Tuple-based Semantic and Structural Mapping for a Sustainable Interoperability

Carlos Agostinho[1,2], João Sarraipa[1,2], David Goncalves[1], Ricardo Jardim-Goncalves[1,2]

[1] Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal {djg15361}@fct.unl.pt
[2] Centre of Technology and Systems, CTS, UNINOVA, 2829-516 Caparica, Portugal {ca, jfss, rg}@uninova.pt

Abstract. Enterprises are demanded to collaborate and establish partnerships to reach global business and markets. However, due to the different sources of models and semantics, organizations are experiencing difficulties exchanging vital information electronically and seamlessly, even when they operate in related business environments. This situation is even worst in the advent of the evolution of the enterprise systems and applications, whose dynamics result in increasing the interoperability problem due to the continuous need for model adjustments and semantics harmonization. To contribute for a long term stable interoperable enterprise operating environment, the authors propose the integration of traceability functionalities in information systems as a way to support such sustainability. Either data, semantic, and structural mappings between partner enterprises in the complex network should be modelled as tuples and stored in a knowledge base for communication support with reasoning capabilities, thus allowing to trace, monitor and support the stability maintenance of a system's interoperable state.

Keywords: Interoperability, Model Morphisms, Semantic Matching, Knowledge Representation, Sustainable Interoperability.

1 Introduction

In the emerging society, characterized by the globalization phenomena, technological evolution and constant financial fluctuations, knowledge is a major asset in people's lives. It is and will remain being, in the future, the principal factor for competition both at personal and organizational levels, conducting tendencies at global markets. The outburst of advanced Web technologies, knowledge bases and resources all over the world, is levelling markets as never, and enabling organizations to compete on an equal basis independently of their size and origin [1].

The traditional way of doing business is not providing the expected efficiency. Nowadays, companies do not survive and prosper solely through their own individual efforts. Each one's success also depends on the activities and performance of others to whom they do business with, and hence on the nature and quality of the direct and indirect relations [2]. These involve a mix of cooperative and competitive elements, and to cope with them, organizations need to focus on

their core competencies by improving their relationships with customers, streamlining their supply chains (SCs), and by collaborating with partners to create valued networks between buyers, vendors and suppliers [3][4][5]. Indeed, in most cases, a single company cannot satisfy all customers' requirements, and where once individual organizations battled against each other, today the war is waged between networks of interconnected organisations [6], e.g. SCs. Therefore, to succeed in this complex environment, enterprise systems need to be interoperable, thus being able to share technical and business information, within and across organisations in a seamless and sustainable manner [5][7]. In this sense, sustainability appears on the context of this paper, related to the information systems (IS) ability to smoothly accommodate technical disturbances in a network of organisations, without compromising the overall network interoperability state.

2   Contribution to Technological Innovation and Sustainability

If systems are only partially interoperable, translation or data re-entry is required for information flows, thus incurring on several types of costs. For example, in SCs if the lower tiers do not have the financial resources or technical capability to support interoperability, their internal processes and communications are likely to be significantly less efficient, thus harming the performance of the entire network. This way, achieving an interoperable state inside heterogeneous networks, is still an ongoing challenge hindered by the fact that they are, intrinsically, composed by many distributed hardware and software using different models and semantics [8]. This situation is even worst in the advent of the evolution of enterprise systems and applications, whose dynamics result in increasing the interoperability problem with the continuous need for model adjustments and semantics harmonization: retail and manufacturing systems are constantly adapting to new market and customer requirements, thus answering the need to respond with faster and better quality production; new organizations are constantly entering and leaving collaboration networks, leading to a constant fluctuation and evolution of system models. All these factors are making interoperability difficult to sustain [9].

Due to this constant knowledge change, ontologies and model mappings are not static and there is always some information to add to a knowledge representation system. Preliminary theories have been advanced in specific scientific disciplines, such as biology and ecology, to explain the importance and evolution of complexity in living systems [10]. Also, some researchers have attempted to extrapolate results from a "general systems theory" or "complexity theory" that could explain the importance of the behaviour of systems in all fields of science [11][12]. These theories view all systems as dynamic, "living" entities that are goal-oriented and evolve over time, thus, IS should be able to manage its dynamics, learning during its existence and being in a constant update [13].

Based on that assumption, this paper contributes for a sustainable interoperable environment, proposing an approach inspired on complex adaptive systems (CAS) that use monitoring and traceability functionalities to act at the network micro level (i.e. local IS) and support sustainability at the macro level (i.e. the network). Data, semantic, and structural mappings are proposed to be modelled as traceable tuples and integrated in knowledge bases dedicated to managing mismatches during

communications. Section 3 summarizes different ways to represent and formalize model morphisms and semantic matching; Section 4 defines the concept of tuple for semantic and structural mapping, as well as the knowledge base for communication support; Section 5 presents a case study scenario for validation; and finally, in Section 6, the authors conclude and outlook on future work.

3   Models and Associated Concepts

3.1   Models

Either being used in the form of traditional databases, architectural models, or domain ontologies, models can be described on multiple formats, languages, expressiveness levels, and for different purposes [14][15][16]. A model can be characterized according to four dimensions [17]: *Metamodel* - the modelling primitives of the language for modelling (e.g. ER, OWL, XSD) are represented by a set of labels defined in the metamodel; *Structure* - corresponding to the topology associated to the model schema; *Terminology* - the labels of the model elements that don't refer to modelling primitives; *Semantics* - given a "Universe of Discourse", the interpretations that can be associated with the model.

This way, model operations can be classified as acting on any of these dimensions.

3.2   Model Morphims (MoMo)

In mathematics, "Morphism" 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 [17]. Recently, this concept as been gaining momentum applied to computer science, namely to systems interoperability. This new usage of "morphism" specifies the relations (e.g. mapping, merging, transformation, etc) between two or more information model specifications (M as the set of models). Therefore, a MoMo describes a model operation.

In this context, the research community identifies two core classes of MoMo: non-altering and model altering morphisms [17][18]. As evidenced in Table 1, 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 [19]. Other relations, such as the merge operation, can also be classified as model altering morphisms, however they are not detailed in this paper.

Being more interested in the mapping operation (non-altering) for this paper, these generic function descriptions are not detailed enough to deal with the specificities of the multiple information models used by the enterprise systems of today's business networks. To respond to the constant knowledge and model changes on heterogeneous and dynamic networks, it is required to use a more detailed and traceable mapping format that provides a semantic "link" between two different models and its components. On the following sub-sections, technologies

and formalization methods will be analysed concerning their usability towards that goal.

Table 1. Cases of Model Morphisms

MoMo	Formalization	Classification
Mapping: $\theta(A, B)$	$\forall A, B \in M: \theta(A, B) \subseteq Sub(A) \times Sub(B)$	Non-altering
Transformation: $\tau: A \times \theta \to B$	$\forall A, B \in M: if\ \exists \theta(A, B)\ then\ \tau(A, \theta) = B$	Model altering

3.3 Semantic Mismatches

Mismatches are inconsistencies of information that result from "imperfect" mappings. Due to the differences among models referred before, almost in every case, a MoMo leads to a semantic mismatch, which can either be lossy or lossless depending on the nature of the related model elements (see Table 2): In lossless cases, the relating element can fully capture the semantics of the related; while in lossy mismatches a semantic preserving mapping to the reference model cannot be built [20].

Table 2. Semantic Mismatches (based on [20])

	Mismatch	Description
Lossless	Naming	Different labels for same concept
	Granularity	Same information decomposed (sub)attributes (see Fig. 2)
	Structuring	Different design structures for same information (see Fig. 1)
	SubClass-Attribute	An attribute, with a predefined value set (e.g. enumeration) represented by a subclass hierarchy
	Schema-Instance	An attribute value in one model can be a part of the other's model schema (see Fig. 1)
	Encoding	Different formats of data or units of measure (e.g. kg and lbs)
Lossy	Content	Different content denoted by the same concept
	Coverage	Absence of information
	Precision	Accuracy of information (see Fig. 1)
	Abstraction	Level of specialisation (e.g. "Car" and "Ford")

This notion of mismatch can bring a semantic meaning to the type of the relationship being established in the mapping. However, the envisaged semantic "link" between two different models needs to account for more than inference of a meaning. It needs to be represented through a formal expression that is traceable and parseable by an intelligent system that can deduce and recommend mapping readjustments, which might even change the mismatch type.
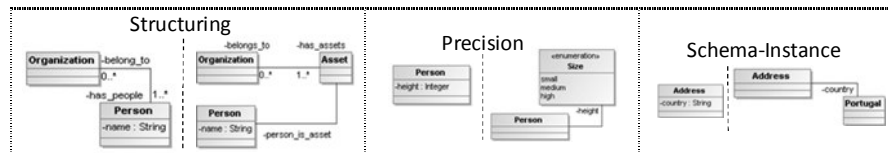

Fig. 1 - Mismatch examples

3.4 MoMo Formalisms

Model Morphisms, as envisaged in section 3.2, are intended to introduce a method of describing relationships/transformations among models. Originally graph theory has been used, but other and theories can be considered to achieve the envisaged goals:

Classical Mathematics: Graph & Set Theory

Graphs are a common way to graphically present models, where the nodes are considered as a domain entity and the edges as relations between them. For the purposes of MoMo, model operations such as the ones of Table 1 can be described using a 6-tuple labelled oriented multigraph (*LDMGraph*) of the form $G=(V,E,s,t,lv,le)$, where: V is the vertex set of G; E is the edge set of G; $s: E \rightarrow V$, is a function that associates an edge with its source vertex; $t: E \rightarrow V$, is a function that associates an edge with its target vertex; $lv: V \rightarrow \sum V$, is a function that associates a vertex with its label; $le: E \rightarrow \sum E$, is a function that associates an edge with its label [17], [21]. This abstract view of models allows formal reasoning on their properties and on the properties of the model operations needed for their effective management.

As graphs, also sets can be used to represent models and operations using first-order logic, algebra and axioms. Being defined as a collection "M" of distinct objects "m", a set can represent objects, numbers, other sets, etc [22]. Operations such as membership "$M1 \subseteq M2$", power "$P(M)$", union "$M1 \cup M2$", intersection "$M1 \cap M2$", complement "$M1 \setminus M2$", or cartesian product "$M1 \times M2$" are already well defined.

Mapping as a model: Model Management [23]

This theory defends that a mapping between models *M1* and *M2* should be a model "*map12*" and two morphisms (one between "*map12*" and *M1* and another between "*map12*" and *M2*).

Fig. 2 - Mapping as a model (*map12*)

Thus, each object "m" in the mapping can relate a set of objects in *M1* to a set of objects in *M2*. In this approach, instead of representing a mapping as a pair of objects, a mapping is represented as a set of objects (see Fig. 2). Using concepts from classical mathematics, this approach enables to define complex algebra to describe major model operations such as match, compose, diff, model gen, or merge.

Mapping as a complex tuple: Matching [24]

The match operator takes two graph-like structures and produces a mapping between the nodes of the graphs that correspond semantically to each other. Mappings between these elements can be described using set-theoretic semantic relations instead of using traditional numeric coefficients. The meaning of concepts (not labels) within a model can determine equivalence "=", more "⊒" and less "⊑" general, as well as disjointness "⊥" relationships. Having this, a mapping element can be defined as a 4 level tuple <*IDij, ai, bj, R*> where: *IDij* is a unique identifier of the given mapping element; *ai* is the i-th node (or vertex) of the first tree; *bj* is the j-

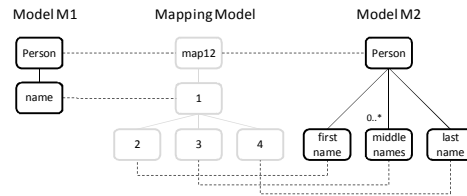th node of the second tree; and R specifies the semantic relation which may hold between them.

The above methodologies seem to be powerful in terms of expressiveness of the morphism. However others exist, such as the composition of complex operations based on a catalogue of primitive transformations [25]. However, this approach is more focused on model altering morphisms.

4   Modelling Morphisms to Enable Sustainable Interoperability

So far, a proven approach to deal with interoperability relies on the usage of dedicated knowledge models and international standards acting as information regulators among organizations. However, due a complexity of reasons many organizations are still focused on P2P relationships, where each one tends to use its own data format and business rules, and handles as many mappings as the number of business partners [9].

Either case, after interoperability is first established and all morphisms defined, the set of organizations within a network demonstrate a period of stability exchanging e-messages following the established mappings [9]. At this stage, the networks display symmetry [26]. However, that might not be sustainable for long if business requirements change. Organizations are managed by people that have different opinions and backgrounds based on several factors such as culture, professional experience, family, etc. They manage, work, and are themselves customers of different organizations, which in turn have different systems that are structured according to several information models implemented on multiple software platforms. All this heterogeneity leads in most cases, the network to experience 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. This is even more evident in multi-domain networks (e.g. collaborative product design) where information is dispersed and frequently replicated in many IS.

To mitigate that, and inspired by CAS, context awareness and traceable morphisms are demanded in support of intelligence. Also, monitoring and decision support systems must be considered in the construction of a framework that implements sustainable interoperability in cooperation networks, thus addressing the robustness of an IS both at conceptual and structure related levels. The sustainable interoperability targeted in this paper is mostly technical with direct impacts on the economical axis of sustainability, but will also produce indirect effects on the social and environmental axis since it enables organizations to redirect money away from technological issues.

4.1 Knowledge Enriched Tuple for Mappings Representation

Observing all previously explained technologies and methodologies for managing morphisms, the authors consider that there is no perfect solution that can provide all the desired goals at once. Some are ideal for structural issues, others for semantics providing good human traceability, while others are more formal and mathematical based. Therefore, we propose the usage of a 5-tuple mapping expression (equation

1), reusing some of the concepts explained in section 3, that formalizes the morphism between two model elements (*a* and *b*) and is enriched with semantic information that enables fast human readability, where $\forall A, B \in M, \exists a \in A$ and $\exists b \in B$: *if M is an LDMGraph then* $a \in V(A)$ *and* $b \in V(B)$.

$$Mapping\ Tuple\ (MapT): <ID, MElems, KMType, MatchClass, Exp> \quad (1)$$

- *ID* is the unique identifier of the MapT and can be directly associated with the *a*'s vertex number: $IDi.j\_x: 1 \leq i \leq |V(A)|$ *and* $1 \leq j \leq |V(Sub(B))|$ *and* $x \in \mathbb{N}$. The depth of the sub-graph detail used in the mapping is not limited, and *x* is a counter for multiple tuples associated with the same concept;

- *MElems* is the pair (a, b) that indicates the mapped elements. If the *ID* specifies a mapping at the n-th depth level of the graph, *a* should be at the same level, i.e. *a.ai* (for *i =1..n*);

- *KMType* stands for Knowledge Mapping Type, and can be classified as: "Conceptual" if mapping concepts and terms; "Semantics" if mapping model schemas; and "InstantiableData" if the mapping is specifying instantiation rules.

Fig. 3 – KMType values

- $KMType = \{Conceptual, Semantics, InstantiableData\}$;

- *MatchClass* stands for Match/Mismatch Classification and depends on *KMType*, such as $\forall (a, b) \in MElems$:

- $\forall KMType$, if a=b, the mapping is absolute and $MatchClass = Equal$;
  - o if $KMType = Conceptual$, the mapping is relating terms/concepts, and $MatchClass \in \begin{Bmatrix} Equal, Naming, Coverage, \\ MoreGeneral, LessGeneral, Disjoint \end{Bmatrix}$ depending on the coverage of the relationship;
  - o Otherwise, the mapping is structural or non-existent and $MatchClass \in$ Table 2$\cup\{Equal, Disjoint\}$;

- *Exp* stands for the mapping expression that translates and further specifies the previous tuple components. It can be written using a finite set of binary operators derived from the mathematical symbols associated with the mapping types and classes (e.g. "$=, \sim, \subseteq, \supseteq, \perp, +, -, \times, \div, concatenate, split$").

This mapping tuple which represents $\theta(a, b)$, can also be used to generate a transformation function τ, where $\tau(a, \theta) = b$, being $(a, b) \in MElems$. Therefore, when used by intelligent systems such as CAS-like IS, the tuple's information enables automatic data transformations and exchange between two organizations working with/on different information models, thus achieving an interoperable state among them and supporting the recovery from any harmonization breaking situation.
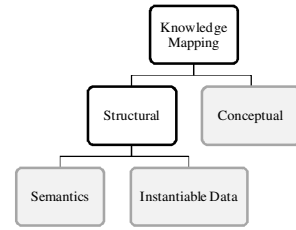
4.2 Communication Mediator (CM)

With the MapT stored in a knowledge base (KB) to support communications intelligence, all information concerning the mappings between models or ontologies of business partners can be accessed by their local systems. This allows communities to build IS with reasoning capabilities able to understand each others' representation format, without having to change their data and communication functions [13].

The proposed CM is defined by an ontology in OWL format. It has been built up as an extension to the Model Traceability Ontology defined in [27], which 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 structure of the evolved communication mediator is presented in Fig. 4 and described as follows.
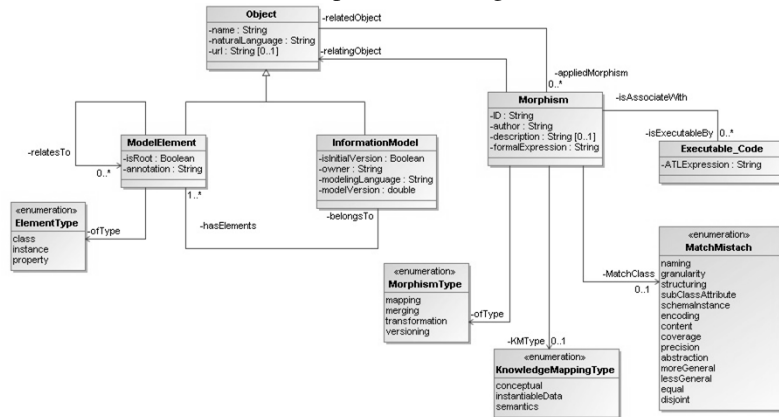

Fig. 4 - Structure of knowledge base for communication support (CM)

The CM has two main classes: *Object* and *Morphism*. The *Object* represents any *InformationModel* (IM) which is the model/ontology itself and *ModelElement*s (also belonging to the IM) that can either be classes, properties or instances. The *Morphism* basically represents the MapT described in the previous section: it associates a pair of *Objects* (related and relating – *Melems* in MapT), and classifies their relationship with a *MorphismType*, *KnowledgeMappingType* (if the morphism is a mapping), and *Match/Mismatch* class (*MatchClass* in MapT). The *Morphism* is also prepared to store transformation oriented *ExecutableCode* that will be written in the Atlas Transformation Language (ATL) and can be used by several organizations to automatically execute the mapping, transforming and exchanging data with their business partners as envisaged in [28].
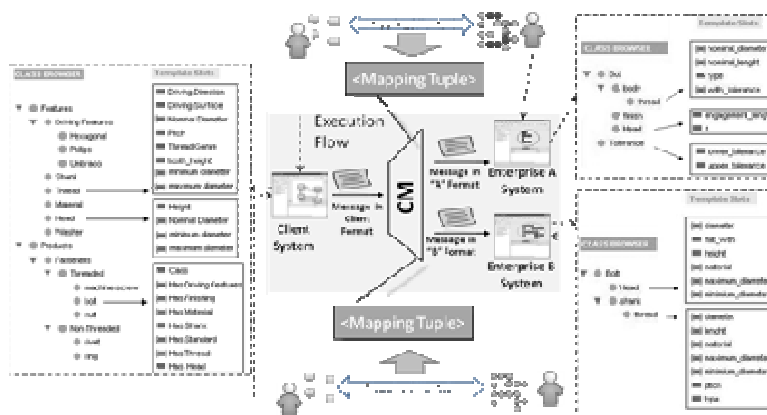

Fig. 5 - Mapping design and execution flow in data exchange

5   Case Study from Mechanical Industry

The simple choice of a "bolt" supplier by a mechanical engineer/designer, very often brings interoperability issues. Suppliers usually define proprietary nomenclatures for their products and its associated knowledge representation (whatever format). Thus, the need to align product data and knowledge emerged as a priority to solve the dilemma. A possible solution is to allow each enterprise involved to keep its terminology and classification in use, and define formal mappings that mediate the communications among them using the tuple and CM here proposed (Fig. 5).

The presented case study is related to a Client System able to represent and act as a retailer of "bolts" which has two different suppliers (Enterprises A and B). Thus, following the previously presented approach it was established a set of mappings between this Client's information model and the others from the suppliers. Such mappings were defined following the MapT and were then recorded in the CM. In the left and right parts of the Fig. 5 it is illustrated the Client, Enterprise A and Enterprise B information structures (in this case, ontologies). Using them it was possible to specify the set of tuples that define a traceable mapping at the information model level (level n+1), and therefore provide the support to establish/re-establish interoperability (i.e. sustain). Indeed, following the model-driven development paradigm, data transformations (level n) can be executed automatically and repeated whenever desired/demanded with little costs to the overall network interoperability (Fig. 6) [28][29].
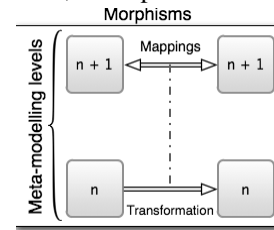

Fig. 6 - Model Driven Framework (from [28])

5.1   Case Study Tuples for Sustainable Interoperability

Taking specifically the example related to the tolerance characteristic of a bolt it is stated that in the client system such tolerance characteristic is defined by two properties, *maximum diameter* and *minimum diameter*. These are equally used by the Enterprise B. However, Enterprise A, uses the concepts *upper tolerance* and *lower tolerance*, which represents the same expected result but using different data values. *Nominal diameter* and *diameter* concepts have the same value and semantics in all the ontologies. Thus, the transformations equations related to the tolerance properties from Client to Enterprise A, are the following:

$$maximum\ diameter\ =\ nominal\ diameter\ +\ upper\ tolerance \qquad (\mathbf{2})$$
$$minimum\ diameter\ =\ nominal\ diameter\ -\ lower\ tolerance \qquad (\mathbf{3})$$

Due to space constraints, the authors decided to include only 3 tuples: one for each *KMType*. Taking the mapping example that relates the conceptual level of the *maximum diameter* and *upper tolerance* concepts respectively from Client and Enterprise A systems, the resulting tuple is specified as in Table 3 (upper section). As an example of MapT definition related to the structure with *KMType = Semantics*, it was chosen the relationship between a Client's bolt and Enterprise B's Bolt (Table 3 - middle section). They use different structures for the same concept, but the mapping is not absolute (e.g. B does not have driving features information). Finally, as an example of MapT *KMType = InstantiableData*, it

was chosen the equation 2. In this case it is needed to define an operation using two data concepts from the related model: the *upper tolerance* and *nominal diameter.*

Table 3. Case Study MapT's

ID		$Client2.1.1.2.7.8\_1$
$MElems$	a	$((Thread)Products.Fasteners.Threaded.bolt.HasThread).maximum\ diameter$
$= (a, b)$	b	$((Tolerance)Bolt.body.thread.with\_tolerance).upper\_tolerance$
$KMType$		$Conceptual$
$MatchClass$		$MoreGeneral$
Exp		$"a \supseteq b"$
ID		$Client2.1.1.2\_1$
$MElems$	a	$Products.Fasteners.Threaded.bolt$
$= (a, b)$	b	$Bolt$
$KMType$		$Semantics$
$MatchClass$		$Coverage$
Exp		$"a \sim b"$
ID		$Client2.1.1.2.7.8\_2$
$MElems$	a	$((Thread)Products.Fasteners.Threaded.bolt.HasThread).maximum\ diameter$
$= (a, b)$	b	$Bolt.body.thread.nominal\_diameter$
$KMType$		$InstantiableData$
$MatchClass$		$Granularity$
Exp		$"a = b + ((Tolerance)Bolt.body.thread.with\_tolerance).upper\_tolerance"$

With the above mapping tuples specified, a simple example can be drawn to illustrate how the network could sustain its interoperability status: if for some reason Enterprise A started using English metric units (inches instead of millimetres), an *Encoding* mismatch would be detected and all mappings reused just by adapting the *InstantiableData* tuples according to the proper conversion rule. For example, the expression (Exp) of the last MapT would change to a= $(b + ((Tolerance)Bolt.body.thread.with\_tolerance).upper\_tolerance) * 25,4$, and the information in the CM updated automatically to enable a swift recovery of the interoperable status using the automatic code generation of model-driven capabilities. In the case of mappings not represented using the MapT format or in any other traceable form, enterprises would need to go through the full integration process again, reprogramming communication functions manually at huge costs.

6   Concluding Remarks and Future Work

The net effect of cheap communications is a perception that individuals and organizations have to deal in a world that is increasingly dynamical, complex and uncertain, and that their actions may have unintended consequences that impact on others [30]. Indeed, business networks are plagued with uncertainty, and systems interoperability has become an important topic of research in the last years not only from the perspective of establishing it, but also of sustaining it.

The development of standards and ontologies helps to master the communication within those networks. However, alone, information standards do not solve today's the enterprise interoperability problems. Indeed, typically each stakeholder has its own business requirements and suffers external influences that might lead to a harmonization breaking phenomena [9]. Therefore, organizations from similar business environments are still having trouble cooperating at a long term.

To address this issue and support sustainable interoperability in the context of collaboration networks, it is required to analyse how intervention strategies on the network evolution, namely attempts to shape local interaction patterns and mappings, affect the network interoperability sustainability. The authors use a knowledge enriched tuple for mappings representation that is stored on a communication mediator that keeps 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.

As for part of the future work is the validation of the CM for other types of morphisms, such as composition of models, merging operations, etc. For model altering morphisms, also the proposed tuple might need some readjustments, or even assume a different format. The major complexity associated to the study of the properties of complex systems is that the associated models, drive to non-linearity, which in turn, drives to difficulties in the system's study and in predicting their behaviour. In this context and also part of future work, at the system microscopic level, prediction could be seen as a proposal for the automatic adaptation of the network morphisms. Thus, the CM envisages to have associated learning capabilities, monitoring, diagnostic and prognostic services based on the operations history and interventions on the involved systems.

References

1. Friedman, T.: The World is Flat. Farrar, Straus & Giroux (2005)
2. Wilkinson, I., Young, L.: On cooperating: firms, relations and networks. Journal of Business Research. Vol.55, Issue 2, pp.123--132 (2002)
3. Amin A., Cohendet P.: Architectures of knowledge: firms, capabilities, and communities. Oxford University Press (2004)
4. Camarinha-Matos L., Afsarmanesh H.: Collaborative networked oganizations: a research agenda for emerging business models. Springer (2004)
5. Jardim-Goncalves, R., Agostinho, C., Malo, P., Steiger-Garcao, A.: Harmonising technologies in conceptual models representation. International Journal of Product Lifecycle Management. Vol.2, No. 2, pp. 187—205 (2007)
6. Peppard, J., Rylander, A.: From Value Chain to Value Network: Insights for Mobile Operators. European Management Journal. Vol.24, Issues 2-3, pp. 128—141 (2006)
7. Ray, S.R., Jones, A.T.: Manufacturing interoperability. Journal of Intelligent Manufacturing. Vol. 17, No.6, pp. 681—688 (2006)
8. White, W.J.; O'Connor, A.C.; Rowe, B.R.; 2004. Economic Impact of Inadequate Infrastructure for Supply Chain Integration. NIST Planning Report 04-2. Gaithersburg.
9. Agostinho, C.; Jardim-Goncalves, R.: Dynamic Business Networks: A Headache for Sustainable Systems Interoperability. In 4th Int. Workshop on Enterprise Integration, Interoperability and Networking (EI2N'2009) of the OTM Conference, November (2009)
10. R Axelrod; The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration (1997); Princeton University Press; ISBN 0-6910-1567-8

11. J Gharajedaghi; Systems Thinking: Managing Chaos and Complexity: A Platform for Designing Business Architecture (2005); Butterworth-Heinemann; ISBN 0-7506-7973-5

12. Sugihara G, May R.M.; Nonlinear Forecasting as a Way of Distinguishing Chaos from Measurement Error in Time Series (1990); NATURE 344 : 734

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

14. Lubell, J., Peak, R. S., Srinivasan, V, and Waterbury S. C. (2004), 'STEP, XML, AND UML: COMPLEMENTARY TECHNOLOGIES', Proc. of DETC 2004, ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, Utah USA, Sep 28- Oct 2, 2004.

15. Guarino, N. and Schneider, L. (2002) Ontology-driven conceptual modeling. Lecture Notes In Computer Science Vol. 2348.

16. Agostinho, C. and Delgado, M and Steiger-Garcao, A. and Jardim-Goncalves, R (2006) Enabling Adoption of STEP Standards Through the Use of Popular Technologies. In: 13th ISPE International Conference on Concurrent Engineering (CE 2006), 18-22 Sep 2006

17. INTEROP NoE. Deliverable MoMo.2 - TG MoMo Roadmap. InterOP, 2005

18. Agostinho C, Sarraipa J, D'Antonio F, et al. Enhancing STEP-based interoperabity using model morphisms. In: 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007)

19. Delgado M., Agostinho C., Malo P., Jardim-Gonçalves R., (2006) A framework for STEP-based harmonization of conceptual models. 3rd International IEEE Conference on Intelligent Systems (IEEE-IS'06), Wetsminster, Jul 4-6 , 2006.

20. INTEROP NoE. Deliverable D.A3.3 - Semantic Annotation language and tool for Information and Business Processes. InterOP, 2006

21. Delgado, Marco (2008). Harmonisation of STEP and MDA conceptual models using Model Morphisms. MSc thesis presented at Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2008

22. Dauben, Joseph W., Georg Cantor (1979). His Mathematics and Philosophy of the Infinite. Boston: Harvard University Press, ISBN 978-0-691-02447-9.

23. Philip A. Bernstein, Applying Model Management to Classical Meta Data Problems, First Biennial Conference on Innovative Data Systems Research, California, January 2003.

24. Giunchiglia, F. Yatskevich, M. Shvaiko, P. Semantic Matching: Algorithms and Implementation. J. Data Semantics 9: 1-38 (2007)

25. Blaha, M.; Premerlani, W. (1996). A catalog of object model transformations. Proceedings of WCRE, Volume 96, pp 87, 1996.

26. Nicolis, G., Prigogine I.: Exploring Complexity: An Introduction. W. H. Freeman and Company, New York (1989)

27. Sarraipa, J.; Zouggar, N.; Chen, D; Jardim-Goncalves, R. (2007). Annotation for Enterprise Information Management Traceability. In Proceedings of IDETC/CIE ASME 2007

28. Agostinho, C. Correia, F. and Jardim-Goncalves, R. (2010) Accepted In: 17th ISPE International Conference on Concurrent Engineering (CE 2010). Sep 6-10, Cracow, Poland

29. Selic B. The Pragmatics of Model-Driven Development. In: September/October IEEE Software Magazine 2003

30. Merali, Y., McKelvey, B.: Using Complexity Science to effect a paradigm shift in Information Systems for the 21st century. Journal of Information Technology. Vol.21, No.4, pp. 211--215 (2006)