Model-driven Approach for the Interoperability of Enterprises' Services Information Exchange

Catarina Marques-Lucena[1], João Sarraipa[2], Carlos Agostinho[2] and Ricardo Jardim-Goncalves[2]
[1]*DEE, FCT, Universidade Nova de Lisboa, Caparica, Portugal*
[2]*Centre of Technology and Systems, CTS, UNINOVA, Caparica, Portugal*

Keywords: Interoperability, Semantic Interoperability, Mappings Representation, Modelling Morphisms, Model Transformations.

Abstract: Nowadays, enterprises have been pushed up the rate of industrial transformation to high level products and services. The capability to agilely respond to new market demands became a strategic pillar for enterprises survival. It leads to the necessity of create mechanisms to allow enterprises to combine forces to compete jointly in the market, in order to raise their own added value and to become specialist in niche activities. But to unite forces, enterprises must exchange information/knowledge between them. A semantic problem emerges when the same representation of a thing can have different meanings to different applications and enterprises. It can be a disabler for information exchange and its interpretation. The Semantic Web Service concept emerges as a solution to facilitate Web Services description and consequently discovery tasks, unifying the domain services. Moreover, the data integration between concepts can be achieved trough mappings establishment between Web Services elements. However, a solution that allows to map Web Service elements to semantic concepts represented using OWL is still missing. To face this issue, a model-driven approach supported by a mapping tool is here presented with the aim of facilitate the information exchange trough the establishment of mappings between enterprises services models and the domain reference lexicon.

1 INTRODUCTION

In the emerging society, competitive markets are becoming increasingly dynamic, and consequently complex, with companies not prospering and surviving through their own individual efforts (Friedman, 2006). Thus, global markets are willing to improve their competitiveness through collaboration and partnerships, motivating companies to look for enhanced interoperability between systems and applications in industry. Therefore, enterprises need to be able to dynamically adapt themselves to take advantage of market opportunities, establishing collaborative business practices to compete with big enterprises (Agostinho et al., 2006). Moreover, the ability of an enterprise to interoperate with others is not only a recognized quality and advantage to obtain competitiveness in today's market, but it also becomes a matter of survival for many companies.

Interoperability can be defined as the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE, 1990). It means that enterprises need their systems/products to work with the surrounding systems/products without great efforts to be interoperable. However, one of the main difficulties regarding the interoperability between systems and applications is related to the high number of semantic representations of the same segment of reality (e.g. systems and products) which are not semantically coincident (even inside the same domain) (Lucena et al., 2013), as a consequence of the heterogeneity of communities and enterprises. It results in a difficult semantic interoperability achievement. Semantic Interoperability, defined as the ability of systems/components to share and understand information at the level of formally defined and mutually accepted domain concepts (Sø lvberg, 1999), traditionally is achieved through peer-to-peer mappings where each participant tends to use its own data format and business rules, handling as many mappings as the number or partners to interoperate. Here, another interoperability issue emerges, one related to systems dynamics. Commonly, systems are time-variant, and even if we are able to find a 'good' model to describe it, when facing the dynamics of the environment, such

Marques-Lucena, C., Sarraipa, J., Agostinho, C. and Jardim-Gonçalves, R.
Model-driven Approach for the Interoperability of Enterprises' Services Information Exchange.
DOI: 10.5220/0005853207890799
In *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016)*, pages 789-799
ISBN: 978-989-758-168-7

model will become obsolete in time. Moreover, a model is just a representation of how an entity see the world. It can (and should) be constantly refined in order to adapt to new requirements. As a consequence, all this dynamics and heterogeneity leads in most cases, the network to experience interoperability problems because if just one of the network members adapts to a new requirement, the harmony is broken, and the network begins experiencing interoperability failure (Agostinho et al., 2011). This is even more evident in multi-domain networks (e.g. collaborative product design) where information is dispersed and frequently replicated in many Information Systems trough Web Services usage.

1.1 Semantic Web Services

Going towards to what was described before, one of the biggest issue in the Service Oriented Architecture (SOA) vision is related to the the data heterogeneity between inter-operating services. This is because, typically, enterprise systems are developed over several periods of time, by diverse organization and not necessarily with the same structures or vocabulary (Nagarajan et al., 2006). As a consequence, a substantial heterogeneity in Web Services elements' syntax, structure and semantics is verified.

Given the pivotal importance of service discovery for service-oriented computing, several attempts to improve the quality service discovery tasks are currently being pursued. One of the major efforts in this direction is promoted by the World Wide Web Consortium (W3C) which strongly advocates the introduction of semantic information in the description of Web services[1] (Brogi et al., 2005). Semantics acquires a particular importance to share services in a Peer to Peer (P2P) system where the lack of common understanding of the world generated the need of explicit guidance in the process of discovering available resources (Arabshian and Schulzrinne, 2007; Schmidt and Parashar, 2004; Haase et al., 2004).

Currently, services description is expressed by means of the Web Services Description Language (WSDL[2]) by declaring a set of message formats and their direction (incoming/outgoing) (Brogi et al., 2005). WSDL, in essence, allows the specification of the syntax of the input ans output messages of a basic service, as well as other details needed for the invocation of the service (Martin et al., 2005). However, it can be considered that WSDL do not offer sufficient semantic richness for services to be machine-processable (Kamaruddin et al., 2012). Human intervention is often needed to interpret the meanings in order to discover, compose, and invoke Web Services. Here, the notion of ontology is seen as an effective way to provide the required specifications to allow the automation of the mentioned tasks (Kamaruddin et al., 2012). Five projects to support the idea of ontologies incorporation in Web Services description are: OntoGov (Tambouris et al., 2004), TERREGOV (Vicente et al., 2006), SemanticGov (SemanticGov Consortium, 2007), Access-eGov (Sroga, 2008) and FUSION (FUSION Consortium, 2008).

1.2 Web Services Interoperability

There are several tools that allow mappings establishment across several data formats. One example is the *Advanced Mapping and Transformation* module of talend (talend, 2015). This module uses a Eclipse-based tooling environment to enable users to build, tests and maintain mappings across several data formats, including transformations between java and complex XML data, JSON, databases, flat files and EDI applications. It can be used to integrate applications, services, and APIs without coding. The *SAP NetWeaver Composition Environment Library*(SAP, 2013), allows to users to define how the data coming from one service operation as input in the service flow has to be mapped (definition of what will be the output) in the service flow. It allows to directly define how the data from one service data structure is mapped to another data structure. Other application tool to consider is IBM's *WebSphere Application Server V8.5.5* (IBM, 2013). It supports services mappings establishment by providing a simple way of performing content-based routing and message transformation that can be applied to web service messages being sent from service clients without needing to make changes to either the service client or service provider. All this tools are valuable solutions contributing to the deployment of interoperable Web Services across platform, applications, and programming languages. However, the authors didn't found any indication that this tools are capable to import OWL, used to define semantic relationships between concepts. This goes against the need of semantic descriptions usage to facilitate services discovery.

To face this issue, the authors proposed a solution able to represent connections or mappings between information systems, or more precisely, between web services elements and semantic concepts described by ontologies. The semantic representation of the domain is accomplished by a domain reference lexicon establishment in collaboration with the domain experts. Thus, the proposed solution to achieve Web

---

[1] http://www.w3.org/2003/10/swsig-charter

[2] http://www.w3.org/TR/wsdl

Services Interoperability is supported by the usage of a common reference lexicon (domain semantics) to be the intermediary in the communications between enterprises Web Services and to the outside. Then, the establishment of mappings between enterprises Web Services and the reference domain lexicon will allow each of the enterprises to keep its own knowledge and semantics unchanged, and still able to smoothly interact with its domain.

In this paper, an initial assessment about the need for semantic descriptions incorporation in Web Services and its interoperability is conducted. Based on this necessity, an approach for the interoperability of enterprises information exchange is proposed in section 2. Then, some key concepts to accomplish the proposed approach are detailed: Modeling Morphisms to Enable Sustainable Interoperability, Communication Mediator and Model Transformations. In section 6, the mechanism developed for the banking and insurance domain, capable of representing connections/mappings between elements of the domains ontology and Native Business Service (NBS), which are existing services deployed and exploited within the context of each service supplier, is presented. Finally, some conclusions and feature work statements are presented.

2 APPROACH FOR THE INTEROPERABILITY OF THE INFORMATION EXCHANGE

The author's approach aims to provide a solution (see Figure 1) able to represent connections or mappings between enterprises Native Business Services (NBS), or more precisely, between message elements of services and semantic concepts described by ontologies (reference domain lexicon). NBS are already existing software installed and exploited within the context of each supplier. The mappings with a reference lexicon can, then, be used to generate Supplier Business Services (SBS) accordingly to the elements of the domain ontology. SBS's are a set of features that are exposed from each supplier infrastructure and consist in the externalization of one or more NBSs. The generation of these SBSs promotes interoperability of enterprises applications in the domain allowing integration across multiple stakeholders (suppliers) and domains of interest (customers).

The first step to materialize this software-based system is to acquire a **platform reference lexicon** together with the domain experts. These are the main actors of their domain, thus their participation in the domain glossary (or reference lexicon) would increase the awareness of the domain terms. Glossaries are lists of specialized terms, mostly in alphabetic order, that sometimes are unique to a specific subject. Each term is composed by its corresponding description. It includes descriptive comments and explanatory notes, such as definitions, synonyms, references, etc (Sarraipa et al., 2014). Given the relevance of glossaries to better understand domains lexicon, several works were already presented using different methods and techniques. While some authors opt for rule-based systems usage to do the text-mining and glossary building (Muresan and Klavans, 2002), (Westerhout and Monachesi, 2007), others presented solutions based on machine learning (Fahmi and Bouma, 2006), and statistical techniques to simulate human consensus (Velardi et al., 2006).

Then, a **mapping tool** that allows to represent both the domain glossary and the supplier services elements is required. This tool is intended to be used for administrative purpose (supplier's manager) and supports the process of mappings establishment and storage between the platform and suppliers data models. This mappings will be stored in a specific ontology - the Communication Mediators (detailed in section 4).

Finally, the **generator** is responsible for the generation of SBSs based on the mappings (represented in the communication mediator) created between the NBS and the platform ontology.

3 MODELING MORPHISMS TO ENABLE SUSTAINABLE INTEROPERABILITY

Models are used to capture the essential features of real systems by breaking them down into more manageable parts that are easy to understand and to manipulate. They are used in systems development activities to draw blueprints of the systems and to facilitate communication between different people in the team at different levels of abstraction. People have different views of the system and models can help them to understand these views in a unified manner (Abdullah et al., 2002).

In mathematics, *Morphism* (MoMo) is an abstraction of a structure-preserving map between two mathematical structures. It can be seen as a function in set theory, or the connection between domain and co-domain in category theory (INTEROP NoE, a). Recently, this concept has been gaining momentum applied to computer science, namely to systems interoperability. This new usage of *morphisms* specifies

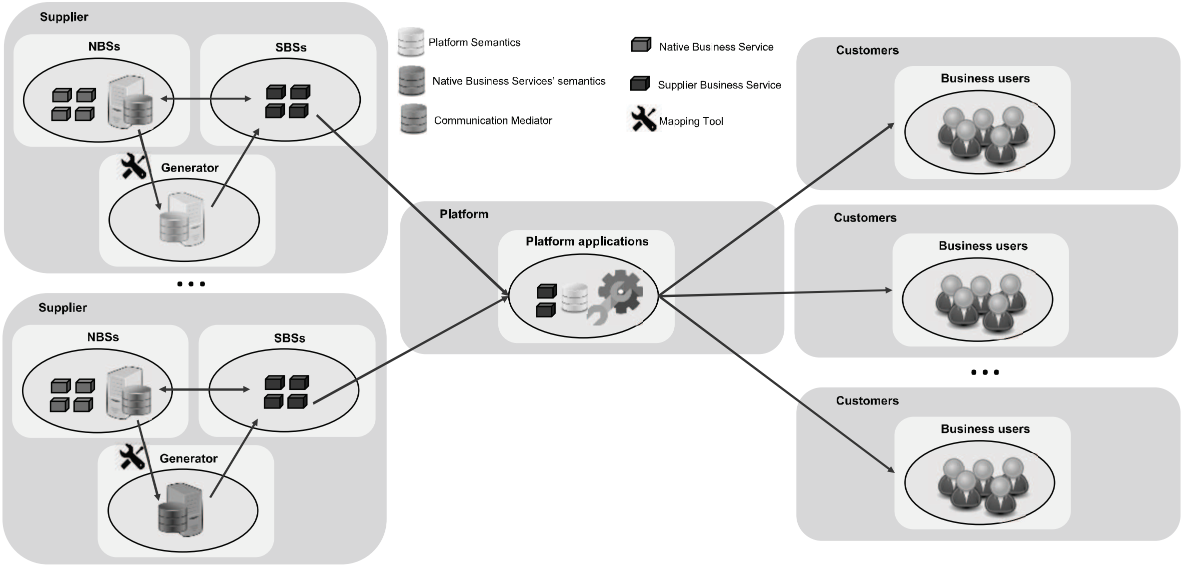


Figure 1: Approach for the interoperability of the information exchange.

the relations (e.g. mapping, merging, transformation, etc) between two or more information model specifications (*M* as the set of models).

In this context, the research community identifies two core classes of MoMo: non-altering and model altering morphisms (INTEROP NoE, a; Agostinho et al., 2007). In the non-altering morphisms, given two models (source *A* and target *B*), a mapping is created relating each element of the source with a correspondent element in the target, leaving both models intact. In model altering morphisms, the source model is transformed using a function that applies a mapping to the source model and outputs the target model (Delgado et al., 2006). Other relations, such as the merge operation, can also be classified as model altering morphisms, however they are not detailed in this work.

3.1 Knowledge Enriched Tuple for Mapping Representations

The research community has developed many proposals to morphisms representations (INTEROP NoE, b). As analyzed in (Agostinho et al., 2011), graph theory has been used in some, although other theories can be considered to achieve the envisaged goals, e.g., set theory (Dauben, 1990), model management (Bernstein, 2003), or semantic matching (Sarraipa et al., 2010). However there is not a single perfect solution that can be used to achieve all the morphisms goals at once. Some are ideal for structural issues, others for semantics providing good human traceability, and others are more formal and mathematical based. For that reason, in this work, is used a 5-tuple mapping expression (Mapping Tuple - MapT), presented in (Agostinho et al., 2011), with the goal of consolidate existent approaches to morphisms:

$$< ID, MElements, KMType, MatchClass, Exp > \qquad (1)$$

being:

- ID - unique identifier of the $MapT$;
- $MElements$ - pair indicating the mapped elements;
- $KMType$ - stands for Knowledge Mapping Type and determines the type of mappings represented in a specific instance of $MapT$;
- $MatchClass$ - stands for Match/Mismatch classification and depends on the $KMType$:
  - if $a = b$, the mapping is absolute and $MatchClass$ = Equal;
  - if $KMType$ = conceptual, the mapping is relating terms/concepts: $\{Equal, Naming, Coverage, MoreGeneral, LessGeneral, Disjoint\}$
  - Otherwise the mapping is either non-existing or more concrete addressing structural issues
- EXP - stands for mapping expression that relates and further specifies the previous tuple components. Normally, this expression can be translated to an executable transformation language (e.g. ATL).

4 COMMUNICATION MEDIATOR

The Communication Mediator Ontology has been built up as an extension of the Model Traceability Ontology defined by (Sarraipa et al., 2007), and it is able to represent ontology semantic operations, like: (1) Semantic mismatches; (2) Semantic transformations; (3) Ontologies mappings; and other ontologies operations. Thus, the MO is able to log ontology and entity operations in a way that is possible to trace changes in all the ontology life cycle. It addresses traceability as the ability to chronologically interrelate the uniquely identifiable objects in a way that can be processed by a human or a system. The mapping relations can be related to a traceability element, in such sense that a specific term defined in the reference ontology has a related one in the organisation member ontology, making possible a way to trace ontology elements. This way, the morphisms are modelled with traceability properties in a sense that they enable to store different versions of information model elements, as well as mappings between specific objects defined in a model or ontology (Sarraipa et al., 2010).

The structure of the MO is described as follows: the MO has two main classes: 'Object' and 'Morphism'. The 'Object' represents any 'InformationModel' (IM), which is the model itself, and 'ModelElements' ($MElements$) (also belonging to the IM) that can either be classes, properties or instances. The 'Morphism' associates a pair of 'Objects' (related and relating), and classifies their relationship with a 'MorphismType' (*MType*), 'KnowledgeMappingType' (*KMType*) (if the morphism is a mapping), and 'Match/Mismatch' (*MatchClass*) class (Sarraipa et al., 2010).

5 MODEL TRANSFORMATIONS

The models mapping specifications can be performed either at a high level of formalization using graphs, sets, tuples, etc. or at lower levels, i.e. specifying the mappings by text. However, in both cases is necessary to implement them using a transformation language.

Atlas Transformation Language (ATL) is one of the most used transformation languages, having a large user base and being very well documented(Bzivin et al., 2003). An ATL transformation is composed by a set of rules (matched rules) that define how the source model elements are linked to the target model elements. These elements can then be filled with information from the source model by called rules (similar to functions in usual object languages like JAVA) and action blocks (blocks of imperative code which can be used by matched rules and called rules).

Model transformations are a current practice to enable interoperability among two organizations or ecosystems of organizations, each with their own service system. With them, one can specify P2P (Peer to Peer) mappings to translate any data provided from one sides format specification into the other, thus allowing a seamless exchange of information.

When performing a horizontal model transformation (e.g. converting instances of a model to instances of another model) an explicit or an implicit mapping of the 'meta-model' has to be performed. Thus, as depicted in 2 the idea that when performing a transformation morphism at a certain level *n*, this transformation has (implicitly or explicitly) to be designed by taking into account mappings at level $n+1$. Once the $n+1$ level mapping is complete, executable languages can be used to implement the transformation at *n* level, e.g. using ATL and the Query/View/Transformation (QVT) (OMG, 2008).

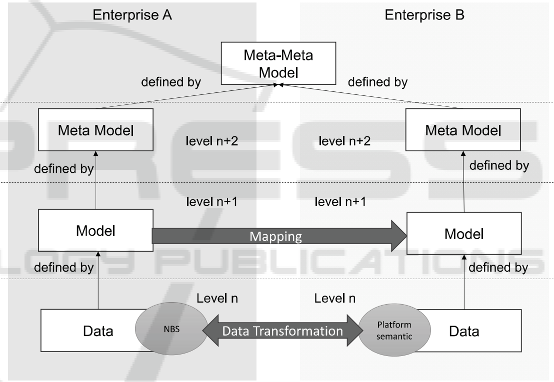


Figure 2: Method for Horizontal Transformation.

This type of transformations are normally static processes that once defined can be repeated any number of times achieving the same results. However due to the constant knowledge change caused by the dynamics of the global market, services and models that regulate enterprise systems, are not. In fact, some researchers have attempted to extrapolate results from a 'general systems theory' or 'complexity theory' that could explain the importance of evolution of systems in all fields of science (Gharajedaghi, 2011). These theories view all systems as dynamic, 'living' entities that are goal-oriented and evolve over time, thus, information systems, services and the mappings that connect them should be prepared to respond to the environment dynamics which is in a constant update. To support this dynamicity, horizontal transformations should be provided with traceability features, and mappings stored in a parseable and structured knowledge-base (Agostinho et al., 2011).

Since it is used a knowledge enriched tuple for mappings representation that are stored on a CM is possible to keep traceability of model mapping changes so that readjustments can be easier to manage, and data exchange re-established automatically using the model-driven development paradigm.

6 ISOFIN CLOUD APPLICATION SCENARIO

The ISOFIN cloud project envisage the establishment of technical solutions, guidelines and standards to support the interoperability resolution, facilitating a seamless sharing of artifacts at the knowledge and software level that would enable a fast and efficient creation of new Financial (e.g. bench and Insurance) products or services. Thus, ISOFIN's overall aims are:

- Analyze the problems resulting from the internalization of financial applications and identify potential solutions
- Create ways of reducing the time that software developers take to achieve interoperability between domain financial stakeholders (e.g. bench and insurance)
- Standardize the domain semantic level, where no harmonization exists
- To implement mechanisms to facilitate the generation of new services in a ubiquitous context that could easily be discovered and accessed by domain customers

Based on the project aims, the expected result is the development of a platform that speeds up the development and deployment of services to be sold to the clients and users of the financial domain. For that, the steps identified in the approach presented in section 2 for the interoperability of the information exchange section must be followed: (1) acquisition of a platform reference lexicon together with the domain experts; (2) definition of mappings between enterprise NBS and the reference lexicon using a mapping tool; and (3) Generation of SBSs based on the established mappings.

6.1 Domain Reference Lexicon Establishment

It is a fact that when an information system intends to represent a domain's knowledge it needs to be aligned to the community that it represents. Consequently it is required to have a solution where community members could present their knowledge about the domain and discuss it with their peers. Additionally, such knowledge must be available and dynamically maintained by all the involved actors (Marques-Lucena et al., 2015).

The solution adopted by the authors for the domain reference lexicon establhishement is based on MENTOR - methodology for enterprise reference ontology development (Sarraipa et al., 2010), supported by an interface accordingly with the works presented in (Lucena et al., 2014; Marques-Lucena et al., 2015) to implement the following MENTOR's steps (1) terminology gathering; (2) glossary building; (3) thesaurus building. The result is a domain lexicon, whose semantics are constantly refined trough a specific front-end in order to handle the knowledge provided by the domain experts.

6.2 Mapping Tool

The Mapping tool was previously developed to provide a graphical means to define different kinds of mapping between models (Agostinho et al., 2012). This tool was developed with the intention to open LIMM (Language Independent Meta-Model) files, an abstract interface on top of information systems.

JGraph has been elected and modified to read the input information model files and store morphisms at the MO ontology. It is a widely used open source project for graph visualization and manipulation, similar to Microsoft Visio, with good documentation and several examples. Features include a complete selection of layouts to automatically position the graph, many styles of shapes and edges, validation of connections, as well as an undo and redo manager.

Some adjustments were made to enable the interaction (mapping definition) between two different information models' graphs, and to become integrated to the Communication Mediator Ontology. To the integration with the CM ontology, JENA was used - a Java API for OWL providing services for model representation, parsing, database persistence and querying ontologies.

In the ISOFIN's application scenario. the mapping tool supports the mappings definition between NBSs elements (e.g. services and message concepts names) described in WSDL 1.0 and XSD, and the ISOFIN's reference lexicon described in OWL 2.0. For the purpose, the enabled features are the ones represented in Figure 3 use case: 1) Open wsdl file; 2) save mappings; 3) import mappings; 4) generate wsdl file. The import of the platform reference lexicon is made when the open wsdl file option is triggered.

That means, when the user decided to open a wsdl file for mappings establishment, both the wsdl and reference lexicon graphical representations are presented side by side (see Figure 9).

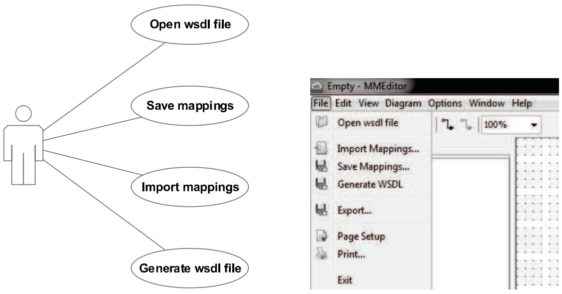


Figure 3: Mapping Tool Use Case.

In addition to the menu bar, three other areas compose the tool: 1) Mapping text description; 2) Minimized view of the full mapping screen; and 3) Mapping screen. The first area is where are described the selected mappings and elements, the second allows to navigate in the full map and finally, the third is where is done the representation of the models and the mappings (see Figure 4).

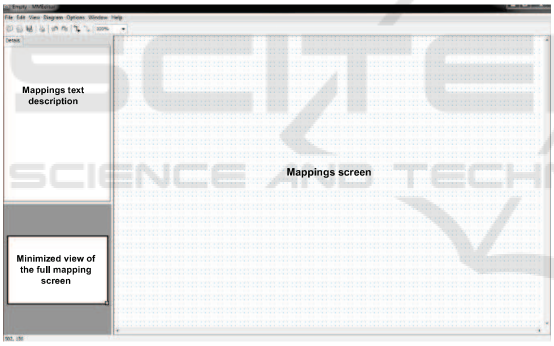


Figure 4: Mapping Tool working area.

Starting from the work presented in (Agostinho et al., 2012), some changes were made to the mapping tool to allow the interpretation of WSDL 1.0 (suppliers web services standard language) and OWL 2.0 (platform semantics description language). The most relevant changes are related to the transformation of both the input files into LIMM, so both the reference lexicon and NBS wsdl can be opened for mappings establishment

6.2.1 Language Independent Meta Model

This language enables the abstraction in relation to technologies and implementation details associated with the different modeling languages, and thus, enlarge the scope of users involved in a traditional mapping definition activity. Thus, having domain experts (suppliers) involved in the mapping process increases the quality of the mappings enabling the interoperability between suppliers and the rest of the platform participants.

6.2.2 OWL to LIMM Transformation

The transformation of OWL 2.0 to LIMM was already existing in the context of the European Project CRESCENDO (CRESCENDO IP, 2009) and explained in (Agostinho et al., 2012). To enable the mapping among the OWL meta-model (W3C, 2008) and the LIMM, one needs to firstly put the OWL data in XMI serialization following the OWL meta-model specifications. Nevertheless the procedure to do so is not straightforward as desirable since, in spite of the inputting OWL model is already XML serialized, it cannot be directly processed by the ATL toolkit which needs XMI as an input. The complete process for accomplishing the language mapping is illustrated in Figure 5, where the first step consists in doing an injection of the original model to an XML MOF meta-model specification. Following that, the second preparatory step consists in mapping that XML format to the reference OWL meta-model which will be the starting point for the actual $\theta(OWL, LIMM)$ language mapping (step 3). The, the step 4 takes place to execute the transformation between models itself using ATL.

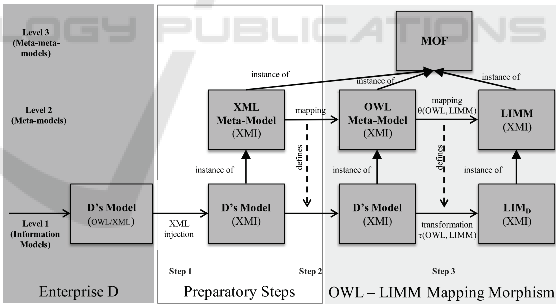


Figure 5: OWL 2.0 to LIMM Transformation (Agostinho et al., 2012).

6.2.3 WSDL to LIMM Transformation

The transformation of WSDL 1.0 to LIMM follows the same approach of the transformation of OWL 2.0 to LIMM described before. Since not all the WSDL model elements are relevant for the described use case, only the elements depicted using yellow (doted line) on Figure 6 where transformed. These are the elements containing the description of the services name, input and output messages, and the categorization of the messages. To allow the representation of services input and output messages, the container types were also considered (continuous line in red). The WSDL 'types' element is a container for XML Schema type definitions. The type definitions are referenced from higher-level message definitions in order to define the structural details of the message. To get more structured representation of the NBS content, the representation of the services was split in two parts, namely the types and the services (Figure 7).

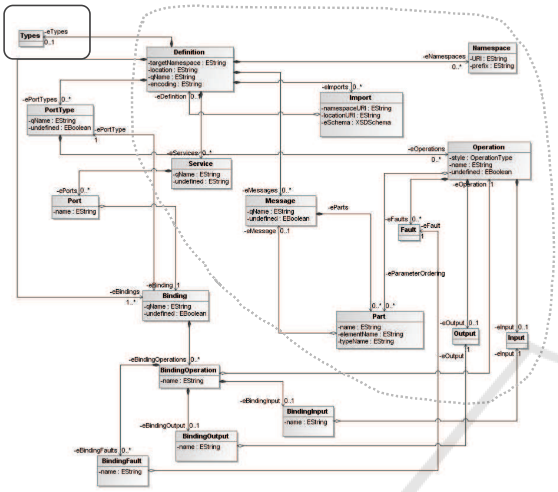


Figure 6: WSDL 1.0 Meta-model.

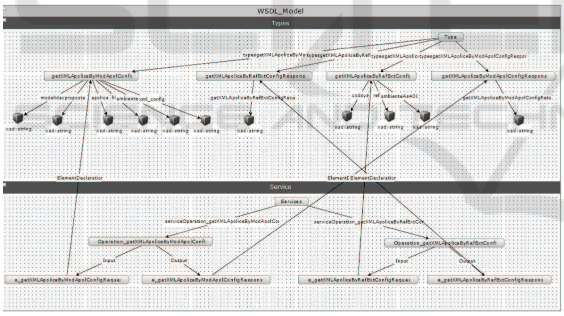


Figure 7: WSDL 1.0 LIMM Graphical representation.

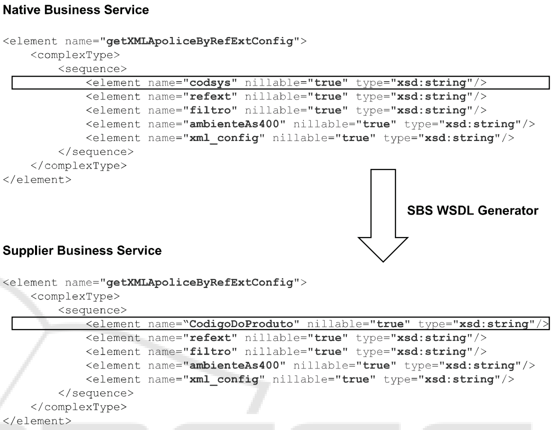
6.3 Mappings Storage

As mentioned before, the mapping tool provides a graphical means to define different kind of mappings while storing them in the Communication Mediator Ontology. Using the mapping tool developed, it is possible to open and show mappings between information models, namely, the NBS services and the ISOFIN reference lexicon, represented respectively in WSDL and OWL (Figure 9).

Since a mapping definition is a complex and time-consuming task, the Mapping tool is capable of halting the process at any time without losing the progress made so far by the user. This functionality still allows keeping mappings traceability.

6.4 Generate SBS WSDL File

To mediate information between the ISOFIN suppliers (NBS) and the ISOFIN platform it is needed to align the semantics of such information. To accomplish that, a generator to enable the representation of NBS services according to the ISOFIN nomenclatures was developed. This generator consists in a component of the mapping tool and its function is to generate a WSDL of the supplier services accordingly to the ISOFIN semantics from the established mappings.



Native Business Service

```
<element name="getXMLApoliceByRefExtConfig">
    <complexType>
        <sequence>
            <element name="codsys" nillable="true" type="xsd:string"/>
            <element name="refext" nillable="true" type="xsd:string"/>
            <element name="filtro" nillable="true" type="xsd:string"/>
            <element name="ambienteAs400" nillable="true" type="xsd:string"/>
            <element name="xml_config" nillable="true" type="xsd:string"/>
        </sequence>
    </complexType>
</element>
```

Supplier Business Service

```
<element name="getXMLApoliceByRefExtConfig">
    <complexType>
        <sequence>
            <element name="CodigoDoProduto" nillable="true" type="xsd:string"/>
            <element name="refext" nillable="true" type="xsd:string"/>
            <element name="filtro" nillable="true" type="xsd:string"/>
            <element name="ambienteAs400" nillable="true" type="xsd:string"/>
            <element name="xml_config" nillable="true" type="xsd:string"/>
        </sequence>
    </complexType>
</element>
```

Figure 8: WSDL output of the established mapping.

In Figure 8, the Supplier Business Service resulting of the mapping established in Figure 9 is presented. In the example provided, the element *codsys* of the complex element *getXMLApoliceByModApolConfig* is replaced by the ISOFIN concept *CodigoDoProduto*. In this way, each enterprise is capable to keep its internal nomenclature (e.g. codsys) and still be able to interact with its domain using a common reference lexicon as the intermediary in the communications established between the enterprise and the outside.

7 CONCLUSIONS

The aim of this work is to provide interoperability integration on the information exchanged between domain suppliers and a specific platform. To achieve that goal, an approach capable to represent connections or mappings between information systems is proposed by the authors. The outcome is that systems information subsystems are able to interact, but keeping their internal nomenclature thanks to the generation of Supplier Business Services.

Thus, in the application scenario, a mapping tool was used to support the establishment and definition

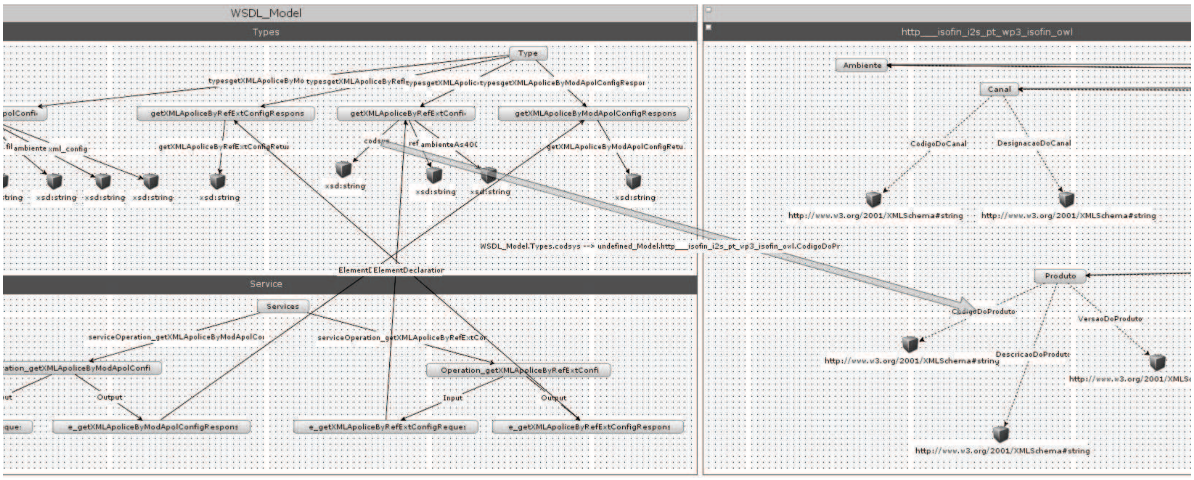


Figure 9: Mapping Tool demonstration.

of mappings between the ISOFIN ontology and supplier data models. These mappings are logged accordingly to a determined specification able to facilitate further traceability mechanisms implementation to facilitate maintenance of the established interoperability between the information systems. Such specifications are represented by a tuple that formalizes the mappings, which then can be stored as knowledge elements in the Communication Mediator Ontology.

The stored knowledge elements enable the generation of SBS (wsdl) web service already compliant with ISOFIN's reference lexicon. Such new web services will facilitate the process of sending and receiving interoperable data between suppliers and the ISOFIN system.

However, due to the infinite possible model formats (meta-models) to represent any kind of information in web services contents composition (e.g. NBS), it is difficult to create a full interoperability solution. Thus, despite the fact of the complexity of reaching a complete automatic or dynamic interoperability it is concluded that partial solutions could be reached when it is acknowledged in advance the nature of the meta-models of the information exchanged and when there is a specification for traceability representation able to represent all the kinds of mismatches.

These traceability representations support the proposed generator in handling specific interoperability situations. In matter of fact this is the specific ISOFIN goal to which this work is focused on. It provides traceability representation able to support some specific interoperability establishment solutions implementation.

ACKNOWLEDGEMENTS

The research leading to these results have received funding from the European Union 7th Framework Programme (FP7/2007-2013) under grant agreement OSMOSE[3] nr 610905, EC HORIZON2020 Program under grant agreement nr C2NET 636909[4], and also QREN under agreement 2010/013837 ISOFIN[5].

REFERENCES

Abdullah, M. S., Benest, I., Evans, A., and Kimble, C. (2002). Knowledge modelling techniques for developing knowledge management systems. In *Third European Conference on Knowledge Management: Trinity College Dublin, Ireland*, page 17.

Agostinho, C., Costa, R., Malo, P., and Jardim-Goncalves, R. (2006). Product Data integration in the demand of interoperability in e-Business. In *Intelligent Systems, 2006 3rd International IEEE Conference on*, pages 807–812.

Agostinho, C., ern, J., and Jardim-Goncalves, R. (2012). Mda-based interoperability establishment using language independent information models. In van Sinderen, M., Johnson, P., Xu, X., and Doumeingts, G., editors, *Enterprise Interoperability*, volume 122 of *Lecture Notes in Business Information Processing*, pages 146–160. Springer Berlin Heidelberg.

Agostinho, C., Sarraipa, J., DAntonio, F., and Jardim-Gonalves, R. (2007). Enhancing step-based interoperabity using model morphisms. In Gonalves, R., Mller, J., Mertins, K., and Zelm, M., editors, *Enterprise Interoperability II*, pages 817–828. Springer London.

---

[3] http://www.osmose-project.eu

[4] http://www.c2net-project.eu/

[5] http://isofincloud.i2s.pt/

Agostinho, C., Sarraipa, J., Goncalves, D., and Jardim-Goncalves, R. (2011). Tuple-based semantic and structural mapping for a sustainable interoperability. In Camarinha-Matos, L., editor, *Technological Innovation for Sustainability*, volume 349 of *IFIP Advances in Information and Communication Technology*, pages 45–56. Springer Berlin Heidelberg.

Arabshian, K. and Schulzrinne, H. (2007). An ontology-based hierarchical peer-to-peer global service discovery system. *Journal of Ubiquitous Computing and Intelligence*, 1(2):133–144.

Bernstein, P. A. (2003). Applying Model Management to Classical Meta Data Problems. In *Proceedings of the 1st Biennial Conference on Innovative Data Systems Research (CIDR)*.

Brogi, A., Corfini, S., and Popescu, R. (2005). Composition-oriented service discovery. In Gschwind, T., Amann, U., and Nierstrasz, O., editors, *Software Composition*, volume 3628 of *Lecture Notes in Computer Science*, pages 15–30. Springer Berlin Heidelberg.

Bzivin, J., Dup, G., Jouault, F., Pitette, G., and Rougui, J. E. (2003). First experiments with the atl model transformation language: Transforming xslt into xquery. In *2nd OOPSLA Workshop on Generative Techniques in the context of Model Driven Architecture*.

CRESCENDO IP (2009). Collaborative and robust engineering using simulation capability enabling next design optimisation (fp7-234344).

Dauben, J. (1990). *Georg Cantor: His Mathematics and Philosophy of the Infinite*. History of science. Princeton University Press.

Delgado, M., Agostinho, C., Malo, P., and Jardim-Goncalves, R. (2006). A framework for step-based harmonization of conceptual models. In *Intelligent Systems, 2006 3rd International IEEE Conference on*, pages 781–785.

Fahmi, I. and Bouma, G. (2006). Learning to Identify Definitions using Syntactic Features. *Learning to identify definitions using syntactic features*.

Friedman, T. L. (2006). *The World is flat : a brief history of the twenty-first century*. Farrar, Straus and Giroux, New York.

FUSION Consortium (2008). Fusion project website.

Gharajedaghi, J. (2011). *Systems Thinking: Managing Chaos and Complexity: A Platform for Designing Business Architecture*. Elsevier Science.

Haase, P., Agarwal, S., and Sure, Y. (2004). Service-oriented semantic peer-to-peer systems. In Bussler, C., Hong, S.-k., Jun, W., Kaschek, R., Kinshuk, Krishnaswamy, S., Loke, S., Oberle, D., Richards, D., Sharma, A., Sure, Y., and Thalheim, B., editors, *Web Information Systems WISE 2004 Workshops*, volume 3307 of *Lecture Notes in Computer Science*, pages 46–57. Springer Berlin Heidelberg.

IBM (2013). Websphere application server v8.5.5.

IEEE (1990). IEEE Standard Glossary of Software Engineering Terminology. Technical report.

INTEROP NoE. *Deliverable MoMo. 2-TG MoMo Roadmap*. InterOp.

INTEROP NoE. *Problem Analysis and Roadmap - I-VLab Platform*. InterOp.

Kamaruddin, L. A., Shen, J., and Beydoun, G. (2012). Evaluating usage of wsmo and owl-s in semantic web services. In *Proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling - Volume 130*, APCCM ’12, pages 53–58, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Lucena, C., Sarraipa, J. a., and Jardim-Goncalves, R. (2014). A knowledge management framework to support online communities creation. In Camarinha-Matos, L., Barrento, N., and Mendona, R., editors, *Technological Innovation for Collective Awareness Systems*, volume 423 of *IFIP Advances in Information and Communication Technology*, pages 29–36. Springer Berlin Heidelberg.

Lucena, C., Sarraipa, J. a., and Jardim-Gonçalves, R. (2013). Semantic Adaptation of Knowledge Representation Systems. In Camarinha-Matos, L., Tomic, S., and Graça, P., editors, *Technological Innovation for the Internet of Things*, volume 394 of *IFIP Advances in Information and Communication Technology*, pages 88–95.

Marques-Lucena, C., Agostinho, C., Koussouris, S., and Sarraipa, J. (2015). Knowledge management framework using wiki-based front-end modules. In *Proceedings of the 17th International Conference on Enterprise Information Systems*, pages 79–86.

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., and Sycara, K. (2005). Bringing semantics to web services: The owl-s approach. In Cardoso, J. and Sheth, A., editors, *Semantic Web Services and Web Process Composition*, volume 3387 of *Lecture Notes in Computer Science*, pages 26–42. Springer Berlin Heidelberg.

Muresan, S. and Klavans, J. (2002). A Method for Automatically Building and Evaluating Dictionary Resources. *Proceedings of the 3rd LREC*, pages 231–234.

Nagarajan, M., Verma, K., Sheth, A., Miller, J., and Lathem, J. (2006). Semantic interoperability of web services - challenges and experiences. In *Web Services, 2006. ICWS ’06. International Conference on*, pages 373–382.

OMG (2008). Meta object facility (omg) 2.0 query/view/transformation specification.

SAP (2013). Sap netweaver composition environment library.

Sarraipa, J., Jardim-Goncalves, R., and Steiger-Garcao, A. (2010). Mentor: an enabler for interoperable intelligent systems. *International Journal of General Systems*, 39(5):557–573.

Sarraipa, J., Marques-Lucena, C., Baldiris, S., Fabregat, R., and Aciar, S. (2014). The alter-nativa knowledge management approach. *Journal of Intelligent Manufacturing*, pages 1–17.

Sarraipa, J., Zouggar, N., Chen, D., and Jardim-Goncalves, R. (2007). Annotation for enterprise information management traceability. In *ASME 2007 International Design Engineering Technical Conferences and Comput-*

ers and Information in Engineering Conference, pages 893–899. American Society of Mechanical Engineers.

Schmidt, C. and Parashar, M. (2004). A peer-to-peer approach to web service discovery. *World Wide Web*, 7(2):211–229.

SemanticGov Consortium (2007). Semanticgov project website.

Sø lvberg, A. (1999). Data and What They Refer to. In Goos, G., Hartmanis, J., van Leeuwen, J., Chen, P., Akoka, J., Kangassalu, H., and Thalheim, B., editors, *Conceptual Modeling*, volume 1565 of *Lecture Notes in Computer Science*, pages 211–226. Springer Berlin Heidelberg.

Sroga, M. (2008). Access-egov-personal assistant of public services. In *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*, pages 421–427.

talend (2015). Advanced mapping and transformation.

Tambouris, E., Gorilas, S., Kavadias, G., Apostolou, D., Abecker, A., Stojanovic, L., and Mentzas, G. (2004). Ontology-enabled e-gov service configuration: An overview of the ontogov project. In Wimmer, M., editor, *Knowledge Management in Electronic Government*, volume 3035 of *Lecture Notes in Computer Science*, pages 122–127. Springer Berlin Heidelberg.

Velardi, P., Poler, R., and Tomás, J. (2006). Methodology for the Definition of a Glossary in a Collaborative Research Project and its Application to a European Network of Excellence. In Konstantas, D., Bourrières, J.-P., Léonard, M., and Boudjlida, N., editors, *Interoperability of Enterprise Software and Applications*, pages 311–322. Springer London.

Vicente, S., Prez, M., Garca, X., Gimeno, A., and Naval, J. (2006). Terregov: egoverment interoperability on a semantically driven world. In Konstantas, D., Bourrires, J.-P., Lonard, M., and Boudjlida, N., editors, *Interoperability of Enterprise Software and Applications*, pages 461–463. Springer London.

W3C (2008). Owl 2 web ontology language mof-based metamodel.

Westerhout, E. and Monachesi, P. (2007). Extraction of Dutch definitory contexts for eLearning purposes. *Proceedings of CLIN*, pages 1–15.