_2w} \|} = \cos \theta = \frac{V_{d_1w} \cdot V_{d_2w}}{\| V_{d_1w} \| \times \| V_{d_2w} \|} \]

\( \theta \): It is the angle between the two vectors.

\( V_{d_1w} \) and \( V_{d_2w} \): Vectors of characteristics of documents \( d_1 \) and \( d_2 \).

According to the chosen threshold, only the concepts having contexts have values of similarity higher than this threshold can have passed at the second step.

3.2. SIMILARITY MEASUREMENT STEP
In this step, each concept A of source ontology is compared to concept B of target ontology to determine the semantic similarity. We propose a similarity model which calculates the similarity between ontologies entities combining various measurements. The first concentrates on the acquisition of a lexical similarity based on WordNet [4]. The second is interested in the informational contents to measure the semantic distance between the concepts [11]. The third computes the similarity between concepts based on their properties (attributes and roles). Then, we propagate the similarity with other parts of taxonomy, the super (ancestors) and the sub (sons) concepts. To be able to apply these measurements, the application ontologies graphs (hierarchies) are linked by a meta-class in order to calculate the distance between concepts in the global graph.

The suggested similarity model includes several aspects (lexical, concepts, properties and distance in the graph) which can influence the similarity value, instead of concentrating only on lexical comparison between element names in different ontologies. The names can be abbreviations, acronyms, phrases, in different languages. So, using one aspect of the concepts can be more or less representative. For instance, the similarity measure which takes only the position of concepts in the graph or their information content that is defined in a probabilistic way appears limited in the frame of the mapping process.
3.2.1 Lexical similarity

WordNet can be seen as a semantic network where each node represents a concept of the real world [4]. Each node is composed of a set of synonyms which represent the same concept. This unit is called synset. The synsets are connected by arcs which describe the relations between concepts. The idea is that two concepts are semantically related if their synsets are connected by at least one way.

To calculate the lexical similarity between the names of concepts, we use the linguistic comparison based on the WordNet thesaurus and the edition distance.

Pantel & al. [1] define the similarity between two words, S1 and S2 in WordNet by:

$$sim_{\text{wn}}(S1, S2) = \frac{\log p(S)}{\log p(S1) + \log p(S2)}$$

Such as:

- $P(S) = \frac{\text{Count}(S)}{\text{total}}$ is the probability of the word occurrence in the synset S, or any subsynset which constitutes it.
- total is the number of words of WordNet.
- Count(S) is the number of words in S and its subsynsets.
- The synset is the common synonym of S1 and S2 in WordNet.

J. Tang [25] defines similarity between two concepts by the maximum similarity between their meanings. It is described by:

$$sim_{\text{str}}(C1, C2) = \max (sim_{\text{str}}(S_{1i}, S_{2j}))$$

$S_{1i}$ and $S_{2j}$ belong respectively to S (C1) and S (C2).

By using the edition distance between the names (character strings) which makes it possible to minimize the distances between the character strings.

The edition distance between C1 and C2 determines how many operations are used to transform C1 into C2 [10]. These operations are:

- Substitution: a symbol of C1 is replaced by the corresponding symbol of C2.
- Insertion: a symbol of C2 is inserted in C1.
- Suppression: a symbol is removed in C1.

The edition distance value (disted) is equal to the minimum number of the operations to pass from C1 to C2. disted is given by [3]:

$$\text{Disted} (C1, C2) = \min \text{ number of substitution } \times \text{cost of substitution,}$$

$$\text{number of insertion } \times \text{cost of insertion,}$$

$$\text{number of suppression } \times \text{cost of suppression}$$

The values of the various costs are given using the heuristic base; the cost of substitution is zero for example. We note that disted can exceed value 1. The similarity of edition (simed) is given by the formula:

$$sim_{\text{ed}}(C1, C2) = \frac{1}{\text{disted}(C1, C2)}$$

The exponential function gives values ranging between [0, $\infty$]. The value of the similarity consequently included in [0, 1], simed is inversely proportional to disted.

Formally, we can define the lexical similarity (Lsim) between the names of concepts C1 and C2 by:

$$Lsim_{\text{wn}}(C1, C2) = \frac{\text{sim}_{\text{str}}(C1, C2) + \text{sim}_{\text{ed}}(C1, C2)}{2}$$

3.2.2 Concept similarity

The concept similarity (Csim) is similar to Lsim with the exception of the ontology taxonomic structure. Here, the notion of $IC$ (Information Content) is introduced [11]. The IC of a concept $C$ is defined as:

$$IC(C) = -\log (P(C)).$$

Intuitively as $p$ increases, the informativeness of the concept $C$ decreases. $P(C)$ denotes the probability of
instance occurrence of a concept C. This probability is calculated by: \( \text{frequency}(C)/N \) where \( N \) is the total number of concepts instances.

We use the Jiang-Conrath measure \([8]\) to compute the similarity between concepts. This measure takes into account the IC of the smallest concept \( S \) that subsumes both \( C_1 \) and \( C_2 \). The distance is defined by:

\[
\text{Distance} (C_1, C_2) = IC (C_1) + IC (C_2) - (2*IC(S)).
\]

So, the concept similarity is:

\[
C_{\text{sim}} (C_1, C_2) = 1/\text{Distance} (C_1, C_2)
\]

### 3.2.3 Property similarity

The property similarity \((P_{\text{sim}})\) is related to the roles and the attributes of concepts.

- **Role similarity**: It measures the similarity between the roles of concepts. Each role of the concept \( C_1 \) is compared to each role of the concept \( C_2 \). A Role matrix \( M_{\text{role}} (N, M) \) of the distances from roles is thus built. The columns represent the roles of \( C_2 \), and lines show the roles of \( C_1 \). \( N \) and \( M \) represent respectively the number of roles of the concept \( C_1 \) and \( C_2 \). The role similarity is calculated by:

\[
sim_{\text{role}} (C_1, C_2) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} M_{\text{role}}[i, j]
\]

Where, \( M_{\text{role}} [i, j] = \text{Lsim} (C_1.r_i, C_2.r_j) \)

- **Attribute similarity**: It measures similarity between concepts attributes. Each attribute of the concept \( C_1 \) is compared with each attribute of the concept \( C_2 \). An Attribute matrix \( M_{\text{att}} (N, M) \) of the distances from attributes is thus built. The columns represent the attributes of \( C_2 \), and lines show the attributes of \( C_1 \). \( N \) and \( M \) represent respectively the number of attributes of concepts \( C_1 \) and \( C_2 \). The role similarity is calculated by:

\[
sim_{\text{att}} (C_1, C_2) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} M_{\text{att}}[i, j]
\]

Where, \( M_{\text{att}} [i, j] = \text{Lsim} (C_1.a_i, C_2.a_j) \)

The property similarity is the composition of the elementary similarities \((\text{sim}_{\text{role}}, \text{sim}_{\text{att}})\) weighted by coefficients \((w_\text{role}, w_\text{att})\) respectively.

\[
\text{Psim}(C_1, C_2) = \frac{(w_\text{role} \times \text{sim}_{\text{role}} + w_\text{att} \times \text{sim}_{\text{att}})}{w_\text{role} + w_\text{att}}
\]

### 3.2.4 Taxonomic similarity

The ontology taxonomy is a graph, where the root is a hierarchical concept and the nodes are the possible values that can take this concept. It is used to express the relations between the concept values basing on their position in the taxonomy.

We adopt the ontology taxonomy to extract information relating to the concept, the set of its ancestors (super concepts) and its sons (sub concepts).

In the propagation theory \([22]\), the concept semantic is carried by their ancestors and sons. For this purpose, we associate to each concept of taxonomy a net value to define the granularity level of compared taxonomies. The net is given by the following formula:

\[
\text{net}T = \frac{\text{prf}(C)}{\text{nv}(C)}
\]

\( \text{net}T \) is the net value associated to the concept \( C \) in taxonomy \( T \).

\( \text{prf}(C) \) is the level of the concept \( C \) in taxonomy \( T \).

\( \text{nv}(C) \) is the depth of taxonomy \( T \).

Concepts having the same level have the same net.

- **Ancestors similarity**: We build an ancestor matrix \( M_{\text{anc}} (K, p) \), where columns represent the ancestors of \( C_2 \), and lines the ancestors of \( C_1 \). Then, we calculate the similarity between the ancestors (similarity between their names). \( K \) and \( p \) represent respectively the number of the ancestors of concepts \( C_1 \) and \( C_2 \).

We associate to each ancestor the net value calculated previously, and then we calculate \( \text{sim}_{\text{anc}} \) according to the following formula:

\[
\text{sim}_{\text{anc}}(C_1, C_2) = \frac{\sum_{i=1}^{p} \sum_{j=1}^{p} \text{net}_{i} \times \text{net}_{j} \times M_{\text{anc}}[i, j]}{\sum_{i=1}^{p} \sum_{j=1}^{p} \text{net}_{i} \times \text{net}_{j}}
\]

\( \text{net}_i \) is the net value associated to the ancestor \( \text{anc}_i \) of the concept \( C_1 \).

\( \text{net}_j \) is the net value associated to the ancestor \( \text{anc}_j \) of the concept \( C_2 \).

\( M_{\text{anc}}[\text{anc}_i, \text{anc}_j] \) is an element of the ancestor matrix, containing \( \text{sim}_{\text{name}}(\text{anc}_i, \text{anc}_j) \).

- **Sons similarity**: \( \text{Sim}_{\text{sons}} \) is calculated in the same manner as \( \text{sim}_{\text{anc}} \). The taxonomic similarity \((\text{Tsim})\) is the composition of the elementary similarities \((\text{sim}_{\text{sons}}, \text{sim}_{\text{anc}})\), balanced by coefficient. It is given by the following formula:

\[
\text{Tsim}(C_1, C_2) = \frac{w_{\text{anc}} \times \text{sim}_{\text{anc}} + w_{\text{sons}} \times \text{sim}_{\text{sons}}}{w_{\text{anc}} + w_{\text{sons}}}
\]

### 3.2.5 Global similarity

The values of partial similarities obtained previously, will be balanced by coefficients. For example, the coefficient associated to the \( \text{Lsim} \) is given by the following formula:

\[
\frac{\text{Lsim}}{W_L} = e
\]
The other coefficients are calculated in the same manner. Therefore, the global similarity ($Gsim$) is given by:

$$Gsim(C1,C2) = \frac{WL \times Lsim + WC \times Csim + WF \times Fsim + WT \times Tsim}{WL + WC + WF + WT}$$

Concepts where $Gsim$ is larger than a threshold value are similar.

### 3.3. BRIDGING STEP

This step is responsible for establishing correspondence between entities based on the similarities computed previously. Each concept according to the source ontology is translated into the most similar concept described according to the target ontology. It intends to associate a transformation procedure. There are three dimensions of bridging:

**Discovering**: A semantic relation is defined between the concept $C_{ik}$ of source ontology $Oi$ is the concept most similar $C_{jl}$ of target ontology $Oj$. The semantic relations established by means of the calculated similarity make it possible to create a SCM (Semantic Corresponding Matrix) which could be used by the inference mechanism.

For each couple of ontologies, we build an SCM ($N \times M$) matrix. $N$ is the concepts number of source ontology and $M$ is the concepts number of target ontology.

Where, $SCM\{C_{ik}, C_{jl}\} = Gsim\{C_{ik}, C_{jl}\}$.

If $Gsim$ of concepts are low than a threshold value, we try to use mapping rules to obtain similarity. We give some mapping rules to define similarity: A combination of rules leads to better mapping results compared to using only similarity measurement.

**R1**: If super-concepts are the same, the compared concepts are similar to each other.

**R2**: If sub-concepts are the same, the compared concepts are similar to each other.

**R3**: If the domain and range of concepts properties are equal, the concepts are also.

**R4**: Concepts that have the same instances are the same.

**R5**: Concepts having common similar properties are similar.

**R6**: If the synsets of concepts have at least a way of terminological relation in WordNet, concepts are also synonymous.

**R7**: Concepts that have the same human expert definition, their context documents are also the same.

...

Similarities which are too little (i.e. below the threshold value) are removed and only best similarities are guarded in the SCM matrix.

**Representation**: is used to represent semantic relations between ontologies. In our case, we use OWL language.

**Execution**: is used to transform instances of source ontology to instances of target ontology. The bridging operation proceeds according to the following algorithm.

Map: is a triangular matrix / where

Map[i, j] = (Oi, Oj) and Map[i, i] = 0.

Map[i, j] = (Oi, Oj) the couple of ontologies to bridge.

N is the concepts number of source ontology

M is the concepts number of target ontology.

$t$: the threshold value

Begin

For i = 1 to N Do

For j = 1 to M Do

Map[i, j] = 0

Create SCM (Oi, Oj)

Endfor

Endfor

Repeat

For k = 1 to N Do

For l = 1 to M Do

If (MCS[C_{ik}, C_{jl}] > t) then

-SCM[C_{ik}, C_{jl}] = 1/*do mapping*/

-Transform instances (IC_{ik}, IC_{jl})

Else

-SCM[C_{ik}, C_{jl}] = 0 /(*no mapping*)

-Invoke the inference mechanism

EndIf

EndFor

EndFor

Until Map is null

End.

**Procedure** Transform instances (C1, C2)

If (SCM[C1, C2] = 1) then

-add the set of instances of concept C1 to the set of instances of concept C2 and to the set of instances of its subsuming concepts if they exist.

EndIf

**3.4. INFERENCE STEP**

The inference mechanism is used when the source concept has not a target concept. It means that the source concept has not a direct counterpart in the target ontology. For example, the concept 4-stars room may differ between several tourism systems. According to the given facilities of the room, one tourism system may categorize the room as 4-stars another only as 3-stars because of the lack of certain services [5].

By using the SCM, the mechanism of inference can detect the concerned concepts. SCM[C_{ik}, C_{jl}] = 0 implies that the source concept $C_{ik}$ of the ontology $Oi$
does not have a target concept. In this case, we try to exploit the mapping rules base in order to obtain the target similar concept. So, steps two and three are repeated a fixed number of times.

4. CONCLUSION
This work reported an ontology mapping process with the aim of integrating heterogeneous applications in the context of EAI. The mapping process includes four major steps that are filtration, similarity measurement, bridging and finally inference. The first one tries to research by the context which constitutes a first filter. The second step uses some metrics for calculating the semantic similarity of concepts basing on their properties. The third one contains three substeps, discovering, representation and execution for transforming instances of source concept to instances of target concept. The fourth one is the inference which calls upon the two steps to compute again the concept similarity in order to find its target concept.

The results obtained are more pertinent and more reliable because we use the context notion for mapping discovery.

For mapping process, we will try to extend the similarity measurement model for supporting instance classification basing on Bayes decision theory and exploits the constraints expressed in ontologies [17]. We will exploit also the techniques of TAL (treatment automatic of language) and of standardization in the first step of mapping process.

For future research, ontology evolution area will have potential. We will develop evolution ontology to describe domain evolution and master ontologies versions.

REFERENCES