Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Dynamic Adaptors to Support Model-Driven Interoperability and Enhance Sensing Enterprise Networks

Carlos Agostinho*. Pedro Pinto.
Ricardo Jardim-Goncalves***

**Centre of Technology and Systems, CTS, UNINOVA, 2829-516 - Caparica, Portugal
(Tel: +351 212948527; e-mail: ca@uninova.pt; rg@uninova.pt)
** Department of Electrical Engineering, Faculty of Science and Technology, NOVA University of Lisbon, 2829-516 - Caparica, Portugal (e-mail: pmp18760@fct.unl.pt)*

Abstract: In an increasingly complex environment, the enterprises of the future should become Sensing Enterprises, evidencing intelligent, dynamic and self-organizing capabilities for understanding and responding to the shifts that impact the networks they belong to. Reconfiguration and reprioritization of industrial processes, information models, and even terminology is now seen as a requirement for survivability, which means that software systems need to become more agile. Hence, software adaptor technologies such as the ATLAS Transformation Language (ATL) are gaining momentum due to the potential to define and regulate peer-to-peer communication among networks of enterprise systems. At present, ATL is the de facto standard for transformations within the framework of model-driven interoperability (MDI), nonetheless it lacks the dynamism required to streamline complex systems, which demand adaptors to be frequently redesigned whenever information models change. This paper presents a novel solution, which aims to reduce the setbacks that arise from this lack of dynamism.

Keywords: Self-Sustainable Interoperability, Sensing Enterprise, Dynamic Networks, Model Transformations, ATL, Model-Driven Interoperability.

1. INTRODUCTION

The survival of enterprises in the near and long term future will depend on the ability to see their own role within the physical and social environment. For sure, short-term planning is essential, but understanding the myriad of new possibilities and shifts that are able to impact our society and global economy will be an important asset for the future. In fact, as analysed by Santucci et al. (2012), our digital society is redefining the *"enterprise"* in a context where *"the network is the business"*. These networks are the foundation of future enterprise systems (Chen et al., 2008).

The combination of two Internet research domains, namely the Internet of Things (IoT - www.theinternetofthings.eu) and the Future Internet Enterprise Systems (FInES - www.fines-cluster.eu), are giving rise to the "Sensing" capability as a new Quality of Being (QoB) necessary to overcome the challenges referred (Santucci et al., 2012). In its pursuit, and in the context of FInES, enterprises need improved dynamic decision support enablers capable of extracting internal and external information, and transforming it to knowledge that can be used in the benefit their business operations.

Indeed, the fact that enterprise networks are constantly growing environments represents a major increase of business opportunities, providing larger sensing and collaboration spectrum, but also a great challenge to interoperability enablers, in response to the exponential growth of complexity in the adaptation of their information systems (Agostinho & Jardim-Goncalves, 2009; Jardim-Goncalves et al., 2012). Nowadays, more than ever, interoperability is a key requirement in the implementation and maintenance of enterprise systems (e.g. ERP, CRM, etc.) and an essential property for development and growth. It is extremely important that systems compatibility is achieved as fast and soon as possible and in a sustainable form.

Research in Enterprise Interoperability (EI) suggests that organizations can seamlessly interoperate with others at all stages of development, as long as they keep their business objectives aligned, software applications communicating, and the knowledge and understanding of the domain harmonized (Berre et al., 2007; Li et al., 2010). The ideal would be to rely on dedicated knowledge models and international standards as information regulators among organizations, covering industrial areas and activities. However, interoperability, especially at the level of software applications, is typically obtained by point-to-point mappings, hardcoded to relate information models, services and terminology.

The above solution has been effective in environments in which a static model remains unchanged for years (e.g. large OEM databases). Nevertheless, the current reality dictates that systems, e.g. service systems as explored by Ducq et al. (2012), are increasing their dynamism to respond and adapt to new requirements and sensed realities. Dynamism is the new paradigm of interoperability and key to a sustainable interoperability (Agostinho & Jardim-Goncalves, 2009).

Technically, to address the whole issue, several model transformations languages arose, among them ATL which stands as the current de-facto standard (Eclipse foundation, 2011). This technology represents models as meta-models and relates their properties through rules that reflect mappings. Despite its robustness in solving the issues of interoperability between information models, each time that it is necessary to change relations between models, manual codification is required to recreate these rules. This behavior is a direct consequence of dynamism flaw.

This paper explores the Sensing Enterprise concept relating it with the emerging self-sustainable interoperability research topic, and proposes a technical a solution to improve the lack of dynamism required to achieve both. After discussing the 2 high level topics in section 2, section 3 includes a brief introduction to the paradigm of model-driven interoperability and existing developments, namely ATL technology. Section 4 explains a concrete solution to mitigate the lack of dynamism of enterprise systems and networks, offering the possibility of automatic ATL code generation. An industrial scenario for validation is included in section 5, and the paper concludes highlighting future perspectives of development.

2. SELF-SUSTAINABLE INTEROPERABILITY AND THE SENSING ENTERPRISE

2.1 Sensing Enterprise (SE)

The Sensing Enterprise concept was created by the FInES Cluster (2009) with the support of the European Commission, in the advent of the Augmented Internet (FInES Cluster, 2011). The community acknowledged the fact enterprises are desperately in need of innovative ideas to adapt, remain competitive, or sometimes simply survive in the digital era.

SE is an attempt to reconcile traditional non-native “Internet-friendly” organisations with the tremendous possibilities offered by the cyber worlds (from the clouds to the dust). It refers to *“an enterprise anticipating future decisions by using multi-dimensional information captured through physical and virtual objects and providing added value information to enhance its global context awareness”* (Santucci et al., 2012). The Sensing Enterprise envisions the enterprise as a smart complex entity capable of sensing and reacting to stimuli, by integrating decentralised intelligence, context awareness, dynamic configurability and sensorial technology into its decision-making process (Danila et al., 2013).

Despite being highly acknowledged at the European level, the concept remains as a research frame, being a part of the FInES Research Roadmap 2025 (FInES Research Roadmap Task Force, 2012), with the goal to harmonize developments from both the IoT and FInES “worlds” rather than delivering something very tangible or a single solution. In fact, as recognized by Santucci et al. (2012), even the EU should aim at developing public-private partnership projects not limited to software providers, with a long-term vision (at least 10 years) and clear scientific roadmaps in order to achieve significant breakthroughs.

2.2 Self-Sustainable Interoperability

Self-Sustainable Interoperability is a concept introduced by Agostinho & Jardim-Goncalves (2009), which can be seen as the next evolution step after the Semantic Interoperability in the context of Enterprise Interoperability.

Today EI has evolved from a complex technical business systems interconnection issue to a larger domain, with multiple dimensions and multidisciplinary issues, which need to be addressed using a more systemic and holistic way. It has matured, and in order to evolve further, needs sustainability, i.e. to be build upon a science base capable of providing solid grounds for dynamicity, as well as repeatability of processes and solutions in multi-domain networks (Jardim-Goncalves et al., 2012; Ducq, Chen & Doumeingts, 2012).

Defined as the *“Interoperability that convenes the needs of the present without compromising the ability of future changes, meeting new system requirements, and performing adequate adaptation and suitable management of the transitory elements”*, sustainable interoperability draws ideas from complex systems science, especially Complex Adaptive Systems (CAS), in order to offer enterprise information systems the possibility of dealing with internal network dynamics and of successfully facing interoperability disruptions, i.e. harmonization breaking.

As developed in Jardim-Goncalves et al. (2012) and followed by Danila et al. (2013), the sustainable interoperability of the Sensing Enterprise is supported by the sustainability recovery cycle running along the adaptive organization lifecycle, and consisting of: (1) *Discovery capabilities*, used to detect network harmonization breaking; (2) *Learning and adaptability*, in order to acquire knowledge about the occurred changes and the specific adaptation required; (3) *Transient simulation and decision*, used for simulation and evaluation of adaptations’ impact and to understand how a network of systems will suffer during the transient period; and (4) *Notification facilities*, used to enable physical and computational connectivity between nodes.

2.3 Discussion and Paper Positioning

Apart from the FInES cluster documentation, there is a lack of literature available addressing the SE subject. Several scenarios have been explored in discussion groups and workshops however much remains to be done to improve the theoretical foundations and the specification of the logical links among neighbouring concepts that exist or emerge at the confluence of the Internet of Everything. This leaves the concept quite open in terms of interpretation and wide in terms of research targets.

In parallel with the challenges of the FInES Research Roadmap 2025 (FInES Research Roadmap Task Force, 2012), several (perhaps more specific) research areas can be identified of potential interest to enable advancements in the State-of-the-Art towards the Sensing Enterprise, under the domains of IoT and FInES, namelly:

(i) *Cyber-Physical Systems (CPS-cyberphysicalsystems.org)* that are engineered systems in which physical components are tightly intertwined with computational elements. Borders are “liquefied” and they are believed to revolutionize not only production but also mobility and

healthcare, to facilitate the communication between intelligent context aware entities (Nikolaus, 2013);

(ii) *Smart Tags & RFID*, which are relevant components for CPS and Wireless Sensor Networks development, and are already spread in many industries starting from automotive assembly lines, pharmaceutics, clothing, etc.;

(iii) *Ambient Connectivity*, to bring the ability to assume connectivity among everything, anywhere and anytime independently of the means and providers (Frankston, 2009; Frankston, 2013);

(iv) *Competitive and Customer Intelligence,* applying methods such as crawling, scraping or data mining to gather information about the surrounding environment (including competitors and markets) as well as knowledge of technological developments (Gilad, 2008);

(v) *Model-Based Systems and Service Engineering*, which promotes modelling to address many of the limitations of the traditional document-based engineering approach. It provides a more rigorous means for capturing and integration of requirements, design, analysis, verification and validation throughout the system later life cycle phases (INCOSE, 2007; Ducq, Chen & Alix, 2012). This enables more understanding between development teams and the other stakeholders, as well as traceability features that facilitate properties such as backward compatibility where the enterprises of the Future should be able to interoperate with non-evolved enterprises.

(vi) *Self-Sustainable Interoperability,* whose principles have already been explained and can here be enlarged into becoming the "glue" to the previous areas. Connected to other interoperability research (e.g. EI, model-driven interoperability, etc.), it includes as well, other relevant topics such as complexity management, model and service matching, transformation, monitoring or strategic decision-making. It can be an important asset in the development of FInES, and in particular the SE, as enterprises will need to permanently adapt to meet their requirements while maintaining interoperable.

If one were to classify, it would be possible to say that areas (i-iii) are closer to the domain of IoT while the remaining ones are closer to FInES. Nevertheless, both domains are closely related, and the Sensing Enterprise cannot exist without one another. The research presented hereafter is more focused on self-sustainable interoperability, namely on the extension of the model and service transformation sub-topic, to include dynamicity concerning the relationship among different enterprises. Model-Driven Interoperability (MDI) is analysed as the methodological framework that provides a conceptual and technical background for the developments.

3. MATCHING AND TRANFORMATION IN MODEL-DRIVEN INTEROPERABILITY

Model transformation is not a new concept. It has been broadly used in Model Driven Development/Engineering (MDD/MDE) methods where models and their roles in the development process should change from contemplative (e.g., used for documentation) to productive, thus envisaging transformations from high-level business models focusing on goals, roles and responsibilities down to detailed use-case and scenario models for execution (MSEE Partners, 2012).

Based on model transformation and model mapping morphisms, the Model Driven Architecture (MDA - www.omg.org/mda/) is one of the most relevant realizations of MDD, while Model Driven Interoperability (MDI) is a recognized extension, envisaging to solve interoperability problems between enterprises not only at the application and code levels, but also at business levels with the support of semantic technologies, e.g. ontologies.

MDI is supported through the extensive use of morphisms in vertical and horizontal integration of the multiple abstraction levels defined in the Reference Model for MDI (INTEROP Partners, 2007; InterOP-VLab, 2013). As detailed in Agostinho et al. (2012) and illustrated by Fig. 1, vertical transformation morphisms imply changes in abstraction (either specialization or generalization) to unify every step of the development of a software application from its start as a *Computation Independent Model (CIM)* of the application's business requirements, through *Platform Independent Model(s) (PIM)* specifying the structure, functions and behaviour, down to one or more *Platform Specific Models (PSM)*, which lead to generated code and deployable applications. With horizontal transformation morphisms, companies can specify P2P mappings at any of the models to translate data from one format to the other, thus allowing an exchange of information.

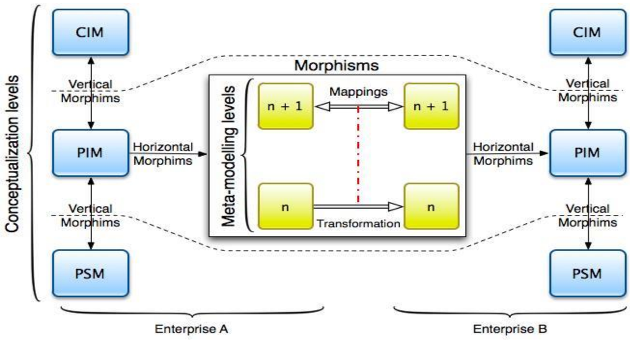

Fig. 1. MDA/MDI Simplified Conceptualization

3.1 *Model Morphisms (MoMo)*

This concept is gaining some meaning in computer science, more exactly in systems interoperability to describe the relations (e.g. mapping, merging, transformation, etc.) between two or more information structures (INTEROP Partners, 2005). MoMo is an attempt to formalise model based operations, drawing the terminology from consolidated mathematical areas, such as set or graph theory, theory of functions, and adapting it to the modelling context.

Nevertheless, classical descriptions are not tailored to deal with the dynamicity of the multiple information models used by the enterprise systems of today's networks. Combining their use with a MoMo ontology, brings the possibility to trace model processing operations, and when properly instantiated, the ontology will provide a valuable knowledge-base for reasoning purposes (D'Antonio et al., 2006).

In a related work, the authors collaborated in the proposition of a communication mediator (CM) ontology to address traceability as the ability to interrelate uniquely identifiable object versions in a way that can be processed by Human or system (Ferreira et al., 2012). Morphisms are chronologically modelled and updated, and such ontology serves not only as a meta-level knowledge base of model relations, but also to make educated suggestions on future behaviour based on past relationships among the same enterprise concepts.

3.2 Transformation Principles

Many scenarios illustrate the value of model transformation. Perhaps the translation of data from one domain to the other is the most known, but other examples include visualization, reengineering, reducing the complexity, etc.

Model transformation describes the process of converting one model description into another, thus not only the model itself has to be analysed but also the meta-models behind the source and target systems. As envisaged in the generic transformations architecture of OMG (2003), the idea is that when performing a transformation "$\tau(A,B)$" at a certain meta-modelling level "n", this transformation has (implicitly or explicitly) to be designed by taking into account mappings "$\theta(A,B)$" at level "$n+1$" (see Fig. 1). Once the "$n+1$" level mapping is complete, executable languages such as the Atlas Transformation Language (ATL), can be used to implement the transformation itself (Agostinho et al., 2012)

However, besides being complex, the definition of mapping morphisms is a time consuming activity as concepts with the same name may not be related, or vice-versa. A concept could be a generalisation or specialisation of another object, and even non-matching concepts may need to be analysed in search for common attributes. This may lead to complex mapping algorithms (direct, functional, or rule-based) and even loss of information. It is therefore important to capitalize the mapping activity by defining it explicitly so that it can be reused whenever needed, and support the dynamicity demanded by Sensing Enterprise networks.

3.3. ATLAS Transformation Language (ATL)

Within the set of transformation technologies, ATL has come to occupy a prominent place as a *de-facto* standard solution in this problematic of systems interoperability. ATL is a rule-based model-to-model transformation language that is not so rigid as the "official" OMG standard (i.e. QVT), and provides unidirectional hybrid declarative(mostly)-imperative constructs in order to ease the specification of mappings.

One great advantage of ATL is being fully and perfectly integrated into Eclipse (www.eclipse.org/atl/), well documented and providing a broad and active community ready to help. Fig. 2 shows the Eclipse architecture of ATL that complies with the generic transformations architecture of OMG (2003). Both source and target models (A and B) are described by meta-models (at meta-modelling level 2) following the ECore EMOF (level 3) specification, a MOF-based variant of the OMG standard (OMG, 2008). Likewise ATL itself is represented by a meta-model.

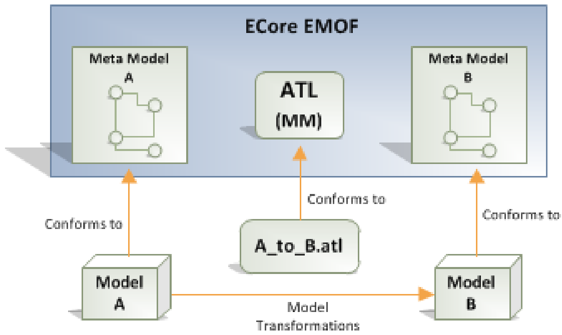

Fig. 2. ATL Transformations Architecture

In ATL, a transformation is composed by a set of rules ("matched rules") that define how the source model elements are linked, navigated enabling and instantiating the elements of the target model. 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"). Bézivin et al. (2003) detail some ATL-based implementations.

4. DYNAMICITY IN SENSING ENTERPRISE NETWORKS

The technical contribution of this research work is detailed in this section and framed with complementary developments. Reconfiguration and reprioritization of industrial processes, information models, and even terminology is seen as a requirement for survivability, which means that software systems need to become more agile. In pursuit of SE networks that enable that goal, the authors consider software adaptor technologies and the morphisms that define them need to be dynamic, facilitating semi-automatic:

1. Suggestion of *Mappings*;
2. Generation of *Transformations* code;

4.1. Technical Architecture

Fig. 3 illustrates the technical architecture of the proposed solution. Using the OMG (2003) general architecture for transformations as a basis, it relies on three main components, i.e. the Mapping Tool (not the focus the paper), the Knowledge Acquisition Engine, and the ATL Generator as a means to achieve the final goal of having semi-automatic and dynamic transformations. These components are supported by dedicated MoMo ontologies, in particular the communication mediator (CM) ontology as specified in Ferreira et al. (2012).

The Mapping Tool uses models at the modelling level n+1 to enable the Human user to identify relationships between 2 systems, or different abstractions of the same system (e.g. CIM, PIM, or PSM). Graphical browsing of models and data visualization plays an important role in the interoperability achievement. Hence, using the model-mapping component it is possible to visualize graphically the source and target models and to define the conditions and relations in a simple and intuitive way. Besides the traditional direct connectivity, many of the mapping morphisms are imperfect and the semantic mismatches found along the various model elements being mapped, need to be identified as well. A prototype that can be adapted to this component has been already developed in a complementary work (Agostinho et al., 2012).

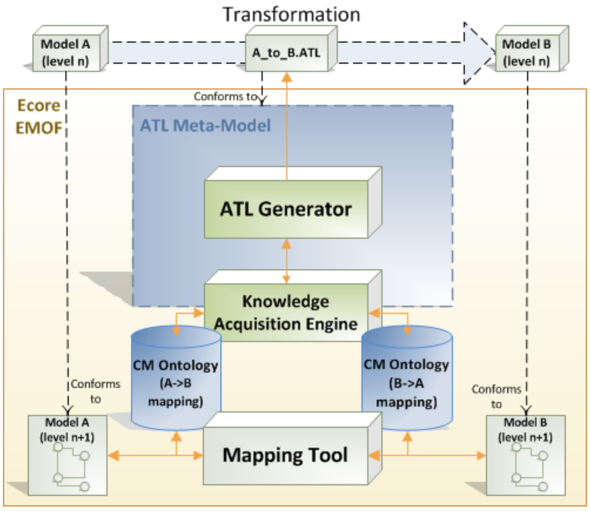

Fig. 3. Technical Architecture to Enable Dynamic Morphisms

It is envisaged that each pair of mapping morphisms (A->B amd B->A) is stored on dedicated Communication Mediators. The objective is that each system has its own CM ontology to maintain a traceable record of relationships to support monitoring and intelligence activities of the Knowledge Acquisition Engine. When a significative amount of information is available, service agents can reason on the knowledge stored, querying and relating it with a specific problem to suggest an initial set of mappings.

The Knowledge Acquisition Engine component also provides the interface among the mapping definition and the ATL Generator, through the CM ontologies. The aim of this last component is to use the stored meta-level mapping morphisms and provide the functionality of automatic generation of ATL rules (in a physical ".ATL" file) that will allow to transform the model A into model B.

4.2. Dynamic Mappings in a Self-Sustainable Interoperability

The Mapping Tool works even when no previous knowledge is available to reason upon. In that case the Human has to do all the work. However, following the event of harmonization breaking, the CM ontologies will already have information of the initial set of mappings, and may contribute to recover the network interoperability status which is compromised and needs to fixed dynamically consuming the least possible resources (human, temporal, material or financial).

Therefore, and to complement the work started by Ferreira et al. (2012), focused on learning and suggestion of terminology associations, it would be relevant to generalize a method to respond to situations caused by versioning morphism(s), e.g. evolution of information models. The methodology conducts the following course of actions to generate a new mapping: (1) Analyse detected versioning; (2) Analyse existing mapping; (3) Propose a new mapping.

The new mapping type remains the same as the existing one, and a functional composition of mapping morphisms is possible. For instance, a change in a structural property (model element) of information systems will cause new mappings of that type, whereas if one were relating only semantic concepts, any change in terminology would force a new mapping of concepts.

Fig. 4 shows an abstraction where an enterprise system needs to respond and propose a new morphism (*w* or w^{-1} depending on the point of view of the harmonization breaking), after an evolution of a specific model ($g = M1 \rightarrow M1'$) is detected (versioning). The figure also represents the existing mapped relation (*f*) between the two models (*M1 and M2*), f^{-1} for the opposite situation, and the respective model elements that are targeted by the algorithm (*x* from *M1*, *y* from *M2*, and *z* from *M1'*). Following equation (1), mappings are formalized in a generalised way, where for every *x* and *y* that are elements of a Model, *f* is the tuple (belonging to the Mappings domain) that enables the transformation morphism τ.

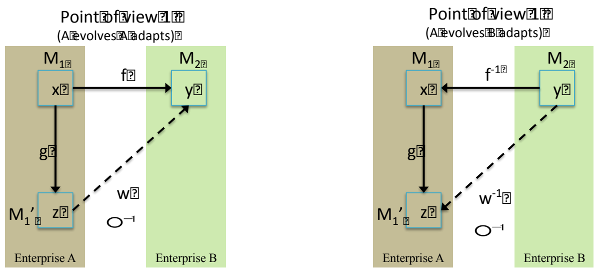

Fig. 4. Generation of New Mapping Morphisms

$$\forall x \in M_1, y \in M_2, and\ f \in Maps(M_1 \times M_2), \quad (1)$$
$$if\ \exists\ f(x,y)\ then\ \exists\ \tau(x,f) = y$$

$$\forall\ x,z \in M_1, y \in M_2, and\ w \in Maps(M_1 \times M_2),$$
$$if\ \ \exists\ \tau(x,f) = y\ and\ \tau(x,g) = z, \quad (2)$$
$$then\ w = f \circ g^{-1}\ and\ \tau(z,w) = y$$

$$\forall\ x,z \in M_1, y \in M_2, and\ w \in Maps(M_1 \times M_2),$$
$$if\ \ \exists\ \tau(y,f^{-1}) = x\ and\ \tau(x,g) = z, \quad (3)$$
$$then\ w = g \circ f^{-1}\ and\ \tau(y,w^{-1}) = z$$

Applying mathematics transitivity principle of binary relations, it is possible to calculate the new mapping, i.e. for all *x*, *y* and *z*, if *xRy* and *yRz* exist, then *xRz* also applies. Equation (2) claims that having a relation from the element *x* to the element *y* (designated by *f*) and another from element *x* to element *z* (designated by *g*), then it is reasonable to say that exists an inverse relationship g^{-1} (from *z* to *x*) that enables a relation from the element *z* with the element *y* (designated *w*), composed of g^{-1} and *f* ($f \circ g^{-1}$). This automatism still has some restriction and is only valid in the case of lossless transformations.

4.3. Knowledge Acquisition using SPARQL

SPARQL (www.w3.org/TR/rdf-sparql-query/) is nowadays the standard language for accessing information in Semantic-Web formats (e.g. OWL and RDF), able to unambiguously retrieve and manipulate data stored in the same format.

Being an ontology, the CM stores knowledge about the meta-models and their mappings using OWL, thus the SPARQL arose as natural solution for handling existing morphisms. In the scope of the Knowledge Acquisition Engine, it enables the interrogation of amalgamated datasets to provide access to their combined information. From there, it becomes possible to build any kind of intelligence (through semantic reasoners or mining applications) required for the semi-automatic suggestion of mappings (section 4.2) or for the automatic generation of transformations code (section 4.4).

```
PREFIX MO:<http://www.owl-ontologies.com/
MediatorOntologyD.owl#>
SELECT ?Morphism ?ID
WHERE {
   ?Morphism MO:morphismType "mapping";
            MO:ID ?ID;}
```

Fig. 5. SPARQL Query Example

A morphism in the CM represents the relation between two information model elements, A (relating) and B (related). This way, the query of Fig. 5 retrieves all the *morphisms* of the type "mapping". The result can be set (casted) as an object and from that, one can use the available functions to get the related elements. With the knowledge given by these morphisms, ATL rules can be generated automatic at least in the cases of direct mappings (1to1).

Similar queries can be executed in order to obtain directly the information *ModelElements*, e.g. *class* or *property*.

4.4. Methodology for ATL Generation

Using SPARQL queries, it is possible to automatically extract all the knowledge stored within the CM and instantiate the previously defined mappings into JAVA objects, which reflect the same structure as the CM. This provides the means to implement an injector that feeds data into the CM Ecore metamodel, and from there use ATL technology to generate ATL itself, and create a physical transformation file that reflects the mapping stored within the CM (see Fig. 6).

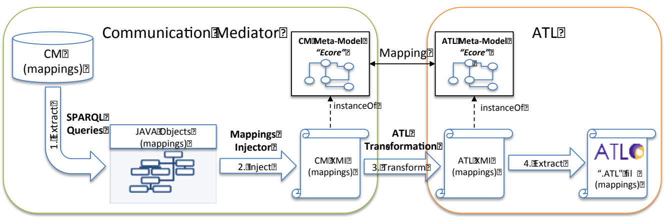

Fig. 6. ATL Generation

Arbitrary documents can be used as source or target of ATL transformations. However, this cannot be done directly but rather through specific injectors that need to be implemented. In practice, there is the need to change from another technical space to the "model engineering technical space", having models serialized in XMI format and conforming to the ATL Ecore implementations (step 2).

The transformation envisaged in step 3 follows the principles previously explained in this document. Finally, the last step is required to extract the code (ATL Rules) from the XMI file resulting from the transformation. Here, we use an ANT task that enables to save the model using the pre-defined ATL extractor and create the ".atl". This file contains the rules for transforming the object A into B as defined in the CM.

5. SCENARIO AND TEST EXAMPLE

The research contributions here presented are being validated in the scope of two EU-funded research projects from the Factories of the Future domain. More specifically MSEE (www.msee-ip.eu) has provided the opportunity to work in vertical integration morphisms from CIM to TIM levels, while IMAGINE (www.imagine-futurefactory.eu) validated horizontal morphisms. The following scenario has been extracted from the IMAGINE furniture industry living-lab, where companies are joining the platform to create Dynamic Manufacturing Networks (DMN) (Papakostas et al., 2012).

The DMN provides companies support across the manufacturing lifecycle, enabling a view of information from various sources and systems, dealing with the new business models of collaboration and self-organisation, thus capable meeting the changes of the customer's requirements. In the project, the DMN is created with the support of the IMAGINE platform, which collects data from the different enterprise systems following a "blueprint" format that is capable of representing *Partner, Product, Process,* and *Quality* data. Refer to Ferreira et al. (2013) for further detail on the blueprints.

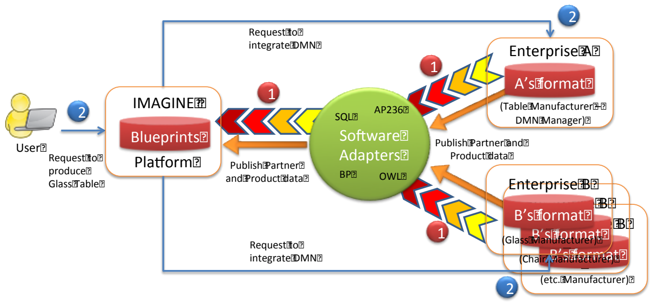

Fig. 7. Scenario from IMAGINE Furniture Living-Lab

The scenario of Fig. 7 illustrates a simple usage of the IMAGINE platform. Here, the different enterprises that want to integrate future DMNs need to publish certain data about themselves (in this case partner/company and product) to enable the platform to find them in the advent of specific production requests. Flow nº. 1 shows the initial sequence, where due to the amount systems and their data exchange formats, software adaptors and data transformations (A->Blueprint, B->Blueprint, etc.) are required. Each company can manage its own adaptor depending of its privacy policy, or in alternative it can share centralized network adaptors. Flow 2 demonstrates the interest of the IMAGINE platform, where after receiving a request to produce a "Glass Table", it reasons on the information and dynamically proposes A and B to integrate the DMN and manufacture the product.

The architecture proposed in section 4 has been tested in flow nº. 1, hence and a small example focusing on the blueprints and A's format is presented next. Mappings had to be defined in order to simulate the semi-automatic suggestion of mappings, as well as the generation of transformations code

(Table 1). It is possible to see that, in this case, every attribute of A has a direct relationship with the blueprint structure, however "capacity" is expressed by system A as the number of units produced by working hour, while in the blueprint (for harmonization with other companies), it is expressed as a daily average. Operating 24 hours per day, this produces:

$$hasAverageCapacityRate = f(capacity) = capacity \times 24 \quad (4)$$

Table 1. Machinery to Equipment (A ->Blueprint Mapping)

Machinery - Equipment mapping (f)	
A's format (Machinery)	**Blueprint (Equipment)**
name (string)	hasName (string)
type (relation)	hasCategory (relation)
certification (string)	hasCertification (string)
description (string)	hasDescription (string)
operationHours (int)	hasTimeInOperationMonth (int)
capacity (int)	hasAverageCapacityRate (int)

5.1. Suggestion of Mappings

Lets consider that enterprise A buys a new module for their machines that enable them to operate even faster and therefore justify a change of their nomenclature to represent "newCapacity", from units per hour to units per minute:

$$newCapacity = g(capacity) = capacity/60 \quad (5)$$

We are in the presence of a versioning morphism that fits the premises of equation (2). The new mapping remains the same with the exception of the relationship given by equation (6):

$$\begin{aligned} &hasAverageCapacityRate = w(newCapacity) \\ &= f \circ g^{-1}(newCapacity) = f\left(g^{-1}(newCapacity)\right) \\ &= f(60 \times newCapacity) = 24 \times 60 \times newCapacity \end{aligned} \quad (6)$$

5.2. Generation of ATL Code

Applying the algorithm for the generation of ATL, the new *w* relationship, resulting from the evolution described previously, can be generated automatically, producing the result of Fig. 8. The table transformation is described by a "matched rule", and the attributes mappings are explicitly represented in the "to" part of the rule.

```
rule W_newMachinery_To_Equipment{
  from
    f: EnterpriseA!newMachinery
  to
    t: IMAGINE!Equipment(
        hasName <- f.newName,
        hasCategory <- f.newType,
        hasCertification <- f.newCertification,
        hasDescription <- f.newDescription,
        hasTimeInOperationMonth<- f.newOperationHours,
        hasAverageCapacityRate <- 24*60* f.newCapacity)
}
```

Fig. 8. ATL Rule to Transform A's newMachinery Table to IMAGINE's Equipment

6. CONCLUSIONS AND FUTURE WORK

The innovative vision on the Sensing Enterprise is expected to have huge impact at economic and societal levels. In that context, paraphrasing Santucci et al. (2012) *"it is difficult to conclude on something that is just beginning"*, but definitely, reconfiguration and reprioritization of industrial processes, information models, and even terminology is now seen as a requirement for survivability, which means that systems need to become more agile. This paper addresses FInES applied research to endorse the vision that, in the future, innovation and developments on Sensing Enterprise networks can be supported by a self-sustainable interoperability.

The authors defend that the development of dynamic software adaptors to support model-driven interoperability, is one of the key challenges to enable SE and self-sustainable interoperability, since it allows the evolution of businesses, models and tools, while maintaining enterprise networked environments interoperable. In fact, based on the theory of CAS, analysing networks as "white boxes" of communication enabling relationships (morphisms), and applying model-driven technologies to increase automation may provide the answer to more efficient behaviour at larger systems' scale.

The choice of MDI as the enabling technology was motivated by morphisms modularity and repeatability through the existing landscape of tools available to support horizontal and vertical transformations. Developments are being validated in the scope of two EC projects, which are providing interesting feedback to further improvements and future work.

Indeed, despite being an auspicious work, the current automatism concerning the mappings proposition in response to harmonization breaking has some restrictions, which could probably be improved if more reasoning on the CM ontologies would be used to complement the structural mappings (Knowledge Acquisition Engine). In fact, the current algorithm for the generation of new mapping morphisms is valid only in the case of a lossless and direct relationship; otherwise, data will be lost by the functional composition and could violate the inverse function premisses. Moreover, under similar conditions (direct relationships) it is clear that automatic generation of ATL code is a reality. More complex mappings (1toN and Nto1) have also been also tested but further research is still pending in the case where advanced functionalities of ATL are required (e.g. "called rules", "lazy rules", etc.).

The platform proposed will remain target of research and development. Next steps are targeted to address the features and limitations identified, as well as a larger scale population of the CM ontology to take advantage of the full benefits of ontologies and semantic technologies.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union 7th Framework Programme (FP7/2007-2013) under the MSEE (FP7-284860), IMAGINE (FP7-285132) & OSMOSE (FP7-610905) grant agreements.

REFERENCES

Agostinho, C., Černý, J. & Jardim-goncalves, R. (2012) MDA-Based Interoperability Establishment Using Language Independent Information Models. In: *4th International IFIP Working Conference on Enterprise Interoperability (IWEI 2012)*. Harbin, China.

Agostinho, C. & Jardim-Goncalves, R. (2009) Dynamic Business Networks: A Headache for Sustainable Systems Interoperability. In: *On the Move to Meaningful Internet Systems: OTM 2009 Workshops (EI2N 2009)*. Vilamoura, Portugal.

Berre, A., Elvesæter, B., Figay, N., Guglielmina, C., Johnsen, S., Karlsen, D., Knothe, T. & Lippe, S. (2007) The athena interoperability framework. In: *3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA'07)*. Funchal, Portugal.

Bézivin, J., Dupé, G., Jouault, F., Pitette, G. & 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*.

Chen, D., Doumeingts, G. & Vernadat, F. (2008) Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry*, 59 (7), pp.647–659.

D'Antonio, F., Missikoff, M., Bottoni, P., Hahn, A. & Hausmann, K. (2006) An ontology for describing model mapping/transformation tools and methodologies: the MoMo ontology. In: *EMOI'06 - Open InterOP Workshop on Enterprise modelling and Ontologies for Interoperability*. Luxemburg.

Danila, C., Stanescu, A., Goncalves, R., Agostinho, C., Florea, A. & Serbanescu, C. (2013) Towards a holistic approach for Developing Sensing Enterprise Modelling Framework. In: *Proceedings of 7th IFAC Conference on Manufacturing Modelling, Management, and Control, 2013*. St. Petersburg, Russia.

Ducq, Y., Chen, D. & Alix, T. (2012) Principles of Servitization and Definition of an Architecture for Model Driven Service System Engineering. In: *4th International IFIP Working Conference on Einterprise Interoperability (IWEI 2012)*. Harbin, China.

Ducq, Y., Chen, D. & Doumeingts, G. (2012) A contribution of system theory to sustainable enterprise interoperability science base. *Computers in Industry*, 63 (8), pp.844–857.

Eclipse foundation (2011) ATL - A Model Transformation Technology [Internet]. From: <http://eclipse.org/atl/>

Ferreira, J., Agostinho, C., Jardim-Goncalves, R. & Sarraipa, J. (2012) Monitoring Morphisms to Support Sustainability of Interoperability in the Manufacturing Domain. In: *14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2012)*. Bucharest, Romania.

Ferreira, J., Ferro de Beça, M., Agostinho, C., Nunez, M.J. & Jardim-Goncalves, R. (2013) Standard Blueprints for Interoperability in Factories of the Future (FoF). In: *Proceedings of 7th IFAC Conference on Manufacturing Modelling, Management, and Control, 2013*. St. Petersburg, Russia.

FInES Cluster (2009) *Future Internet Enterprise Systems (FInES) Cluster Position Paper: Final Version (Version 3.0)*. Available from: <http://ec.europa.eu/digital-agenda/events/cf/ict2010/document.cfm?doc_id=12512> [Accessed 8 November 2013].

FInES Cluster (2011) *Future Internet Enterprise Systems (FInES) Position Paper on Orientations for FP8 "A European Innovation Partnership" for European Enterprises: Version 3.0*. Available from: <cordis.europa.eu/fp7/ict/enet/documents/fines-position-paper-fp8-orientations-v3.pdf> [Accessed 7 Nov. 2013].

FInES Research Roadmap Task Force (2012) *FInES Research Roadmap 2025: version 3.0*. <www.fines-cluster.eu/jm/Manage-Documents/Download-document/323-FInES-Research-Roadmap-2025-v3.0.html> [Accessed 18 Nov. 2013].

Frankston, B. (2009) Ambient Connectivity: An Introduction [Internet]. From: <http://frankston.com/public/?n=IAC> [Accessed 8 Nov. 2013].

Frankston, B. (2013) The Internet of Things Versus the Access Framing. *IEEE Consumer Electronics Magazine*, (july), pp.56–57.

Gilad, B. (2008) The Future of Competitive Intelligence: Contest for the Profession's Soul. *Competitive Intelligence Magazine*, 11 (5), p.22.

INCOSE, T.O. (2007) *Systems Engineering Vision 2020 (INCOSE-TP-2004-004-02)*. INCOSE.

INTEROP Partners (2007) *Deliverable DTG2.3: Report on Model-Driven Interoperability*. INTEROP NoE Project (FP6 IST-1-508011).

INTEROP Partners (2005) *Deliverable DTG3.1 (MoMo.2): MoMo Roadmap*. INTEROP NoE Project (FP6 IST-1-508011).

InterOP-VLab (2013) Model Driven Interoperability [Internet].<en.wikipedia.org/wiki/Model_Driven_Interoperability> [Accessed 11 November 2013].

Jardim-Goncalves, R., Agostinho, C. & Steiger-Garcao, A. (2012) A reference model for sustainable interoperability in networked enterprises: towards the foundation of EI science base. *International Journal of Computer Integrated Manufacturing*, 25 (10), pp.855–873.

Li, Q., Zhou, J., Peng, Q.-R., Li, C.-Q., Wang, C., Wu, J. & Shao, B.-E. (2010) Business processes oriented heterogeneous systems integration platform for networked enterprises. *Computers in Industry*, 61 (2), pp.127–144.

MSEE Partners (2012) *Deliverable D11.3: MDA/MDI Model Transformation - Application to MDSEA (M8 Issue)*.

Nikolaus, K. (2013) Building the Nuts and Bolts of Self-Organizing Factories. *Siemens Pictures of the Future: Manufactoring and Innovation (Industry 4.0)*. From: <www.siemens.com/innovation/apps/pof_microsite/_pof-spring-2013/_pdf/en/Self-organizing_factories_EN.pdf>

OMG (2003) *MDA Guide Version 1.0.1 (omg/2003-06-01)*. Object Management Group.

OMG (2008) Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification - Version 1.0 (formal/2008-04-03).

Papakostas, N., Efthymiou, K., Georgoulias, K. & Chryssolouris, G. (2012) On the configuration and planning of dynamic manufacturing networks. *Logistics Research*, 5 (3-4), pp.105–111.

Santucci, G., Martinez, C. & Vlad-câlcic, D. (2012) The Sensing Enterprise. In: *FInES Workshop at FIA 2012*. Aalborg, Denmark. Available from: <www.theinternetofthings.eu/sites/default/files/%5Buser-name%5D/Sensing-enterprise.pdf> [Acc. 6 Nov. 2013].