

# A contextual semantic mediator for a distributed cooperative maintenance platform

Mohamed-hedi Karray, Brigitte Chebel-Morello, Noureddine Zerhouni  
Femto-st institute, Automatic control & Micro-Mechatronic Systems department (AS2M)  
{hedi.karray, Brigitte.morello, Noureddine.zerhouni}@femto-st.fr

**Abstract-**Platforms expand maintenance systems from centralized systems into e-maintenance platforms integrating various cooperative distributed systems and maintenance applications. This phenomenon allowed an evolution in services offered to maintenance actors by integrating more intelligent applications, providing decision support and facilitating the access to needed data. To manage this evolution, e-maintenance platforms must respond to a great challenge which is ensuring an interoperable communication between its integrated systems. By combining different techniques used in previous works, we propose in this work a semantic mediator system ensuring a high level of interoperability between systems in the maintenance platform.

**Key words:** e-maintenance platforms, semantic mediation, ontology, interoperability.

## I. INTRODUCTION

While systems' integration and collaboration are believed to be the key enabling technologies driving maintenance system in improving productivity and efficiency, concept of e-maintenance follows the idea of application integration within the context of maintenance - Maintenance Application Integration (MAI). In fact, because of the complexity of the maintenance process, and the use of heterogeneous software and hardware systems/tools, systems integration becomes an important prerequisite to achieve efficient and effective collaboration in maintenance. In fact, systems integration is all about interoperability [1].

In the scope of the European project SMAC<sup>1</sup> (Semantic MAintenance and life Cycle), we are developing a distributed cooperative platform of maintenance. The main goal of this platform is to provide the means for moving from coexistence to the interoperability and cooperation of these applications within the same environment. The interoperability should provide the condition for orchestration of components in order to provide a global integrated service for the platform user(s). The integration is necessarily done by the exchange of data between the components.

When the platform is based on knowledge and semantics, we have developed in previous work [2] a common ontology of maintenance domain. This Ontology will be tacked as a reference by systems which communicate via the platform. These systems create their local ontologies based on the common Ontology of maintenance. These Local ontologies grow independently from each other. In purpose that ensure semantic interoperability between these systems, the platform must take into account the evolution of the local ontologies of

its integrated systems. Hence, we set up a semantic mediator system allowing an alignment between the different versions of ontologies of these different systems.

A mediator is defined by Wiederhold [3] as a software module that exploits encoded knowledge about a particular dataset to bring the source information into a common form for a higher layer of applications. Therefore, a mediator can provide value-added services (called semantic mediation services), such as (1) accessing and retrieving relevant data from multiple heterogeneous resources and (2) abstracting and transforming retrieved data into common representations and semantics [4]. Thus, the term semantic mediator refers to a software agent that is responsible for semantic reconciliation, that is, the identification and resolution process of semantic conflicts in the maintenance platform.

Since the effective construction of mediators requires some common representation of the meanings (i.e., a common Ontology) of the resources and applications to which they connect [5], we have employed our developed Ontology of maintenance. The used Ontology is presented in PowerLOOM [6] a description language based on descriptions logics to benefit from its reasoning capacities in the mediator system. This latter is context based [7] and it uses a similarity algorithm. We developed this algorithm to ensure the ontology alignment feature [8].

Therefore we present in this paper the architecture of the mediator system, its functioning manner and its various components as well as the similarity algorithm.

The remainder of this paper is organized as follows. Section two is devoted to present an overview about various existent mediator systems. The following section is devoted to present two use cases of maintenance management as motivating example. Section four and five are devoted to present the architecture and the functioning of the semantic mediator system and to discuss and evaluates the mediator system. Finally, a review and conclusions are developed in the last section.

## II. OVERVIEW

In our knowledge semantic mediation is not enough investigated in the scope of the e-maintenance platforms, Enterprise Asset Management Systems or CMMS (Computerized Maintenance Management System). Most of these latter use web services technologies to ensure a structural level of interoperability [2]. Therefore, we will focus in this section on the presentation of some mediator

<sup>1</sup> European project INTERREG IV (France and Switzerland).

systems generally developed in different computer science areas.

In fact, many research has been devoted to the problem of semantic mediation [9,10,11]. This latter adds some additional considerations to the integration problem.

Several prototype mediator architectures have been designed by projects. First generation of these mediator systems as TSIMMIS [12], Information Manifold [13], Garlic [14], and MIX [15] achieves integration mainly on a structural level [16].

The most prominent approaches are TSIMMIS and Information Manifold. TSIMMIS [12] is based on the semi-structured data model OEM and uses the logic-based language MSL for defining global views and formulating queries. One of the successors of TSIMMIS is MIX [15] which is an XML based mediator, i.e., it uses XML for representing instance and schema information and supports the query and view definition language XMAS, which is based on ideas from XML-QL and MSL.

These approaches focus generally on structural and schema aspects; however the problem of semantic mediation has also been addressed. In contrast, approaches based on semantic mediation represent domain knowledge explicitly by semantic relationships and constraints and they exploit this information for the formulation and the processing of queries [16].

As such system, we find STYX [17] an XML mediator which follows the LAV [13] principle and uses an Ontology as integration model. In the DIKE system [18], the focus is on automatic extraction of mappings between semantically analogous elements from different schemas. ODB-Tools [19] is a system developed on top of the MOMIS [20] system for modeling and reasoning about the common knowledge between two to-be-integrated schemas. They present the object-oriented language ODLI3 derived from a description logic (OCDL). Another example of such system is KIND [21], where domain knowledge is represented by so-called domain maps based on a subset of F-logics. Calvanese et al [22] performed semantic information integration using an LAV approach by expressing the conceptual schema by a description logic language. The semantic integration system SIMS [23] is based on the knowledge representation language Loom by presenting the domain model using “*is-a*” relationships between local and global concepts. The Context Mediator presented in [7] also uses a domain model. However, in this case it comprises a set of primitive and semantic types. Instances of semantic types may have different values in different contexts.

By browsing these various works, we note the essential use of domain knowledge especially Ontologies, as well as the use of description logics. In fact what they provide in terms of expressiveness and reasoning capacities.

These observations argue our taken decisions to build the semantic mediator system for the platform of maintenance management.

According to Wache et al in [24] there are three directions that are employed to use Ontology-based integration

approach. These directions are: *single ontology approaches*, *multiple ontologies approaches* and *hybrid approaches* (see Fig. 1).

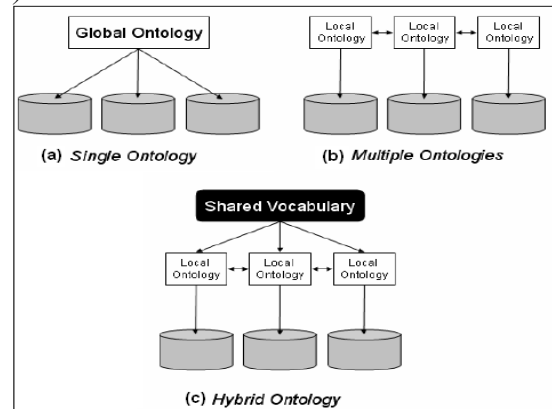


Fig. 1. Ways for using ontologies for content explication. It illustrates the three approaches used on Ontology based integration. Taken from [24].

Despite that many mediator systems use the single ontology approach (Fig. 1 a) as SIMS or the multiple ontology approach (Fig. 1 b) as OBSERVER [25], we adopt the hybrid approach (Fig. 1 c). This latter was developed to overcome the drawbacks of the single or multiple ontology approaches. Similar to multiple ontology approaches the semantics of each source is described by its own ontology. But in order to make the local ontologies comparable to each other they are built from a global shared vocabulary [24]. Generally the shared vocabulary is a general ontology which covers all possible refinements like in BUSTER [26]. The advantage of a hybrid approach is that new sources can easily be added without the need of modification. It also supports the acquisition and evolution of ontologies. The use of a shared vocabulary makes the source ontologies comparable and avoids the disadvantages of multiple ontology approaches.

This approach seems the most adapted to our context and in concordance with our previous work. We developed the common Ontology of maintenance to serve as the shared vocabulary and the platform gives the possibility to its integrated application to develop their local ontology in accordance with their needs.

### III. MOTIVATING EXAMPLE

We present in this section two use cases to show the functioning of the mediator system and the importance to set up this system as a component in the maintenance platform.

Let's consider the parts describing human resources and intervention in the maintenance domain Ontology exposed in Fig. 2. As shown, the Ontology does not consider human resources competencies or roles.

That to say S1 and S2 two systems integrated in the maintenance platform. S1 is a system managing interventions in a maintenance site (MS). S2 is a system of an enterprise E1 providing human resources needed by MS. S1 and S2 use the shared Ontology of maintenance as base to develop their local ontologies; each system in each site enriches the Ontology to adapt it to their needs.

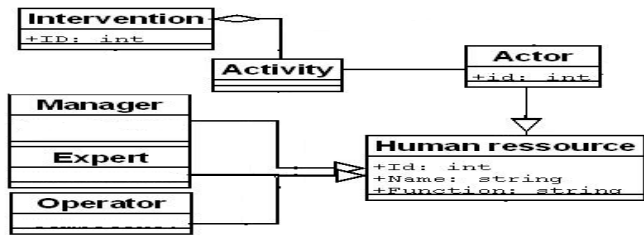


Fig. 2. Maintenance Ontology: Shared Human resource part.

To adapt the Ontology to their needs, operators of the maintenance system in S1 add the concept “Expertise” to the local ontology used by their system like shown in Fig. 3. Expertise is related to the concept actor by a relation  $R1$ :  $Actor\_has\_Expertise$ .

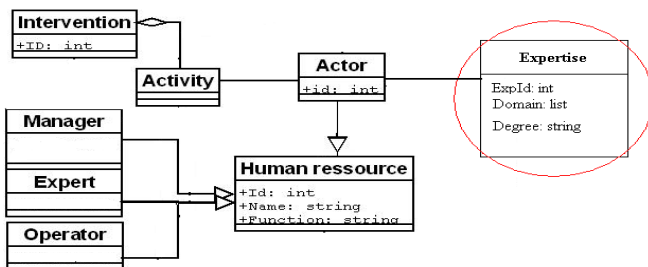


Fig. 3. S1 has its local version of the human resource Ontology part developed by its operators. The added concept Expertise is surrounded with red line.

The same action was done on S2; human resource managers of E1 add the concept “COMPETENCY” related to Human-resource concept by the relation HUMAN-RESOURCHE-HAS-COMPETENCY as presented in this part of PowerLOOM ontology:

```
(DEFCONCEPT COMPETENCY)
(DEFRELATION HUMAN-RESOURCHE-HAS-COMPETENCY ((?HR
HUMAN-RESOURCHE) (?C COMPETENCY)))
(DEFRELATION COMPETENCY-AREA ((?C CAMPETANCY)
(?AREA STRING)))
(DEFRELATION COMPETENCY-EXPERIENCE ((?C CAMPETANCY)
(?EXPERIENCE STRING)))
(DEFRELATION COMPETENCY-DIPLOMA ((?C CAMPETANCY)
(?DIPLOMA STRING)))
(DEFRELATION COMPETENCY-QUALIFICATION ((?C
CAMPETANCY) (?QUALIFICATION STRING)))
(DEFRELATION COMPETENCY-REFERNE ((?C CAMPETANCY)
(?COMPETENCY-REFERENCE INTEGER)))
```

SM needs an ACTOR having the expertise EES (experience in electrical engineering domain with high level degree of experience (senior)). Via the maintenance platform, S1 asks for this specific actor from S2. It is noteworthy, S2 did not know about Expertise and S1 hasn’t any knowledge about the Competency. This communication will be managed by the platform through its mediator system to ensure semantic interoperability between these two systems.

The second example is about two systems Sys1 and Sys2 when Sys1 is an events management system used by maintenance experts. It uses the shared Ontology of maintenance especially the part shown in Fig. 4. Sys2 is a monitoring system implanted in the production site and it

uses as local ontology an enhanced version of the shared Ontology as presented in Fig. 5.

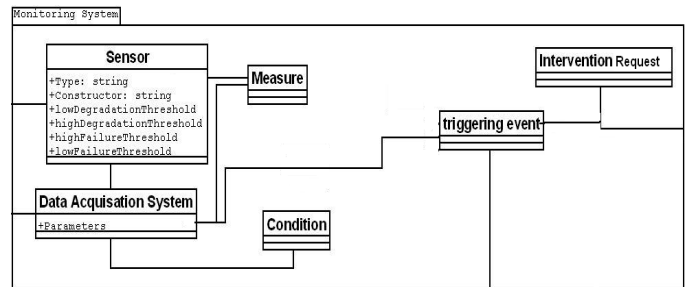
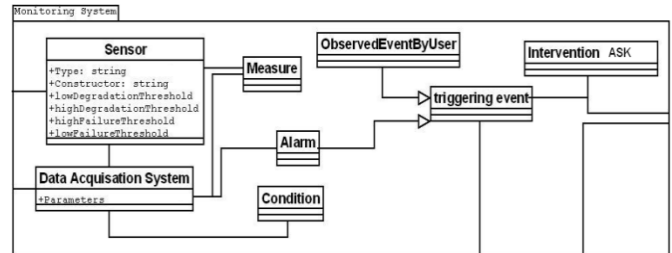


Fig. 4. Maintenance Ontology: Shared Monitoring part.



```
(DEFCONCEPT ALARM (?T TRIGGERING-EVENT))
(DEFCONCEPT ALARM-OBSERVED-BY-USER (?T TRIGGERING-EVENT))
(DEFRELATION TRIGGER-ALARM ((?DAS DATA-
ACQUISITION-
SYSTEM) (?A ALARM)))
```

Fig. 5. Sys2 develops his local version of monitoring Ontology part. The figure contains the PowerLOOM definitions of the added concepts and relations.

#### IV. A CONTEXTUAL SEMANTIC MEDIATION SYSTEM

##### A. Our proposal

To build the mediator system, we use some techniques inspired from different works and we introduce some notions adapted to the scope of maintenance’s domain. Firstly, this mediator system is context based; inspired from [28] where authors provide a context model for semantic mediation in web services. Also, in [7] authors present the Context Interchange strategy as good feature for mediated data access in which semantic conflicts among heterogeneous systems. Next section will be devoted to present notions that we developed to take into contexts of concepts. Secondly, consistent with the trend of the use of description logics languages mentioned above, we adopt PowerLOOM - the successor of LOOM - as the presentation language of the maintenance Ontology and the mediator system. In the same mind of SIMS, we believe that using PowerLOOM as the language of the mediator system, this latter can benefit from expressivity and reasoning capacities of the reasoning engine provided by PowerLOOM.

In fact, PowerLoom’s focus is on expressivity of its representation language while still providing good scalability to large ontologies and knowledge bases. It supports various reasoning mechanisms as logical deduction which infers statements that logically follow from the asserted statements and rules, hypothetical reasoning, equality reasoning, arithmetic and reasoning with inequalities. In the other hand,

PowerLOOM has a static and dynamic query optimizer, that, similar to optimizers used in database systems, orders conjunctive goals based on relation extension sizes and rule fan-out to minimize intermediate result sets and chaining. Given this mechanism it is possible to run PowerLOOM queries that return 100,000's of solutions. PowerLOOM also has a powerful relational database interface that allows it to utilize the power of databases for handling large assertion bases [6].

These characteristics are useful to enforce the mediator functionalities. These latter are generally, queries management, information sources identification and especially find correspondences between concepts of local ontologies (called alignment) [8]. To guarantee this alignment, we use techniques of similarities measurements that are not used in the scope of main mediator systems. These techniques are generally used by ontology alignment methods. These methods have various goals, not specific to mediation tasks.

Finally, when summarizing, the added value of this work is the richness of the mediator system by combing a range of techniques aided to provide a more efficient and more accurate mediation services, and especially the closest to full-automatic.

### B. Definitions

We define a *context* as an explicit specification of a body of knowledge on values or features of the concept. A context is essentially composed of a list of keywords describing a concept in a specific situation.

Also, we take account to the different views which can exist in the domain. For example the *view* of a *Maintenance Operator* is different from the *storekeeper view*. These different views of the domain are clearly apparent in the domain Ontology versions developed for diverse applications integrated in the s-maintenance platform. To treat these possible semantic conflicts, we introduce the concept of the *Domain\_View*.

A *Domain\_View* is defined as a specific vision of the concept in the scope of an area of expertise.

Usually in the aim to find concepts correspondences, we use the extensional technique [8] by comparing concepts' instances; to obtain the optimal comparison we believe that not all attributes of the concept are important. Each concept has some important attributes which are essential to characterize the concept. So we introduce the notion of *Essential attributes*.

*Essential attributes* are defined as a set of attributes which characterize a concept. If these attributes are absent; we consider that the concept is poorly defined.

Subsequently, we consolidate these defined notions to be used as annotation called descriptor.

A Descriptor is a set of informative data about the concept, presented as a 4-uplet  $\langle \text{CN}, \text{EA}, \text{Ctx}, \text{DV} \rangle$  when CN presents the concept name, EA is the set of essential attributes, Ctx is

the context of the concept, and DV refers to the domain view.

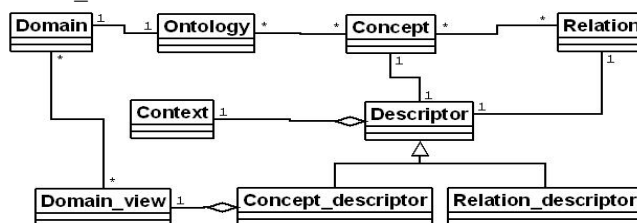


Fig. 6. This figure shows the relation between introduced notions and the domain Ontology.

It is important to note that in PowerLOOM attributes are defined as relations. As shown in Fig. 2 we introduce the notion of *relation\_descriptor* which present an annotation about *essential attributes*. These annotations serve to find correspondences between essential attributes of the compared concepts. This correspondence helps the mediator to measure the similarity between compared concepts.

A *relation\_descriptor* is defined as the 4-uplet  $\langle \text{EAN}, \text{T}, \text{Ctx}, \text{U} \rangle$  when EAN defines the essential attribute name, T presents the attribute type (i.e. string, list...), Ctx presents the attribute context and finally U presents attribute unity; this element is optional (i.e. euros, pouce...) (see Fig. 7).

```

Concept_Descriptor
<Competency, {AREA, EXPERIENCE, DIPLOMA, QUALIFICATION},
{knowledge, empowered, expertise aptitude}, Human-resources- management>

Relation-Descriptor
<Diploma, string, {university degree, study}, -->
<Area, string, {domain, field, speciality}, -->

```

Fig. 7. Example of descriptors. It shows an example of the concept\_descriptor of Consumable as well as relation\_description of the essential attributes of this latter.

Note that descriptors edition will be done when creating concepts. When creating the shared Ontology of maintenance, we stored in a repository various domain views which we suppose cover the entire maintenance domain. Then we stocked different contexts in a second repository. These two repositories are shared and accessible via the platform. When an integrated system develops its local ontology (e.g. a version of the common Ontology), each concept added must have a *Concept\_descriptor* and each essential attribute defined in the descriptor must have its *Relation\_descriptor*. So these descriptors will be defined by designers of the local ontology by consulting the two repositories. In the case when a designer doesn't find the adequate *Domain\_View* or *Context* he can add the missing *Context* or *view*.

### C. Mediator architecture

The mediator system includes the following seven internal components:

*Mediator TBOX*: is a terminology box contains the TBOXes of all local ontologies versions and the TBOX of the common Ontology. In general, the TBOX contains sentences



describing concept hierarchies (i.e., relations between concepts) and logical rules [27].

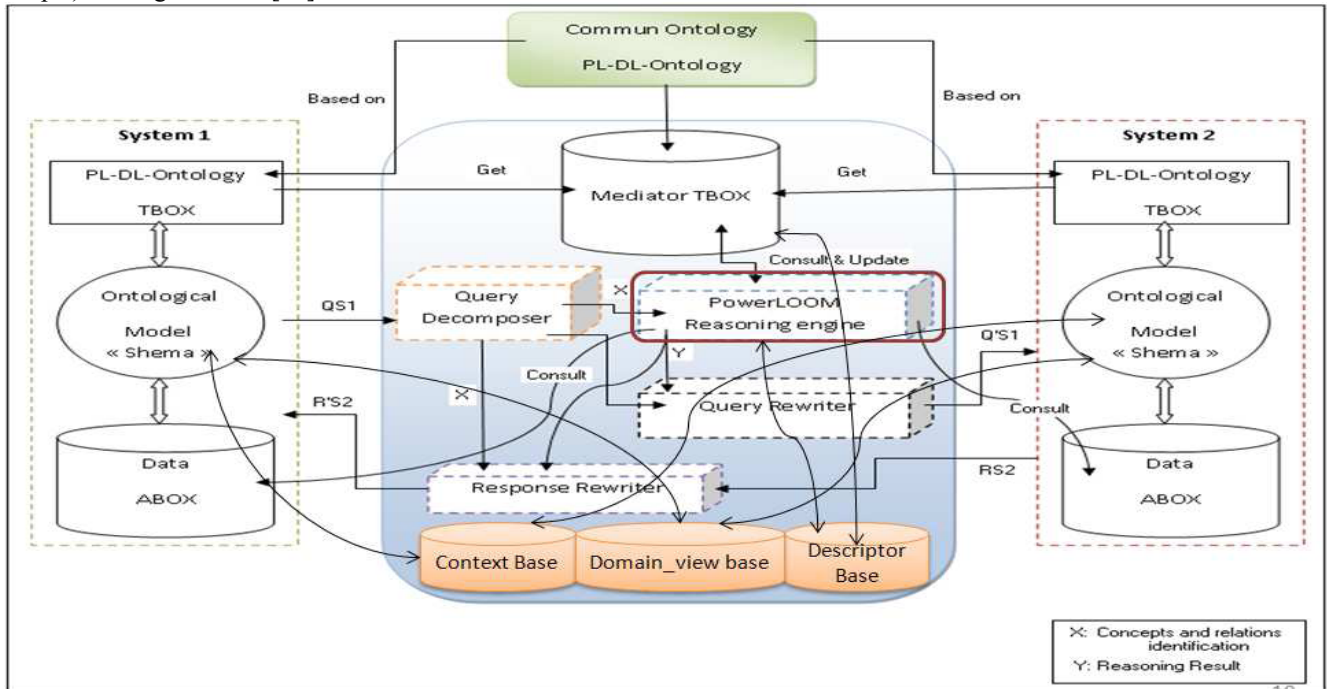


Fig. 8. Mediator architecture. The figure presents the different components of the mediator and arrows show interactions between them

*Query Decomposer*: this component gets the received query from a sender system and identifies concepts and relation used in the query.

*PowerLOOM Reasoning engine*: infers rules of the TBOX to define mapping between concepts and relations of local ontologies. This component identifies - by similarity measurements - the correspondences between concepts and relations of the local ontologies used in query received by the platform. It updates the mediator TBOX by adding rules resulting from the correspondence task.

*Context base, Domain View base and Descriptors base* are three repositories for contexts, domain views and descriptors of concepts and relations.

*Query Rewriter*: rewrites queries received by the platform to a form understandable by the local ontology of the target system.

*Response Rewriter*: rewrites the received responses to a form understandable by the local ontology of the sender system.

#### D. Mediator functioning

Various systems are connected to the maintenance platform. Each system has its proper local ontology which is a version of the common domain Ontology.

These ontologies (PL-DL-Ontology in Fig. 8) are presented by description logics in PowerLOOM. We dissociate the TBOX and the ABOX of the local ontology. The ABOX (assertion box) contains instances of concepts and relations of TBOX [27].

The “Mediator TBOX” gets dynamically the local TBOXes of the platform’s integrated systems. Each update made on

the local ontology the Mediator TBOX is automatically updated.

We also have to note that if the query contains only concepts from the shared Ontology the platform doesn’t need to pass by the mediator system to ensure communications.

When a System sends a query via the s-maintenance platform, in reality the platform sends the query received to another/other integrated system(s) via the “Mediator system”.

The “Query Decomposer” identifies concepts and relations expressed with the local ontology of the sender system in the received query.

Then it sends these concepts to “PowerLOOM Reasoning engine” (PLRE) which consults the “Mediator TBOX” to identify the mapping (correspondences) between concepts of ontologies of the sender and target systems. This component consults the TBOX to verify if it contains rules of subsumption or equivalence between concepts identified in the query and concepts of the local ontology of the target system. If the case, it sends this correspondence to the *Query Rewriter*. Else based on the common Ontology the PLRE identifies concepts that can be probably correspondent (i.e. with the reasoning capacities the engine doesn’t need to compare all concepts of the origin system with all concepts of the local ontology of the target system.)

The PLRE identifies two sets SLO1 and SLO2. SLO1 presents concepts used on the query which are identified by the Query decomposer, and SLO2 presents a set of concepts in the local ontology of the target system. Hence, the PLRE uses the semantic technique of ontology alignment [8]. When the semantic technique based on concepts model, firstly the PLRE identifies a set of concept which we called SCO (Set of concept from the Common Ontology), these concepts are

those included in the common Ontology and which have a direct or indirect relation with concepts in SLO1. Then, to identify concepts of SLO2, the PLRE identifies concepts in second local ontology which have direct or indirect relation with concepts of SCO. Then, it compares each concept in SLO1 with each concept in SLO2 by applying the algorithm of similarities measurement (SIMGO) which we developed. Finally, the *PLRE* adds the resulting correspondence as rules in the *Mediator TBOX*.

After that, the resulting correspondence is sent to the "Query Rewriter". This latter gets the origin query from the "Query Decomposer", and rewrites it with concepts of the target system.

The target system edits the response and sends it to the platform. The "Response Rewriter" adapts the response with concepts of the local ontology of the sender system.

### E. Illustration

We illustrate this functioning by the first motivating example presented above:

S1 sends a query via the platform asking for an actor having specific Expertise from S2.

The platform knows that Expertise is not defined in the shared Ontology. So it passes the query to the mediator system.

The query decomposer analyses the query and identifies that Actor and Expertise are concepts and EES is an instance of Expertise. It returns this result to the PLRE.

After consulting the mediator TBOX, the PLRE doesn't find a correspondence rule between Expertise with a concept in S2LO (S2 local ontology). It identifies that the concept Expertise of S1OL (S1 local ontology) is related to the concept Actor of the shared Ontology. So it identifies concepts related to Actor in S2OL and don't exist in the shared Ontology. It identifies the concept Competency. This latter is related to human-resource concept which subsumes the concept Actor (indirect relation). Then it applies SIMGO to identify correspondence between Expertise and Competency.

Similarity functions used by SIMGO use Concept-descriptors and relation-descriptors to compare these two concepts:

```
<Expertise, (Domain, Degree), {Knowledge,
empowred, aptitude, skills}, intervention>
<Competency, {Area, Experience, Diploma,
Qualification}, {Knowledge, empowred, aptitude,
expertise}, Human-resource_management>
```

If correspondence is identified, PLRE formulates an equivalency or a subsumption rule and adds it in the mediator TBOX.

The verified correspondence between Experience and Competency is submitted to the Query rewriter.

The Query\_Rewriter rewrites the initial query using the concept Competency and send it to S2.

S2 receives the query, and responds the platform; the received response is treated in the mediator system by the Response rewriter. This latter uses the found correspondence

between Experience and Competency to adapt the response and sends it to S1.

Regarding the second example, when Sys2 enhances its local ontology, new concepts are added in the "Mediator TBOX". Consequently, when Sys2 sends a query to Sys1 via the platform about an Alarm, the PLRE consults the "Mediator TBOX" to find a correspondence with the concept Alarm which doesn't known in Sys1. PLRE finds the rule which subsumes Alarm with Triggering\_event, so it sends this rule to the Query\_rewriter to conclude the mediator function.

### F. Similarity algorithm

This algorithm applies some function of similarities based on the *terminological, internal structure and extensional techniques* [29] by using introduced notions of Context, Domain\_view and Descriptor.

*Terminological technique:* we base our approach on the analysis of semantic annotation added as descriptors of concepts. We define these two functions  $\Phi_{\text{Domain-View}}$  and  $\Phi_{\text{Context}}$  to compare the two concepts  $C_1$  and  $C_2$  belonging respectively to SLO1 and SLO2.

$$\Phi_{\text{Domain-View}}(C_1, C_2) = \begin{cases} 1 & \text{if Descriptor.Domain\_View}(C_1) = \text{Descriptor.Domain\_View}(C_2) \\ 0 & \text{else} \end{cases}$$

$$\Phi_{\text{Context}}(C_1, C_2) = \begin{cases} 1 & \text{if Descriptor.Context}(C_1) = \text{Descriptor.Context}(C_2) \\ \text{Else } \Phi_R = \frac{\text{Descriptor.context}(c_1) \cap \text{Descriptor.context}(c_2)}{\text{Descriptor.context}(c_1) \cup \text{Descriptor.context}(c_2)} \end{cases}$$

We mentioned above that a context is presented as list of key words, in the case when the contexts are different; it is possible to find some key words appearing in the two contexts of  $C_1$  and  $C_2$ . In this case we calculate  $\Phi_R$  to calculate similarity of these two contexts.

*Internal structure technique:* in the scope of this compare technique we focus on the essential attributes and especially its Relation\_Descriptor. To this end, we introduce  $\Phi_{EA}$  to calculate similarities between essential attributes of the two concepts.

$$\Phi_{EA} = \text{Sim}(\text{Essential\_attributes}(C_1), \text{Essential\_attributes}(C_2)) =$$

$$\frac{\text{Sim}(\text{RelationDescriptor.context}(C_1.\text{EssentialATTRIBUTE}_M), \text{RelationDescriptor.context}(C_2.\text{EssentialATTRIBUTE}_N))}{\text{Max}(\text{card}(\text{essentialAttributes}(C_1)), \text{essentialAttributes}(C_2))}$$

If  
RelationDescriptor.context(C1.EssentialATTRIBUTE\_M) = RelationDescriptor.context(C2.EssentialATTRIBUTE\_N)

Then  
Sim (RelationDescriptor.context(C1.EssentialATTRIBUTE\_M), relationDescriptor.context(C2.EssentialATTRIBUTE\_N))= 1

else  
Sim (RelationDescriptor.context(C1.EssentialATTRIBUTE\_M), relationDescriptor.context(C2.EssentialATTRIBUTE\_N))= 0

*Extensional technique:* this technique focuses on concepts instances (stocked in the ABOX). In our case, we will focus on instances of the concept and values of its essential attributes. So we measure similarities between this latter by the function  $\Phi_I$ :

$$\Phi_I = (\Phi_{IC} + |\Phi_{VEAC} - \Phi_{IC}|)$$

When  $\Phi_{IC}$  presents similarity between instances of the two concepts to be compared:

$$\Phi_{IC} = \text{sim}(\text{instances\_C1}, \text{instances\_C2}) = \frac{\text{card}(\text{instance\_C1} \cap \text{instance\_C2})}{\text{card}(\text{instance\_C1} \cup \text{instance\_C2})}$$

$\Phi_{VEAC}$  presents the similarity between concepts' essential attributes values.

$$\Phi_{VEAC} = \frac{\sum \text{sim}(\text{values\_EssentialAttribute\_c1}, \text{value\_EssentialAttribute\_c2})}{\text{card}(\text{EssentialAttribute\_compared})}$$

$$\text{sim}(\text{values\_EssentialAttribute\_c1}, \text{value\_EssentialAttribute\_c2})$$

$$= \frac{\text{card}(\text{values\_EssentialAttribute\_c1} \cap \text{values\_EssentialAttribute\_c2})}{\text{card}(\text{values\_EssentialAttribute\_c1} \cup \text{values\_EssentialAttribute\_c2})}$$

These different functions are used in the scope of a similarity algorithm:

```
Algorithm: Concept_Similarity
Input: SLO1, SLO2
Variables: CtxSim, Sim_Vector
Begin
  For each (C1 ∈ SLO1) do
    For each (C2 ∈ SLO2) do
      If (ΦDomain_View(C1, C2) = 1) then
        If (ΦContext(C1, C2) = 1) then
          CtxSim ← 1
        Else CtxSim ← ΦR
        End if
      Else CtxSim ←  $\frac{\Phi_{EA} + |\Phi_{EA} - \Phi_I| + \Phi_R}{2}$ 
      End if
      If (sim(C1, C2) < CtxSim) then
        Sim(C1, C2) ← CtxSim
      End if
    End for
  End for
  If (sim(C1, C2) > 0.5) then
    Sim_Vector [].Add ([C1, C2])
  End if
End for
Return Sim_Vector[]
End Concept_Similarity
```

SIMGO has SLO1 and SLO2 as input; and as output, it has a vector of similar concepts. We defined a threshold of similarity, when  $\text{sim}(C1, C2)$  is greater than 0.5 we consider that concepts can be correspondents. But we consider only the greatest similarity value (i.e. if we find two similarity values greater than the threshold we consider the greatest value between them). The chosen threshold is defined to refine similarities results. It can be considered that this threshold is not high, but it should be noted that the similarity value is calculated by aggregating similarity techniques. Thus the similarity value is influenced by the aggregation of  $\Phi_{EA}$ ,  $\Phi_R$  and  $\Phi_I$ . To avoid giving weight to these similarity values according to their importance, we chose a threshold around the average for facilitating our task.

## V. DISCUSSION

Currently, mediator systems are not usable automatically on large scale systems. Its results lose in precision when the ontologies expand in size or/and heterogeneity. For that reason, these systems are currently used interactively or semi-automatically so that users enhance and manage the results' quality [8].

In this context, the help of similarities algorithms is as powerful as the ontologies grow in size and complexity. With

this mentality we built our mediator system. This initiative is totally contrary to the idea provided by Garcia et al when presenting TSIMMIS [12]. This idea outcome from the thesis that information integration could not, and should not, be fully automated. They opt in favor of providing a mediator system to assist users in their information processing and integration activities.

We agree that this assumption may be effective when the data sources are poorly structured and when compromise on a shared vocabulary cannot be achieved which is the case in most systems and fields. However, in our context, when we adopt an ontology hybrid approach we ensure that data sources are relatively well structured and consensus is reached especially by fencing the maintenance domain and identifying its deferent competence areas. Here human intervention is not appropriate or necessary to do the mediation task, but contrary to TSIMMIS spirit, humans provide necessary knowledge to assist it in its automatic mediation task. This mind explains introduced notions in the paper and the orientation to build a context oriented mediator as well as the role of the Ontology and descriptors.

In the other hand, adopting a hybrid approach can be seen as a limit for this mediator system, because most existing systems use their own ontology and does not share the same vocabulary with other systems. So the multiple ontology approach can be seen as the closest approach of reality. But building a mediator based on this approach is a great challenge that requires a lot of work on alignment methods and algorithms to more specific similarities [24]. Despite the hybrid approach eliminates to the platform the possibility to integrate a system that does not use the common Ontology, it can be seen the right approach to adopt if it is assumed that systems to be integrate respect the maintenance Ontology used by the platform.

Compared to SIMS [23] the mediator system does not only resolve semantic conflicts having the type "is-a" which is an easily achievable task through the initial reasoning engine provided by PowerLOOM. But thanks to the similarity algorithm, the presented mediator system is able to treat any type of relationship (i.e. is-a, association or composition).

SIMGO exploits much the aggregation of several similarity techniques. In the scope of these techniques, it uses the introduced notions of *context* and *Domain\_view* which allow the obtaining of a more accurate similarity measurements based on additional knowledge about each concepts in the ontology.

The mediator confronts a problem when it cannot find correspondences; it doesn't know what to do in this case. In a full automated mediation, like our case, this situation cannot be resolved without new learned knowledge. This end requires lots of testing to determine the percentage of non-accuracy results and percentage of not matching cases. Capturing these different cases allow us to put necessary knowledge to the mediator to overcome this problem. Another unresolved problem yet, is the composite correspondences, which is to align one concept in the first ontology with two or more concepts in the second ontology.



## VI. CONCLUSION

In order to evolve its provided services, new generation of maintenance platforms must ensure not only a data exchange between its integrated systems, but it must ensure a knowledge exchange. This latter needs a level of interoperability higher than the one ensured by existing platforms which is structural interoperability. Therefore, we provided in this work a semantic mediator system. This system is based on a common Ontology of maintenance domain that we have developed in previous work using PowerLOOM; an ontology description language based on description logics. The mediator ensures various functionalities as query analyzing, query rewriting and finding correspondences between concepts. These functionalities are provided thanks to some notions that we introduced in this work especially *domain\_view*, *descriptors* and the similarity algorithm *SIMGO*.

Our future work is to set up the mediator system into the maintenance platform which we develop with our industrial partners in the scope of SMAC project. In addition, we will evaluate SIMGO compared with other methods of ontologies alignment in the aim to evolve our measurements functions to obtain more exact and surer similarities.

## REFERENCES

- [1] W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani, H. Xue, "Systems integration and collaboration in architecture, engineering, construction, and facilities management", *A review, Advanced Engineering Informatics* (2009), doi:10.1016/j.aei.2009.09.001.
- [2] M. H. Karray, B. Morello-Chebel, N. Zerhouni, "towards a maintenance semantic architecture", *The Fourth World Congress on Engineering Asset Management (WCEAM 2009) Athens*. In press.
- [3] G. Wiederhold, "Mediators in the architecture of future information systems", *IEEE Comput.* 25, 3, pp. 38–49.
- [4] G. Wiederhold, M. R. Genesereth, The conceptual basis for mediation services. *IEEE Expert* 12, 5, pp. 38–47, 1997.
- [5] M. N. Huhns, M. P. SinghSINGH, "Managing heterogeneous transaction workflows with cooperating agents", In *Agent Technology: Foundations, Applications, and Markets*, N. R. Jennings and M. J. Wooldridge, Eds. Springer, Berlin, Germany, pp. 219–239.
- [6] H. Chalupsky, R. M. MacGregor, T. Russ, "PowerLOOM manual Powerful knowledge representation and reasoning with delivery" In *Common-Lisp, Java, and C++ Version: 1.40*, 25 May 2006.
- [7] C. Goh, S. Bressan, S. Madnick, M. Siegel, Context interchange: new features and formalisms for the intelligent integration of information. *ACM Trans Inf Sys*, 17(3), pp. 270–293, 1999.
- [8] J. Euzenat, A. Mocan, F. Scharffe, "Ontology Alignments. An Ontology Management Perspective". *Kap. 6*, pp. 177–206, 2008.
- [9] A. Gupta, B. Ludäscher, and M. E. Martone, "Knowledge-Based Integration of Neuroscience Data Sources". In *Intl. Conference on Scientific and Statistical Database Management (SSDBM)*, 2000.
- [10] B. Ludäscher, A. Gupta, and M. E. Martone, "Model-Based Information Integration in a Neuroscience Mediator System". In *Intl. Conf. on Very Large Data Bases (VLDB)*, pp. 639–642, Cairo, Egypt, 2000.
- [11] B. Ludäscher, A. Gupta, and M. E. Martone, "Model-Based Mediation with Domain Maps". In *17th Intl. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany, 2001.
- [12] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom, "The TSIMMIS Approach to Mediation: Data Models and Languages", In *Next Generation Information Technologies and Systems*, 1995.
- [13] A. Y. Levy, A. Rajaraman, and J. J. Ordille, "Querying Heterogeneous Information Sources Using Source Descriptions", In *Intl. Conference on Very Large Data Bases (VLDB)*, pp. 251–262, 1996.
- [14] L. M. Haas, D. Kossmann, E. L. Wimmers, and J. Yang, "Optimizing Queries Across Diverse Data Sources", In *Intl. Conf. on Very Large Databases (VLDB)*, pp. 276–285, Athens, Greece, 1997.
- [15] C. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Velikhov, V. Chu, "XML-Based Information Mediation with MIX". In *Intl. Conf. on Management of Data (SIGMOD)*, pp. 597–599, 1999.
- [16] K. U. Sattler, I. Geist, and E. Schallehn, "Concept-based querying in mediator systems", *The VLDB Journal*, 14(1), pp. 97–111, 2005.
- [17] B. Amann, C. Beeri, I. Fundulaki, M. Scholl, "Ontology based integration of XML Web resources", In *Proceedings of ISWC 2002*, Sardinia, Italy, 9–12 June 2002. Lecture notes in computer science, vol 2342. Springer, Berlin Heidelberg New York, pp. 117–131, 2002.
- [18] L. Palopoli, G. Terracina, and D. Ursino, "The System DIKE: Towards the Semi-Automatic Synthesis of Cooperative Information Systems and DataWarehouses", In *Proc. ADBIS-DASFAA Symposium*, pp. 108–117, 2000.
- [19] D. Beneventano and S. Bergamaschi, "Extensional Knowledge for semantic query optimization in a mediator based system", In *Int. Workshop on Foundations of Models for Info. Integ. (FMII-2001)*, 2001.
- [20] S. Bergamaschi, S. Castano, M. Vincini, "Semantic Integration of Semistructured and Structured Data Sources", *SIGMOD Record*, 28(1):54–59, 1999.
- [21] B. Ludäscher, A. Gupta, M. Martone, "Model-based mediation with domain maps". In *Proceedings of ICDE'01*, Heidelberg, Germany, 2–6 April 2001, pp 82–90.
- [22] D. Calvanese, S. Castano, F. Guerra, D. Lembo, M. Melchiori, G. Terracina, D. Ursino, M. Vincini, "Towards a Comprehensive Methodological Framework for Semantic Integration of Heterogeneous Data Sources". *Intl. Workshop on Knowledge Representation meets Databases (KRDB)*, 2001.
- [23] Y. Arens, C. Chee, C-N. Hsu, CA. Knoblock, (1993) "Retrieving and integrating data from multiple information sources". *Int J Intell Cooper Inf Sys*, 2(2):127–158.
- [24] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, "Ontology-Based Integration of Information: A Survey of Existing Approaches", In *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, Seattle, WA, pp.108-117, 2001.
- [25] E. Mena, V. Kashyap, A. Sheth, and A. "Illarramendi. Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies". In *Proceedings 1st IFCIS International Conference on Cooperative Information Systems (CoopIS '96)*. Brussels, 1996.
- [26] H. Stuckenschmidt, H. Wache, T. Vögele, U. Visser. "Enabling technologies for interoperability". In *Ubbo Visser and Hardy Pundt, editors, Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, pp. 35–46, Bonn, Germany, 2000. TZI, University of Bremen.
- [27] F. Baader, D. McGuinness, D. Nardi, P.F. Patel-Schneider, "The Description Logic Handbook: Theory, Implementation and Applications". Cambridge University Press, 2003.
- [28] M. Mrissa, C. Ghedira, D. Benslimane, Z. Maamar, F. Rosenberg, S. Dustdar, "A context-based mediation approach to compose semantic web services". *ACM Trans. Internet Techn.*, 8(1), 2007.
- [29] Rahm E., Bernstein P., "A survey of approaches to automatic schema matching", *VLDB Journal*, 10(4), pp. 334-350, 2001.