Carlos Manuel Melo Agostinho

Mestre em Engenharia Informática

Sustainability of Systems Interoperability in Dynamic Business Networks

Dissertação para obtenção do Grau de Doutor em Engenharia Electrotécnica e de Computadores

Orientador: Prof. Doutor Ricardo Luís Rosa Jardim Gonçalves, Professor Associado com Agregação da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor Luís Manuel Camarinha de Matos
Arguente(s): Prof. Doutor Hervé Panetto
Prof. Doutor Manuel Martins Barata

Vogais: Prof. Doutor Adolfo Sanchez Steiger Garção
Prof. Doutor António Carlos Bárbara Grilo
Prof. Doutor Aurelian Mihai Stanescu
Prof. Doutor João Pedro Mendonça de Assunção da Silva
Prof. Doutor Ricardo Chalmeta Rosaleñ

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Outubro, 2012

Copyright

Sustainability of Systems Interoperability in Dynamic Business Networks

To Maria,
my wife and inspiration…
without your support this would not be possible

To my parents,
my roots

For the future,
why else would I be doing this?

Acknowledgements

During the slow and often intermittent evolution of this work I had the chance to meet and work with people that have inspired and supported me during this journey. Many people think research is a solitary activity. Nevertheless that could not be furthest away from truth. I owe a great deal to colleagues, friends and family who have helped extend my commitment, and who, through their own research, comments and questions have encouraged, supported and enlightened me. I don't know if I will have the chance of writing anything similar, so I want to thank everybody ☺

I would like to start thanking my friend and supervisor Ricardo Gonçalves for the opportunity he gave me, more than ten years ago, to work with him and join his research team. Along the years he has always demonstrated his support and encouraged me find my own path and working methods. His direction, suggestions, and remarks during the development of the thesis have been very useful to the successful accomplishment of this work. Thank you for telling me when to stop.

My gratitude is also to my colleagues from GRIS, at UNINOVA. Without them, especially João Sarraipa, this task would be even harder. Fernando Ferreira, Maria João Lopes, and most recently Edgar Silva, thanks for your support and companionship every day. A special thank goes also for the master students to whom I collaborated or had the pleasure of co-supervising, namely Filipe Correia, Tiago Gaspar, Miguel Beça, and José Ferreira.

Professor Adolfo Steiger Garção, thank you for your knowledgeable advices.

My recognition to the participants of the international projects I was involved in, and whose results contributed to the work here presented, namely CRESCENDO (FP7-234344), ENSEMBLE (FP7-257548), MSEE (FP7 284860), and IMAGINE (FP7 285132). A special mention to my international colleagues, partners and friends: Amparo Roca de Togores, Maria José Nuñez, Daniel Pina, Hervé Panetto and Fenareti Lampathaki. They know the support they have given me.

I must not forget all my family who, since the beginning, have encouraged and gave me strength in the most difficult times. My parents for the efforts they did to provide me a good education, and for the perseverance values that I follow on a daily basis. To my uncle Henrique, the older "brother" who was present whenever I needed. To my grandmother, Etelvina, for the role taken throughout my childhood and for the tranquility she passes to the ones surrounding her. For the ones that have gone but still keep watching for me; and finally to the joy and enthusiasm of the younger, Beatriz, Artur, and Afonso. They keep reminding me life is about enjoying and having fun.

I wish to acknowledge also my friends that, directly and indirectly, supported and assisted me. Sometimes a dinner and night out among friends can solve a great deal of problems.

Last, but most importantly, my words of recognition go to Maria, my wife, my rock, and my source of inspiration. Thank you for being there in days everything goes wrong and for sharing the successes at my side. May our love endure and that all our dreams come true!

Agradecimentos

Em português, gostaria de repetir os agradecimentos à minha família que, desde o início, me tem incentivado e apoiado nos momentos mais difíceis.

Aos meus pais, pelo esforço levado a cabo durante todos estes anos, proporcionando-me uma boa educação, e transmitindo-me os valores de perseverança que sigo diariamente.

Ao meu tio Henrique, o “irmão” mais velho que sempre esteve presente nos momentos importantes. À minha avó, Etelvina, pelo papel activo ao longo da minha infância e pela paz transmitida aqueles os que a rodeiam. Aos que já não se encontram entre nós mas que continuam a olhar por mim; e finalmente à alegria e entusiasmo dos mais novos, Beatriz, Artur e Afonso. Relembram-me frequentemente que a vida consiste em alegria e que tem que ser disfrutada.

Por último, mas mais importante, o meu obrigado à Maria, minha esposa, meu porto de abrigo, a minha fonte de inspiração. Obrigado por lá estares nos dias maus e por partilhares os sucessos a meu lado. Que o nosso amor perdure e que todos os nossos sonhos se concretizem!

Abstract

Collaborative networked environments emerged with the spread of the internet, contributing to overcome past communication barriers, and identifying interoperability as an essential property to support businesses development. When achieved seamlessly, efficiency is increased in the entire product life cycle support. However, due to the different sources of knowledge, models and semantics, enterprise organisations are experiencing difficulties exchanging critical information, even when they operate in the same business environments. To solve this issue, most of them try to attain interoperability by establishing peer-to-peer mappings with different business partners, or use neutral data and product standards as the core for information sharing, in optimized networks.

In current industrial practice, the model mappings that regulate enterprise communications are only defined once, and most of them are hardcoded in the information systems. This solution has been effective and sufficient for static environments, where enterprise and product models are valid for decades. However, more and more enterprise systems are becoming dynamic, adapting and looking forward to meet further requirements; a trend that is causing new interoperability disturbances and efficiency reduction on existing partnerships.

Enterprise Interoperability (EI) is a well established area of applied research, studying these problems, and proposing novel approaches and solutions. This PhD work contributes to that research considering enterprises as complex and adaptive systems, swayed to factors that are making interoperability difficult to sustain over time. The analysis of complexity as a neighbouring scientific domain, in which features of interoperability can be identified and evaluated as a benchmark for developing a new foundation of EI, is here proposed. This approach envisages at drawing concepts from complexity science to analyse dynamic enterprise networks and proposes a framework for sustaining systems interoperability, enabling different organisations to evolve at their own pace, answering the upcoming requirements but minimizing the negative impact these changes can have on their business environment.

Keywords: Enterprise Systems Interoperability, Complex-Adaptive Systems, Model-driven Interoperability, Model Morphisms, Self-organisation, Monitoring

Resumo

Os ambientes colaborativos em rede têm proliferado com o crescimento da Internet, contribuindo para superar barreiras de comunicação, e identificando a interoperabilidade como uma propriedade essencial ao desenvolvimento das empresas. Quando alcançada de forma consistente, o suporte ao ciclo de vida do produto torna-se mais eficiente. No entanto, devido às diversas fontes de conhecimento, modelos e semântica, e apesar de actuarem nos mesmos ambientes de negócio, as organizações empresariais têm vindo a experimentar dificuldades na partilha de informação vital. Para resolver este problema, a maioria delas tenta alcançar a interoperabilidade através da criação de mapeamentos ponto-a-ponto com os mais variados parceiros de negócios, ou em redes optimizadas, usam normas de dados e de representação de produto, como formatos neutros e nucleares à partilha de informação.

Em ambiente industrial a prática dita que os mapeamentos entre modelos, que regulam as comunicações entre diferentes empresas, são normalmente definidos uma única vez e embutidos directamente no código dos sistemas de informação. Esta solução tem sido eficaz e suficiente em ambientes estáticos, onde os modelos empresariais e de produto são válidos durante décadas. No entanto, cada vez mais, os sistemas empresariais são dinâmicos, procurando adaptar-se e responder a novos requisitos; uma tendência que está a causar novas perturbações em relações interoperáveis e a reduzir a eficiência em parcerias existentes.

A interoperabilidade entre empresas (IE) é uma área de investigação aplicada bem cimentada, que estuda o tipo de problemas acima descritos, e propõe novas abordagens e soluções para a sua resolução. Este trabalho de doutoramento contribui para essa investigação considerando as empresas como sistemas complexos e adaptativos, influenciados por factores que estão a dificultar a manutenção da interoperabilidade ao longo do tempo. É aqui proposta a análise da complexidade como uma área científica contígua, na qual características de interoperabilidade podem ser identificadas e avaliadas como uma referência para o desenvolvimento de uma nova fundação da IE. Esta aproximação prevê a utilização de conceitos provenientes da ciência da complexidade para analisar redes empresariais dinâmicas, e propõe uma estrutura de suporte à sustentabilidade da interoperabilidade entre sistemas, permitindo que diferentes organizações evoluam ao seu próprio ritmo, respondendo a futuros requisitos, mas minimizando o impacto negativo que essas mudanças podem causar no seu próprio ambiente de negócios.

Palavras Chave: Interoperabilidade de Sistemas Empresariais, Sistemas Complexos e Adaptativos, Interoperabilidade Orientada por Modelos, Morfismos de Modelos, Auto-organização, Monitorização

Table of Contents

Table of Figures

List of Tables

List of Acronyms and Abbreviations

***aaS**	Everything as a Service
ABM	Agent-Based Modelling
ACL	Agent Communication Language
AI	Artificial Intelligence
AIF	ATHENA Interoperability Framework
ANN	Artificial Neural Networks
AP	Application Protocol
ARIS	Architecture of Integrated Information Systems
ATL	Atlas Transformation Language
b2b	Business-to-Business
b2c	Business-to Consumer
BCIM	Bottom CIM Level
BDA	Behavioural Digital Aircraft
C4IF	Connection, Communication, Consolidation, Collaboration Interoperability Framework
CAD	Computer-Aided Design
CAS	Complex Adaptive System
CAS-SIF	CAS-Based Framework to Support Sustainable Interoperability
CIM	Computation Independent Model
CIMOSA	Computer Integrated Manufacturing Open System Architecture
CM	Communication Mediator
CPD	Collaborative Product Development
CPSI	Conference Proceedings Citation Index
CRESCENDO	Collaborative and Robust Engineering using Simulation Capability Enabling Next Design Optimisation
DMN	Dispersed Manufacturing Networks
DoDAF	Department of Defense Architecture Framework
EA	Enterprise Architecture
EC	European Commission
EEP	Eurostep EXPRESS Parser
EI	Enterprise Interoperability
EIRR	Enterprise Interoperability Research Roadmap
EIF	European Interoperability Framework
EISB	Enterprise Interoperability Science Base
EMF	Eclipse Modelling Framework
ENSEMBLE	Envisioning, Supporting and Promoting Future Internet Enterprise Systems Research through Scientific Collaboration
ER	Entity-Relationship
EPS	European Public Services
FI	Future Internet

FInES	Future Internet Enterprise Systems
FIPA	Foundation for Intelligent Physical Agents
GIM	GRAI Integrated Methodology
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMAGINE	Innovative End-to-end Management of Dynamic Manufacturing Networks
Interop-Vlab	International Virtual Laboratory for Enterprise Interoperability
IoS	Internet of Services
IPyM	Interoperability Efficiency Pyramid Model
IS	Information System
ISO	International Organisation for Standardization
ISU	Interoperability Service Utility
JADE	JAVA Agent Development Framework
JAXB	Java Architecture for XML Binding
KQML	Knowledge Query Manipulation Language
LCIM	Levels of Conceptual Interoperability Model
LIM	Language Independent Model
LIMM	Language Independent Meta-Model
LISI	Levels of Information System Interoperability
MapT	Knowledge Enriched Tuple for Mappings Representation
MAS	Multi-Agent System
MBE	Model Based Enterprise
MBSE	Model-Based Systems Engineering
MDA	Model-Driven Architecture
MDD	Model-Driven Development
MDE	Model-Driven Engineering
MDI	Model-Driven Interoperability
MIRAI	Monitoring as a Support of Sustainable Interoperability of Enterprise Systems
MOF	Meta Object Facility
MoMo	Model Morphisms
MRS	MoMo Recommendation System
MS	Model Store
MSEE	Manufacturing SErvice Ecosystem
NC3TA/RMI	NATO C3 Technical Architecture Reference Model for Interoperability
NIST	National Institute of Standards & Technology
OCL	Object Constraint Language
OIM	Organisational Interoperability Maturity

OMG	Object Management Group
OWL	Web Ontology Language
P2P	Peer-to-Peer
PaaS	Platform as a Service
PDMA	Product Development and Management Association
PERA	Purdue Enterprise Reference Architecture
PIM	Platform Independent Model
PLC	Product Life Cycle
POC	Proof-of-Concept
PSM	Platform Specific Model
QVT	Query View Transformation
R&D	Research and Development
SaaS	Software as a Service
SC	Supply Chain
SCI	Science Citation Index
SME	Small and Medium sized Enterprise
SOA	Service Oriented Architectures
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SQuaRE	Software product Quality Requirements and Evaluation
STEP	STandard for the Exchange of Product model data
SUS	System Under Study
SUT	System Under Test
TC	Test-Case
TCIM	Top CIM Level
TOGAF	The Open Group Architecture Framework
TTCN	Tree and Tabular Combined Notation
UDDI	Universal Description, Discovery, and Integration
UEML	Unified Enterprise Modelling Language
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WS	Web Services
WSDL	Web Service Definition Language
WWW	World Wide Web
XMI	XML Metadata Interchange
XML	eXtensible Markup Language

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 organisational levels, as well as conducting tendencies at global market. As Friedman (2005) claims in his book, "The World is Flat", with the explosion of advanced Web technologies, knowledge pools and resources became available all over the planet, levelling the market as never before and reducing gaps among organisations. Thus, nowadays, it is the way knowledge is used that makes the difference.

With markets becoming increasingly complex and dynamic, the traditional way of doing business does not provide the expected leverage. Currently, companies do not survive and prosper solely through their own individual efforts and isolated knowledge. In fact, each one's success also depends on the performance of others to whom they do business with, and hence on the nature and quality of knowledge sharing (Wilkinson & Young, 2002)*. This involves a mix of both cooperative and competitive elements, and to cope with them, organisations need to focus on their core competencies by improving relationships with customers, streamlining supply chains, and collaborating with partners to create valued networks between buyers, vendors and suppliers* (Amin & Cohendet, 2004; Camarinha-Matos & Afsarmanesh, 2004; Jardim-Goncalves, Agostinho, et al., 2007)*.*

1.1. Research Framework and Motivations

This thesis recognises interoperability as an essential property within these collaborative and competitive environments. Without interoperation, companies cannot communicate and share knowledge effectively and collaborations cannot be exploited as they should, thus diminishing

business potentialities. The research developed envisages to address interoperability issues not only at the time of the creation of the valued network, which by itself poses many challenges, but also at the network's life cycle, sustaining its interoperable status, coping with the disturbances and uncertainties that characterise the current business scenarios.

The following sub-sections try to scope the work and pinpoint the major motivations in terms of industrial importance, economical impact, answering to identified research challenges.

1.1.1. Enterprise Collaboration

Work processes all over the world usually involve some combination of types of "work" that can be classified as a continuum (Beck, 2005). The entirely independent effort of an individual/organisation to produce some kind of product or service is found at one end of this continuum. The dependent work where one uses someone else's product or service is next, followed by collaboration and information sharing among people and organisations crossing multiple disciplines in order to achieve a mutual objective. At this level no single person or entity has sufficient knowledge to accomplish a task on their own. On the other end of the continuum is the ubiquitous vision of the full integrated world where partners or systems are tightly coupled, therefore interdependent and inseparable (D. Chen & Doumeingts, 2004).

According to Schrage (1990), collaboration *"…is the process of shared creation: two or more individuals with complementary skills interacting to create a shared understanding that none had previously possessed or could have come to on their own. Collaboration creates a shared meaning about a process, a product, or an event. (...) It can occur by mail, over the phone lines, and in person."* Being a definition from 1990, I believe that it can be updated to include the internet and systems themselves as means to collaborate. This way, the concept of collaboration is of high interest both in the business-to-business (b2b) and business-to consumer (b2c) industry (Mathew, 2002). In the latter, customer service manufacturers and providers may try to develop their collaborative skills to better reach out to their customers and to maintain constant contact with them. In b2b, collaborative product development (CPD) processes or value chain interactions, involving people, processes and technologies across multiple organisations working in the same line of business is becoming the industry standard in light of globalization and outsourcing.

Therefore, the collaborative process may be described as a coordinated and synchronous activity characterized by reciprocal interactions at high frequency that normally require the transfer of rich information among several organisations, i.e. knowledge sharing. However, not every activity requires the same intensity of interaction between the partners and due to cost-benefit considerations they have to be based on the requirements of existing situations (Kern & Kersten, 2007). It is possible to subdivide interactions in four categories (CRESCENDO Partners, 2009b):

- <u>Independence</u>: Where there is no particular need for integration. In this case, organisations do not need to interact nor share any kind of data or information. The activities are driven uniquely by the local company objectives;

- Communication: This is at the basis of any collaborative task. It refers to the process of transferring information from one place to another and simply refers to the act of exchanging information by the use of some kind of media (e.g. verbally, paper, e-mail, etc.);
- Coordination, Cooperation and Collaboration (Kern & Kersten, 2007): Refers to process of building collective knowledge to pursuit a common goal. From an IT perspective, these concepts apply to either autonomous, independent, federated or loosely coupled systems connected together via a communication network. This category points mainly to the virtual enterprise, multi-OEM, heterogeneous organisations in which there is limited and varying alignment between local and global objectives;
- Integration: "Tightly coupled" systems and networks where components are interdependent and cannot be separated. This category points to a homogeneous organisation/network, characterized by the interdependency of their own components and by the standardisation of languages, methods and tools. It refers to the end of the continuum with the ubiquitous vision of the fully integrated world without communication issues.

The research framework for this thesis work is focused on the third category applied to enterprise systems. Being fundamental for the actual global market, organisations can only reach the full collaboration potential if partners develop enhanced capabilities to communicate, coordinate, cooperate, collaborate, and most importantly **interoperate** despite of different organisational structures, technologies or processes (INTEROP Partners, 2006c).

1.1.2. Motivations for Systems Interoperability

Schrage (1990), emphasizes that the issue in collaboration *"isn't communication or teamwork, it is the creation of value"*. Following this, it is possible to conclude that he was looking at collaboration from a perspective of the desired result. Nonetheless, it is in the processes behind that final goal (i.e. the product manufactured, sold, etc.) that interoperability becomes an issue.

In today's networked economy, strategic business partnerships and outsourcing have become dominant business paradigms evidencing a tremendous increase in trade and investments between nations. According to Friedman (2005), the world is becoming a *"tiny flat place"* with information exchanged and applied innovatively across continents, independently of races, cultures, languages or systems; where mass-customization has become a major business hub replacing mass-productions; and with trends changing businesses from technology and product-driven to market and customer-driven thus increasing trade and information exchange, as well as the need for interoperability (Gunasekaran & Ngai, 2005; Pine & Gilmore, 1999).

This evolution has provided consumers a fundamental role on supply chains and on product design. Reliability and rapid delivery of defect-free products to customers is no longer seen as a competitive advantage, but as a requirement (Mentzer et al., 2001; Panetto et al., 2006). A single company cannot satisfy all customers' requirements, and today the war is waged between networks of interconnected organisations (Peppard & Rylander, 2006). Therefore, to succeed in this collaborative but at the same time competitive environment, enterprise systems and

applications need to be interoperable, sharing technical and business information seamlessly within and across organisations, and must be adaptable to different business network environments at all product life cycle (PLC) phases (Jardim-Goncalves et al. 2007; Ray & Jones 2006).

In this sense, being defined as the ability that two or more systems have to exchange information and use it accurately (IEEE, 1990; Software Engineering Institute, 2002), interoperability, more precisely the lack of it, could disturb the creation of new markets, networks, and can diminish innovation and competitiveness of business groups. If multiple systems are being used to manage different portions of a business network (i.e. nodes), several types of costs will be incurred unless those systems have been designed to interoperate. Moreover, if systems are only partially interoperable (IEEE, 1990), translation or data re-entry is required to assure the efficiency of information flows, e.g. in supply chains 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 disturbing the performance of the entire business network (W. J. White et al., 2004).

Nevertheless, apart from being a technical issue, interoperability challenges also appear in the enterprises at organisational and semantic level, underlying the need for patterns and solutions that support the seamless cooperation among ICT systems, information and knowledge, organisational structures and people (Jardim-Goncalves, Grilo, et al., 2006).

Nowadays, an enterprise's competitiveness is largely determined by its ability to seamlessly interoperate with others (Enterprise Interoperability Cluster, 2008). Indeed, the lack of interoperability has been identified in several industrial sectors has a major cost, blocking the achievement of the time-to-market demanded by today's competitive environment (Ray 2002):

- In the year 1999, NIST[1] published a report on the analysis of interoperability costs in the U.S. automotive supply chain, where the estimated annual costs were calculated at about $1.05 billion per year (Brunnermeier & Martin, 1999). These values have been reinforced on a later study (in 2004 also by NIST), which estimates that the total cost of managing supplier-customer inventory and schedule information on a supply chain exceeds $5 billion per year in the automotive industry, and almost $4 billion in the electronics sector (W. J. White et al., 2004).
- Equally dramatic costs are evidenced in the aeronautics industry as well. In 2006, Airbus® assumed that interoperability problems with the design software used in the multiple factories involved in the A380 plane manufacturing, were the cause for a 2 year delay and a $6 billion slippage (Matlack, 2006). Since then, investment on interoperability and more efficient production and design have been a major driver, e.g., only with the CRESCENDO European research project, the industry is expecting to create an impact of several billions of Euros (CRESCENDO IP, 2009).

---

[1] NIST - National Institute of Standards & Technology (www.nist.gov/index.html)

Still, interoperability is not only a concern for large companies. As them, also small and medium sized enterprises (SMEs) do data exchange by implementing business interfaces manually or buying expensive interoperable software solutions, which must suffer complex adaptations in order to meet their own business needs. At the same time, SMEs are, today, confronted with the same level of complexity regarding interoperability as their bigger partners, since they are forced to support similar business interaction patterns. The difference is that they have less human and financial capital to invest in (INTEROP Partners, 2006b).

Within this scope, the funStep initiative (FunStep, 1998) as estimated that the furniture sector (which is SME lead) could benefit largely if interoperability was achieved through a massive adoption of the ISO 10303-236 standard for product data exchange (ISO TC184/SC4, 2006). At the time of the study (2008), it was measured that if the standard implementation followed the expected parameters, the sector could increase the annual benefits in 5% to €6 million by 2010, with a possibility of raising that number to 54% (€70 million) by 2018 and €131 million, 10 years later (INNOVAFUN Partners, 2008). Other references evidencing the value proposition of interoperability are available also for the building and construction sectors, space, and public services (Gallaher et al., 2004; Grilo & Jardim-Goncalves, 2010; Kempler et al., 2009; The United States conference of mayors, 2004).

In the above studies, the costs of poor interoperability show themselves in many ways, some direct and some indirectly, which could be reduced using adequate solutions. Typical areas for incurring cost include (Brunnermeier & Martin, 1999; INNOVAFUN Partners, 2008) :

- Avoidance costs: associated with preventing interoperability issues before they occur (e.g. the cost of developing translation software, maintaining legacy systems, or buying redundant software to enable business transactions with more partners).
- Mitigation costs: are normally the highest interoperability cost type (almost 90% in some of the above studies), and include the resources required to address interoperability problems after they have occurred, such as manually processing or re-entering data.
- Delay costs: arise from interoperability problems that cause delay in the introduction of a new product, or prolong the sale of bespoke products. These costs include: specification costs such as the cost to re-design a product according to miss-interpreted customer requirements; loss of market share, where customers turn to alternative suppliers for a faster response; post-manufacturing interoperability costs as the marketing and sale of a product; etc.
- Future proofing costs: are generally unknown costs that will be faced at some time in the future in order to integrate with new (currently unknown) system requirements.

Mitigation and delay costs could be eliminated if organisations implement “true” interoperability, where each piece of information needed by a network participant is entered only once. Subsequent use and exchange of that information would be managed automatically through software programs, without the need of manual intervention or translation (Enterprise Interoperability Cluster, 2008; W. J. White et al., 2004). This way, it is in avoidance that research and investment should be

encouraged so that the other costs could be reduced and communication across frontiers, among facilities within and between organisations is accomplished seamlessly. Reinforcing this view, the Enterprise Interoperability (EI) Research Roadmap (EIRR) introduces the Interoperability Service Utility (ISU) concept as a grand challenge to obtain a *"utility-like capability that enterprises can invoke on the fly in support of their business activities"*, with specific IT functions being delivered as services that are cheap, fast, reliable, and without major integration efforts (see Figure 1.1). In summary, effective interoperability would support the Single Market[2] and its associated *"four freedoms of movement of people, capital, goods and services"* (Comptia, 2004).

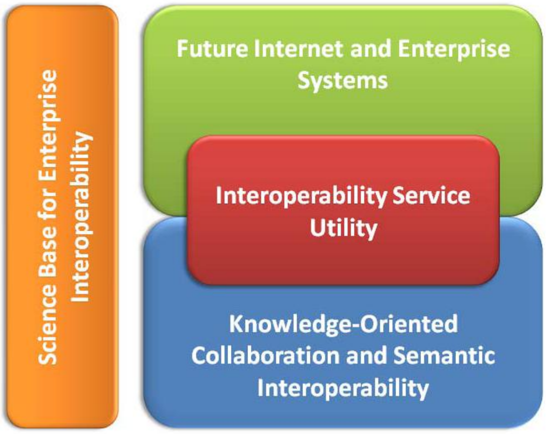


Figure 1.1: EI Grand Challenges (Enterprise Interoperability Cluster, 2008)

Considering the future proofing costs, they are an incognita as they are part of the future. However, to avoid them from becoming measurable in the other cost categories, implementors of the ISU should consider its **sustainability** over time, including the capacity to deal with market dynamicity and technology evolution. In fact, the 2010 version of the EIRR reinforces the Enterprise Interoperability Science Base (EISB) grand challenge as a research challenge (RC8) with similar concerns. It encourages researchers to formulate the scientific foundations to the Future Internet Enterprise Systems (FInES) activities, which would avoid the development of too many technology-driven solutions that tend to be static and not support changes. Past experiences show that solutions tightly connected to a specific technology are difficult to be updated when the latter is outperformed by a new one. Thus it is expected that applications within the scope of ISU are founded on solid scientific basis and have clearly defined properties and foreseeable behaviour, supported by proper metrics and measurement techniques (FInES Research Roadmap Task Force, 2010).

1.1.3. The Problem of Sustainability of Interoperable Solutions

Although specific attempts for answering key issues in the ISU domain should be encouraged, the Grand Challenges of Figure 1.1 are expected to be interrelated when trying to provide complete, self-sustainable and adaptive solutions to the enterprise of the future.

---

[2] European Commission: The EU Single Market (ec.europa.eu/internal_market/index_en.htm)

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 organisations, and covering many industrial areas and activities, from design to production and commercialization (European Comission, 2006; ISO TC184/SC4, 1994). However, implementing those models inside heterogeneous collaborative business networks is still an ongoing challenge hindered by the fact that they are, intrinsically, composed by many distributed hardware platforms, software tools and Information and Communication Technology (ICT). Even some standardisation groups, pressured by local communities, are developing specific solutions to provide national capabilities, when the full potential benefits can only be achieved if interoperability is underpinned by a coherent set of open, and internationally accepted ICT standards (Mason, 2007; Ray & Jones, 2006).

In this direction, efforts like STEP (Standard for the Exchange of Product model data) have tried to deal with integration and interoperability issues (ISO TC184/SC4, 1994). STEP appeared in the mid 80's and has evolved to become an important family of standards for the computer-interpretable representation of product information and for the exchange of product data under the manufacturing domain. It defines a framework which provides neutral mechanisms that are capable of describing products throughout their life cycle. Still, despite of the potentiality of STEP, its success has been limited to the large industries such as aeronautics, automotive, or shipbuilding industries (PDES Inc., 2006). Smaller sectors dominated by SME organisations have been evidencing reluctance in moving towards such kind of standards. They claim the technology is hard to understand and expensive since only a small set of tools work with STEP, especially comparing with the rising web technologies, such as XML[3] (Gielingh, 2008; Jardim-Goncalves, Agostinho, et al., 2007; Lubell et al., 2004; Pratt, 2001).

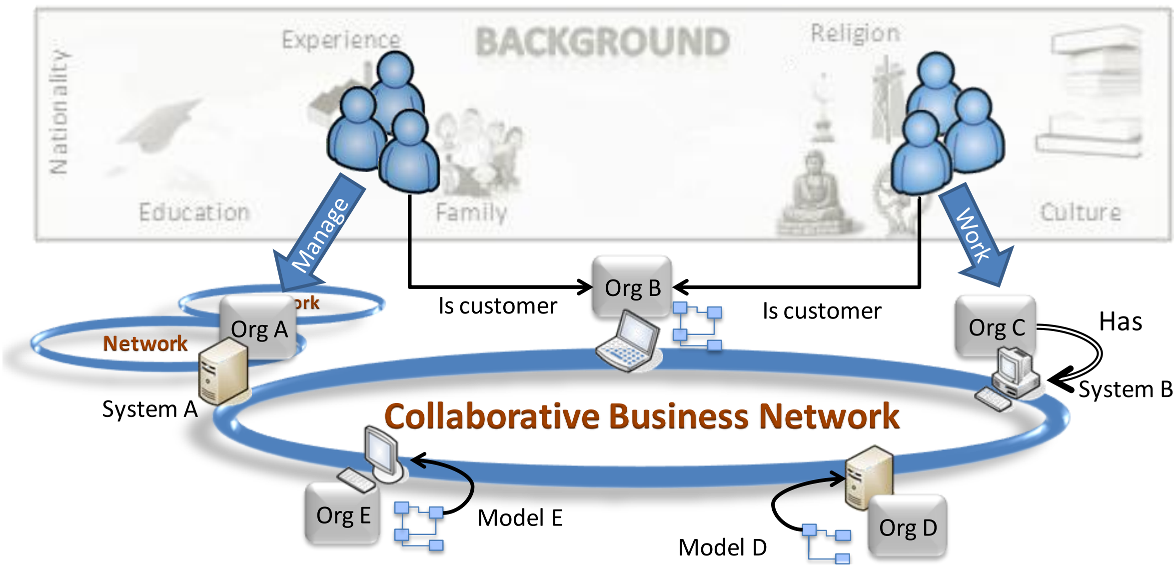


Figure 1.2: Example of Complexity in Collaborative Business Network

The absence of a universal solution for interoperability is an evidence of the **complexity** currently existing in business networks. As illustrated by Figure 1.2, even when collaborating, networks are

[3] XML - eXtensible Markup Language (www.w3.org/XML/)

filled with complexity at macro and micro levels. They are formed by organisations from the same or complementary domains, which are managed by people with different opinions and backgrounds based on several factors such as culture, professional experience, family, etc. In fact, humans play a fundamental role on networks. They manage, work, and are themselves customers of the different organisations which can belong to several networks at the same time. On the other hand, organisations have different enterprise systems structured according to several information models and standards implemented on multiple software platforms and technology.

Nowadays, enterprises have many structures and relationships. Besides being influenced by people, organisations influence each other, which lead to a wide complexity of interactions and a high probability of changing requirements, models or even systems. Therefore, all this heterogeneity leads, in most cases, the network (either multi-domain or not) to experience interoperability problems at the enterprise level, including systems communication.

Nevertheless, considering that the above issues will be gradually overcome as soon as research becomes successfully applied in industry, an emergent problem for seamless interoperation is arising, i.e. its sustainability along the network life cycle. This way, understanding the network and organisation’s interactions is considered a major factor in contributing for the success of interoperability solutions and the performance of the entire enterprise. Implicit in both is the view that enterprises are Complex and Adaptive Systems (CAS), e.g:

- Retail, design, and manufacturing systems are constantly adapting to new market / customer requirements, thus replying the need to respond with faster and better quality production;
- In a typical CPD process the design occurs largely before the final full-scale manufacturing, and in most cases this design is later modified or refined for the manufacturing difficulties, problems detected at simulation time, or even changing requirements;
- New enterprises are constantly entering and leaving collaboration networks, leading to a constant fluctuation and evolution of business networks and system models;
- And, in addition, even standards need to be adjusted from time to time.

All these factors are making interoperability difficult to sustain over time. Preliminary theories have been advanced in specific scientific disciplines, such as biology and ecology, to explain the importance and evolution of complexity in these “systems” (Axelrod, 1997). Some researchers have attempted to extrapolate results to a “general systems theory” or “complexity theory” that could explain the importance of the behaviour of systems in all fields of science (Gharajedaghi, 2005; Sugihara & May, 1990). These theories view all systems as dynamic “living” entities that are goal-oriented and that evolve and adapt to the surrounding environment over time.

Research challenges in the area of sustainability of systems interoperability within dynamic business networks are several, and include the study of:

- Enterprise interoperability itself;
- System monitoring, behaviour and adaptability;

- The “system” aspects of interoperability, from software component design to organisational structure to the communication, collaboration and coordination facilities;
- Decision support to minimize the impact of changing requirements and information models;
- Interoperability of digital ecosystems as complex systems of systems.

Breakthroughs in any of these areas would impact the way organisations address the problem of interoperability, hence would also impact today’s global market and economy. This way, the motivation for this thesis is to tackle these challenges as much as possible contributing for self-sustainable and interoperable collaboration networks of the future.

1.2. The Scientific Method

When referring to science, people are normally thinking about the act of gathering, organising and condensing knowledge about the world into testable laws and theories (Morris, 1992; Wilson, 1999). However, there is a plurality of different sciences that differ very much from each other, and might lead to confusion when trying to get to the concrete meaning of science. Everyone accepts physics as a well defined science, but for example, also history and linguistics are often catalogued as sciences. Thus, the method for scientific practice is therefore neither simple nor unambiguous, but it can be agreed that logic and mathematics (the most abstract and at the same time the most exact sciences) must be an integrant part of every science. They are essential for physics, less important for chemistry or biology, and their significance continues to decrease towards the more social and humanistic sciences (Dodig-Crnkovic, 2002).

Modern sciences introduce a paradigm shift since, unlike the traditional philosophy of science, they do not apply to a single domain, i.e. are interdisciplinary and “eclectic”. New sciences search their methods and raise questions in very broad areas crossing borders and addressing different scientific fields. For example, modern computer science is predominantly synthetic, in that formalisms and algorithms are created in order to support particular desired behaviour using concepts from physics, chemistry, biology and others (Berners-Lee et al., 2006; Dodig-Crnkovic, 2002). Concerning the topics addressed in this thesis, in spite of the research developed so far and the science base challenges launched by the last two EIRR, nowadays the scientific foundations for enterprise interoperability have not yet been established, and the work has mostly been related to computer science and the newly born services science.

Nevertheless, still within the category of modern sciences, the notion of network, namely collaboration and business networks, as introduced in section 1.1.1 “Enterprise Collaboration” is nowadays a central issue to address science within the topics of EI. Collaborative networks are complex systems, emerging in many forms in different application domains, and consist of many facets whose proper understanding requires the contribution from multiple disciplines (Camarinha-Matos & Afsarmanesh, 2005). EI, namely the systems interoperability field addressed in this dissertation, suggests that systems can seamlessly interoperate with others across networks, throughout research development in focal areas, removing barriers to interoperability, and transferring and applying the results in several industrial sectors.

Concerning the scientific method there are several variants in use, yet, the process of investigation is often referred in many textbooks and science courses as a linear set of steps through which a scientist moves from observation through experimentation and to a conclusion (see Figure 1.3). The method attempts to minimize the influence of the researchers' bias, i.e. personal preferences, common sense assumptions, concealing of data not supporting the hypothesis, etc., on the outcome of an experiment (Wolfs, 1996). However, this classic representation can have a number of problems because processes can be iterative or, in some cases, can even be skipped.

Figure 1.3: Classic View of the Scientific Method

Despite not sharing that vision, some claim the scientific method is not always required to start with a question, and sometimes does not even involve experiments (Carpi & Egger, 2008; Schafersman, 1997). Some scientific investigations can achieve results leading in directions not originally anticipated, or even in multiple directions (Carpi & Egger, 2010). Therefore, the logic of science is recursive/ iterative and also theory-contaminated, especially in modern interdisciplinary fields where hypotheses have its origins in the existing knowledge of the researcher (on the field itself or parallel fields), which is never universal and can change after experimentation, thus leading to new hypothesis (Dodig-Crnkovic, 2002).

1.2.1. Adopted Research Method

Ultimately, the choice of which research method (instantiation of the scientific method) to use is personal and depends on the scientist and the nature of the question addressed. To open to any potential result, the research method adopted for this PhD work is based on an eight step methodology that, as suggested in Dodig-Crnkovic (2002) considers the influence of the researcher's background knowledge in the scientific process, and envisages recursive iteration through different steps depending on the results obtained in the hypothesis testing (see Figure 1.4). The adopted method is described in more detail as follows:

1. *Choose a Topic*: The first step towards a successful scientific research consists in choosing a meaningful topic. In fact this can be seen as a preliminary step towards the real method because there is no point in conducting research in areas where the researcher has no interest. The next section 1.2.2 of this dissertation, reports on the results of this step.
2. *Formulate a Research Question*: This is one of the most important steps of the full method, since it scopes the entire work, and will never be revisited in the same research loop until a conclusion is achieved based on the analysed results (Camarinha-Matos, 2009). The research question may be complemented with secondary questions to narrow the focus of the study, but all must be capable of being confirmed or denied, i.e.

answered. This way, statements formulated under this step should always be clear and interrogative.

Depending on pre-existing knowledge the researcher tends to avoid questions guessed of not leading to concrete answers, thus prior knowledge influences the formulation of the research questions. Section 1.2.3 “Research Questions”, reports the results of this step.

3. *Do Background Research*: With the background research step, any studies elaborated for the PhD thesis gain a solid basis on the work of peers. During this stage the researcher does literature review to build a consistent scientific base for the envisaged research. Books, scientific publications, and other dissertations are used, and discussion groups consulted to verify if the work has been done previously or if there are similar approaches to build upon. Chapters 2 and 3 summarize the finding of this step’s activities.

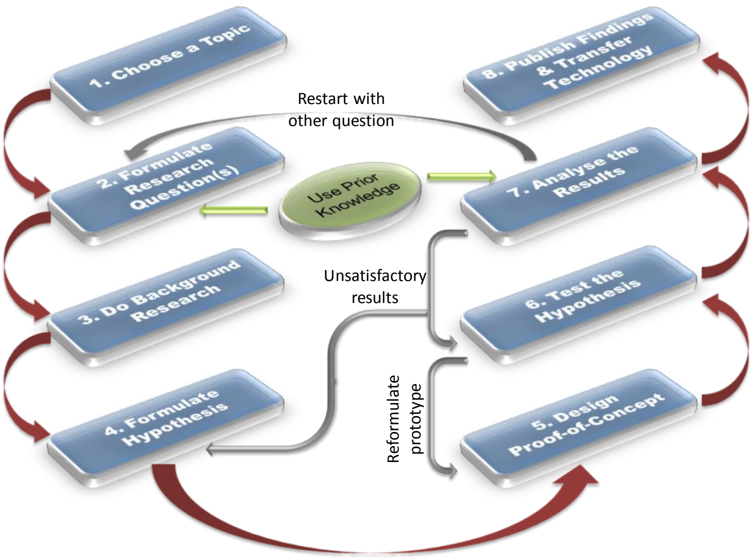


Figure 1.4: Adopted Research Method

4. *Formulate Hypothesis*: A scientific hypothesis uses the background research to state an educated guess regarding the variables involved (Camarinha-Matos 2009). It should be stated in a declarative format, which brings clarity, specificity and focus to a research problem. In critical thinking, as in science, the hypothesis or proposed answer to the research question must be testable, otherwise it is of no use to pursue further investigation. In fact, and as illustrated by Figure 1.4, the hypothesis can be revisited and reformulated in case unsatisfactory results are achieved during the more advanced stages of the scientific method.

Section 1.2.4 of this document presents the hypothesis statement.

5. *Design Proof-of-Concept*: The Proof-of-Concept (POC) is frequently related to engineering research and the development of a prototype. It is the evidence that demonstrates if an idea or contribution is feasible. This way, and since many times the complete validation of the hypothesis in a real world environment involves resources (time and money) that few have access, this step relates directly to the design of an experiment in a controlled environment.

To ensure that the POC is not only associated to a prototype but also to the thesis validation method, this step also includes planning in detail the validation phase. Section 5.1 “Application Scenarios and Test-Cases for a Self Sustainable Interoperability” details possible application scenarios and test-cases for the validation of the thesis contributions in a POC style, and section 6.1 “Validation Components” explains how the POC testing is part of the complete validation methodology.

6. *Test the Hypothesis*: This is the step where the testing of the hypothesis is actually done. It includes the implementation of the prototype, collection of data, and execution of tests according to the pre-defined validation method.

   Considerations regarding the implementations and ultimately the hypothesis are drawn and the researcher may find evidence that the hypothesis needs to be reformulated, thus it will need to jump back to step 4, or might need to propose adaptations to the prototype design (previous step). The POC implementation specificities are explained on chapter 5, while the test results are available throughout chapter 6 according to the validation methodology.

7. *Analyse the Results*: The factual results of the testing are to be verified under this step by means of quantitative and qualitative analysis. During this step it is important to have a critical spirit and promote discussion (Camarinha-Matos, 2009). If the conclusion is that the hypothesis failed the tests, it must be rejected and either abandoned or modified. The dissertation final considerations are focused on this stage.

8. *Publish Findings and Transfer Technology*: If the hypothesis passes all tests, it is considered to be a corroborated hypothesis, and can be published. To ensure a valuable scientific method, it is mandatory to publish findings and provide peers from the scientific and industrial communities the chance to verify, comment, and use the developed work. Despite appearing only as the final step of the adopted research method, intermediate findings should also be published.

   Part of the validation methodology considers this need, thus results are detailed under sections 6.3 ”Acceptance by Scientific Community” and 6.4 “Industrial Acceptance”, while the final publication will be the dissertation document itself.

During the course of this PhD work, a regular flow of the adopted research method, from steps 1 to 8 has been followed. Intermediate findings target publications at recognised conferences and journals, and backward loops were applied especially to improve the POC test cases.

1.2.2. Research Topic

The topic chosen and reflected in the dissertation title is **“Systems Interoperability”** which is a sub-topic of the wider domain “Enterprise Interoperability”.

1.2.3. Research Questions

Considering the chosen topic, the research question that gives the motto for this thesis work addresses the sustainability of the interoperable status of a collaborative business network, as introduced in section 1.1.3 "The Problem of Sustainability of Interoperable Solutions".

RQ: "How can the principles of complexity science be applied on dynamic business networks and contribute for a self-sustainable interoperability?"

To assure the research focus and targeted results, the major question is split threefold according to three major incognita:

- The detection of a problem
  - Q1.1: "How can complex collaboration networks be monitored to detect harmonization breaking points?"
- The response to the problem
  - Q1.2: "Which methods and tools can be reused, or need to be developed, to support a swift and integrated response to the network harmonization breaking?"
- The impact of the solution
  - Q1.3: "To what extent does the transient of the node dynamics impact the remaining business collaboration network?"

Despite not being stated as a specific research question because it is not part of the major line of work, this thesis targets to contribute for the development of the enterprise interoperability science base, a common challenge of the last two EIRRs (Enterprise Interoperability Cluster, 2008; FInES Research Roadmap Task Force, 2010).

1.2.4. Hypothesis

Based on the background observation summarized on chapters 2 and 3, as a response to the research questions and identified challenges, one can estimate that:

"If the model morphisms, defined at the time of interoperability establishment between the systems' nodes operating within collaboration networks, are monitored by expert agents capable of triggering CAS-like self-organisation procedures when interoperability harmonization breaking is detected, then dynamic business networks could display sustainability of their systems interoperability"

The above statement is therefore the adopted hypothesis that was implemented, tested and challenged during this PhD work.

1.3. Dissertation Outline

The dissertation document is organised in eight chapters:

1. Introduction
2. Enterprise Systems Interoperability

3. Complexity and Dynamic Environments
4. Self-Sustainable Interoperability
5. Proof-of-Concept Implementation
6. Implementations Testing and Hypothesis Validation
7. Final Considerations and Future Work
8. References.

The Introduction, i.e. the current chapter, begins by explaining the enterprise collaboration framework that scopes this work and motivates the reader, highlighting the importance of systems interoperability in today's global networks, and unfolding the problem of sustainability of interoperable solutions. The research method that has been adopted during this PhD work is also explained presenting the research topic, research questions, and the hypothesis, drawn based on the background analysis reported in the following chapters.

The second and third chapters include the background findings on the two areas targeted in the PhD, i.e. systems interoperability, and complexity and dynamic environments. In the context of the first, novel research areas such as enterprise interoperability, information modelling and ontologies, model-driven engineering, model morphisms, service-oriented interoperability, semantic-based interoperability, and sustainability of systems and interoperability, are addressed, presenting some real challenges that the EI community is facing. The second topic explores older and more solid-grounded concepts such as systems, complexity science and its respective theory, complex networks, and also looks upon some system/network modelling approaches, namely agent-based modelling as a means towards optimisation of systems and networks behaviour. This second topic is analysed envisaging a possible input it might have for systems interoperability frameworks.

Chapter number four presents the core contribution, a framework for sustaining systems interoperability based on complex adaptive systems theory. It starts by positioning the contribution concerning the level of interoperability addressed, drawing a line between other existing frameworks. The conceptual solution is explained, firstly at a higher abstraction level introducing some novel concepts such as "harmonization breaking", and then in more detail, covering the specificities of each package composing the proposed framework. Still within this chapter, the author provides a vision of how the contributions cover the identified background challenges and research questions.

A proof-of-concept implementation together with the possible application scenarios and technical test cases is depicted on chapter 5, namely focusing on the explanation of the implementation steps, decisions made, and the technology required to address the scenarios. To maximize the appreciation of the framework and also the validation of implementations, two types of industrial networks are considered, a supply chain and collaborative product development. Nonetheless, the actual testing of the implementations is only reported under chapter 6, with a hybrid functional and non-functional testing methodology. In this chapter, the reader is also informed about the wider acceptance of the work presented, both by peer researchers and also industrials. Not focusing only on the framework for sustaining systems interoperability *per se*, the dissertation document,

illustrates the parallel contributions this work enabled towards the identification of a science base for enterprise interoperability.

The final considerations and future work topics are presented in the seventh chapter, enhancing the novelty of this research and major milestones achieved. This is followed by the extensive list of bibliography.

2. Enterprise Systems Interoperability

Enterprise systems interoperability is a growing interest area due to the continuous need of integration of new, legacy and evolving systems, in particular in the context of networked enterprises and e-business (Athena Partners, 2007). *Although enterprises are heavily dependent on ICT solutions in their day-to-day business operations, these are often inflexible and difficult to adapt to meet the requirements of dynamic and evolutive networks* (Campos et al., 2005; Truex et al., 1999). *The current ICT solution space still suffers badly from lack of interoperability, with systems having information exchange difficulties and compatibility issues, thus damaging the competitiveness of enterprises and forcing the adoption of expensive and complex software platforms that not all can support.*

2.1. Enterprise Interoperability (EI)

Enterprise systems are large-scale, application-software packages that use the computational, data storage, and data transmission power of modern information and communication technology to support processes, information flows, reporting, and data analytics within and between complex organisations. The integrated content managed by these systems may be used to provide a configuration management solution throughout the life cycle of products and processes.

It is recognised that the advantage of one company over another stems from the way it manages its process of innovation. However, if the enterprise systems used are not efficient and experience communication and automation issues, innovation might not be realised. Hence, EI has become an important area of research to ensure the competitiveness and growth of enterprises (Enterprise Interoperability Cluster, 2008; FInES Research Roadmap Task Force, 2010).

2.1.1. Interoperability Definitions

There are numerous definitions for interoperability, which is why it is such a challenging topic to cover. Following, some relevant interoperability definitions that help understanding the concept and motivation for this thesis are presented:

- *"The ability of two or more systems or components to exchange information and to use the information that has been exchanged"*[4] (IEEE, 1990; Software Engineering Institute, 2002);
- *"The ability to operate in synergy in the execution of assigned tasks"* (DoD, 2011);
- *"The ability of disparate and diverse organisations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between the organisations, through the business processes they support, by means of the exchange of data between their respective ICT systems"* (ISA, 2010).
- *"A property referring to the ability of diverse systems and organisations to work together (interoperate). The term is often used in a technical systems engineering sense, or alternatively in a broad sense, taking into account social, political, and organisational factors that impact system to system performance"* (Wikipedia, 2012);

As a well established area of applied research, EI examines the problems related with the lack of interoperability in organisations and proposes novel methods and frameworks to contribute with innovative solutions addressing situations where the information exchanged is not being understood or used correctly. EI can apply to both inter and intra-enterprise needs and includes the concepts of extended enterprise, virtual enterprise and sub-systems of one enterprise (ISO TC184/SC5, 2011).

2.1.2. Background: Enterprise Architecture and Integration

Chronologically speaking, the term enterprise interoperability is quite recent and one needs to understand the enterprise itself in order to realise the challenges behind EI. Even when speaking of systems interoperability, it is possible to go back to the beginning of the World Wide Web (WWW) in 1980s. Of course before that there were ways for systems to interoperate, but it was not given the research importance as after the WWW and the price drop of technology, when enterprises begun to recognise the potential of interoperability through e-commerce and e-business (already in the 1990s).

According to ISO 15704, an enterprise is defined as one or more organisations sharing a specific mission, goals and objectives to offer an output such as a product or a service (ISO TC184/SC5, 2000). This broad definition also covers the extended and virtual enterprise concept, that despite of the operational differences, are based on the idea of putting together on a network, capabilities and competencies coming from different organisations (Camarinha-Matos & Afsarmanesh, 1999; Dyer, 2000). Due to such a vague definition, Enterprise Architecture (EA) is both a challenging and confusing concept that also needs to be understood when talking about the enterprise.

---

[4] most used definition

When compared to other fields where the architect has a solid role and needs to use recognised standards to specify and architect (e.g. a building), in enterprise it tends to use many heterogeneous and often overlapping approaches, which create obstacles for correct understanding of systems and their capabilities in industry (D. Chen et al., 2008). EA should be organised in a way that supports reasoning about the structure, properties and behaviour of the system, thus defining its components and providing a blueprint from which it can be developed. Therefore, integration is a property that necessarily needs to be part of an EA, and has become an established research domain since the 1990s, as the extension of computer integrated manufacturing.

Enterprise integration is an essential component of enterprise engineering, concerning the usage of specific methods, models and tools, to design and to continually maintain an enterprise in an integrated state so that it can fulfil domain objectives (Panetto & Molina, 2008). During the 1980s and 1990s, a lot of research has been carried out in Europe and the USA to develop integrated enterprise architecture frameworks (Chalmeta et al., 2001; D. Chen et al., 2008). Among some of the most known are CIMOSA[5], PERA[6], GIM[7] architecture, ARIS[8], DoDAF[9], TOGAF[10], or the recognised Zachman Framework[11] which specified the initial fundamental structure for EA. However, as explained in the introduction chapter, integration involves some degree of functional dependence and goes a step further from coordination, cooperation and collaboration where interoperability has major importance (CRESCENDO Partners, 2009b; INTEROP Partners, 2006c). An integrated system loses significant functionality if the flow of services is interrupted.

Today's enterprises are often faced with how to best manage their investment in existing system architectures but also need some degree of flexibility, evolving these architectures to meet ever-changing business demands. In many scenarios not covered by the above frameworks, enterprises encounter the challenging task of enabling coexistence between new and existing systems. They need to split functionalities, and besides integration they also need interoperability (Panetto, 2007).

2.1.3. Interoperability Maturity Models and Levels

Several authors have been proposing different solutions to assist in achieving interoperability between systems, which requires resolution of issues at various distinct interoperability layers. Normally, the proposed types of interoperability follow a scale of advancement, in which the higher a type is placed in the scale, the more advanced and complete the interoperability is accomplished. An early classification for interoperability maturity was defined in the Levels of Information System Interoperability (LISI), focusing on the assessment of systems against increasing levels of sophistication regarding the exchanging and sharing of information and services during the

---

5 Computer Integrated Manufacturing Open System Architecture (http://www.cimosa.de/)
6 Purdue Enterprise Reference Architecture (http://www.pera.net/)
7 GRAI Integrated Methodology (D. Chen & Doumeingts, 1996)
8 Architecture of Integrated Information Systems (Scheer & Schneider, 2006)
9 Department of Defense Architecture Framework (http://dodcio.defense.gov/sites/dodaf20/)
10 The Open Group Architecture Framework (http://www.opengroup.org/togaf/)
11 http://test.zachmaninternational.com/index.php

system's life cycle (C4ISR Architecture Working Group, 1997). As evidenced in Figure 2.1, this occurs through five levels, i.e.:

- *Level 0 - Isolated interoperability* in a manual environment, where direct electronic connection is not allowed or available and systems are typically stand-alone;
- *Level 1 - Connected interoperability* in a Peer-to-Peer (P2P) environment, where systems are providing some form of simple electronic exchanges for homogeneous data files, e.g. media;
- *Level 2 - Functional interoperability* in a distributed environment that enables data sets to be passed from system to system with the use of the formal data models;
- *Level 3 - Domain-based interoperability* in an integrated environment of shared data, which is understood by multiple users thanks to a domain-based data model; and finally
- *Level 4 - Enterprise-based interoperability* in a universal environment, where systems are capable of operating using a distributed global information space across multiple domains. This level of interoperability means that multiple users can access and interact with complex data simultaneously that is shared and distributed throughout the environment.

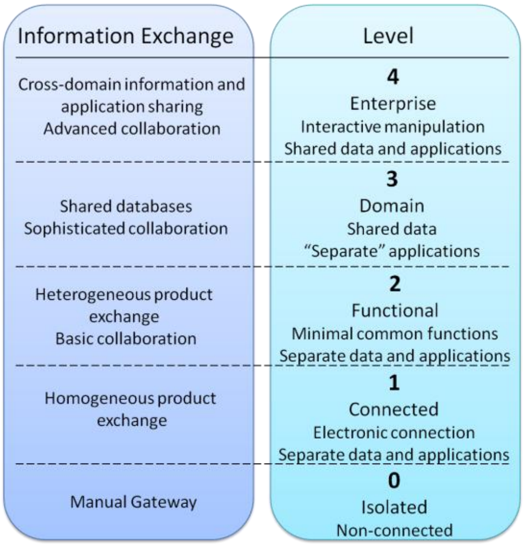


Figure 2.1: LISI Maturity Levels (adapted from C4ISR Architecture Working Group (1997))

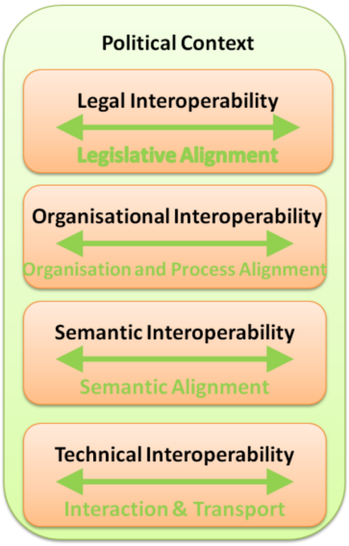


Figure 2.2: EIF Framework Interoperability Levels (adapted from ISA (2010))

Later, other examples of classification from relevant interoperability frameworks and models followed and have been widely referenced, e.g., Peristeras & Tarabanis (2006) proposed the Connection, Communication, Consolidation, Collaboration Interoperability Framework (C4IF), which uses basic linguistic concepts to categorize information systems communication in four layers. Tolk & Muguira (2003) proposed the Levels of Conceptual Interoperability Model (LCIM) to bridge the gap between implementation focused methods and conceptual models (later revisited in Wenguang Wang et al. (2009)), while Panetto (2007) summarized several others proposing different classifications based on maturity levels for interoperability.

The European Interoperability Framework (EIF) is another example of interoperability levels classification (ISA, 2010). Developed with the aim of addressing the provision of European Public Services (EPS), a sector supplied by public administrations, the EIF recognises interoperability in four complementary levels within a political context where visions are shared, objectives agreed upon, and priorities aligned in response to a new EPS requirement (see Figure 2.2):

- *Technical Interoperability*, which covers the technical aspects of the information systems such as interface specifications, data integration services, data presentation and exchange, etc;
- *Semantic Interoperability* that enables organisations to process information from external sources in a meaningful manner. The exchange of information at this level follows a minimum set of understanding that needs to be preserved through the different exchanges;
- *Organisational Interoperability*, which occurs at the business process level ensuring that organisations collaborate to achieve their mutual agreed goals; and
- *Legal Interoperability* that currently is one of the biggest problems in EPS as each works within its own national legal framework, causing significant barriers for interoperability.

With the risk of not presenting all, Table 2.1 summarizes an extensive list of interoperability maturity models and levels that have been analysed in this review.

Table 2.1: Interoperability Maturity Models and Levels

Approach	Classification Layers	Type
LISI (C4ISR Architecture Working Group, 1997)	Five interoperability maturity levels focused on technical interoperability and the complexity of interoperations: • Isolated Systems, Connected Systems, Distributed Systems, Domain Systems and Enterprise Systems, Affecting four layers of information: • Procedures, Applications, Infrastructure and Data.	Maturity Levels & Interoperability Layers
OIM[12] (Thea Clark & R. Jones, 1999)	Five maturity levels extending LISI into the more abstract layers of organisational command and control support: • Independent, Ad hoc, Collaborative, Integrated, and Unified.	Maturity Levels
NC3TA[13] **Reference Model for Interoperability (RMI)** (NATO, 2003)	Five degrees and sub-degrees of technical interoperability: • No Data Exchange, Unstructured Data Exchange, Structured Data Exchange, Seamless Sharing of Data, and Seamless Sharing of Information.	Maturity Levels
IDEAS Interoperability Framework (IDEAS Thematic Network, 2003)	Three layers of interoperability contemplated inside an enterprise: • Business, Knowledge, ICT systems, And a Semantic and Quality dimensions that cuts across the three identified layers, focusing on supporting mutual understanding among all.	Interoperability Layers

[12] Organisational Interoperability Maturity model
[13] NATO C3 Technical Architecture

Approach	Classification Layers	Type
CEN/ISSS Levels of Interoperability (CEN/ISSS, 2006)	Three basic levels of interoperability: • Technical interoperability, Semantic or business interoperability, and Organisational interoperability. Sector-specific issues can cut through the entire stack.	Interoperability Layers
C4IF (Peristeras & Tarabanis, 2006)	Four interoperability layers for systems interchange: • Connection, Communication, Consolidation, and Collaboration, Containing three objects of integration: • Channel, Information, Process.	Maturity Levels & Interoperability Layers
AIF[14] (Athena IP, 2006; A. Berre et al., 2007)	Five level maturity scale: • Performed, Modelled, Integrated, Interoperable, and Optimising, Where the organisational context is provided for more specific and technical improvements across the three views on interoperability: • Conceptual, Applicative, and Technical. All of which, can be addressed across four interoperability layers with different concerns: • Enterprise/Business, Processes, Services, and Information/Data.	Maturity Levels, Interoperability Views & Layers
Interoperability classification framework (Panetto, 2007)	Six kinds interoperability solutions: • Synchronic Interoperability, Model-driven Interoperability, Semantic-driven Interoperability, Vertical Interoperability, Horizontal Interoperability, Diachronic Interoperability.	Maturity Levels & Interoperability Layers
Maturity levels for interoperability in digital government (Gottschalk, 2009)	Five-level model: • Computer Interoperability, Process Interoperability, Knowledge Interoperability, Value Interoperability, and Goal interoperability. Might be applied by public organisations to identify current maturity and future direction for improved interoperability.	Maturity Levels
EIF (ISA, 2010)	Four complementary levels within a political context: • Legal Interoperability, Organisational Interoperability, Semantic Interoperability, and Technical Interoperability.	Interoperability Layers
LCIM (Wenguang Wang et al., 2009)	Seven levels of conceptual interoperability that go beyond technical models like LISI: • L6 (Conceptual), L5 (Dynamic), L4 (Pragmatic), L3 (Semantic), L2 (Syntactic), L1 (Technical), and L0 (No Interoperability)	Interoperability Layers

2.1.4. Holistic View on Interoperability

The ATHENA Interoperability Framework (AIF) was developed with the aim of adopting an holistic perspective on interoperability (Athena IP, 2006; A. Berre et al., 2007). The vision is that interoperation is only meaningful, when all relevant levels of an enterprise are addressed. Thus, the

[14] ATHENA Interoperability Framework (AIF)

diversity, heterogeneity, and autonomy of software components, application solutions, business processes, and the business context of an enterprise must be considered. Following that trend, the definition of interoperability has been revisited a number of times to reflect the importance it has been gaining to the enterprise. Adapting the concepts from EA, interoperability should be a design principle that must consider every enterprise asset and the complexity of existing relationships, not only at data and applications levels but also at personal, management, and even geographical and cultural levels.

Enterprise interoperability can accordingly be defined as *"providing services to enable two users working in different locations, projects and settings to share knowledge, information, data, software, solutions and ICT resources"* (IDEAS Partners, 2003). In EI, each organisation retains its independence and should gain the capability to communicate and interoperate with others. Nevertheless, enterprise systems frequently fail to interoperate because of barriers of various kinds that must be addressed in this holistic view.

2.1.4.1. Interoperability Levels and Concerns

The reference research projects on EI, INTEROP NoE (2006c) and Athena IP (2007b), later complemented in the ISO standard 11354 (ISO TC184/SC5, 2011), begun to build the holistic vision addressing interoperability across four levels (see Figure 2.3), each with different concerns:

- <u>Business</u>: is the top layer of the enterprise and where all issues related with the management of the operations are addressed. Amongst others, they include the way an enterprise is organised, how it operates to produce value, how it takes decisions, and how it manages its relationships (both internally with its personnel and externally with partners, customers, and suppliers). Interoperability at this level should be seen as the organisational and operational ability of an enterprise to accurately cooperate with other external organisations, whether these are enterprises or public institutions;

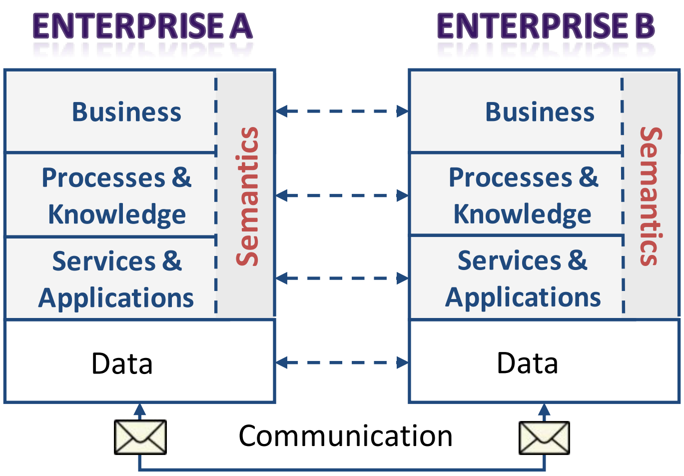


Figure 2.3: Holistic Approach to Interoperability (adapted from Athena IP (2007a))

- <u>Processes & Knowledge</u>: this layer deals with acquiring a deep and wide knowledge of the enterprise. It includes the alignment of information and other units needed for process operation, i.e. knowledge of internal aspects such as process orchestration, products, but

also of external aspects such as laws and regulations, legal obligations, and relationships with public institutions. Interoperability at this level should be seen as the compatibility of the skills, competencies, and knowledge assets of an enterprise with those of external organisations;

- *Services & Applications:* is focused on the application level of the enterprise, dealing with ICT solutions that allow an enterprise to operate, namely methods, tools and services to make decisions, and interact within and outside its boundaries. Interoperability of services refers to the ability of partners to request, provide, and utilize each other's services. Therefore, this level is frequently designated as technical interoperability and should be seen as the ability of an enterprise's ICT systems to cooperate with those of other external organisations;
- *Data:* represents the actual data exchange layer that according to this ubiquitous view is dependent on the all the above described.

As illustrated in Figure 2.3, *semantics* are a vertical dimension cutting across the other layers. It is concerned with capturing and representing the actual meaning of concepts and thus promoting understanding and wisdom. To overcome the semantic issues that emerge from different interpretations of descriptions, precise computer processable meaning must be associated with each concept used in any of the layers described. It has to be ensured that semantics are exchangeable and based on a common understanding to be indeed a means to enhance interoperability.

2.1.4.2. Interoperability barriers

Many interoperability issues are specific to particular application domains while barriers are generic incompatibilities and mismatches that obstruct the sharing and exchange of information. Following the framework for enterprise interoperability proposed in the ISO standard 11354, three kinds of barriers need to be detailed to enable the correct identification of a problem and consequent approach towards a holistic solution (ISO TC184/SC5, 2011):

- *Conceptual barriers*: are the most significant to interoperability because of the need to overcome both the exchange and usability of content. Conceptual barriers shall be detailed in terms of the syntactic, semantic, and semiotic incompatibilities of exchangeable items;
- *Technological barriers*: are additional barriers with respect to conceptual barriers. They exist because of one or more ICT mismatches, e.g. a significant technological barrier for enterprise interoperation involves incompatible interfaces between the different systems, which can be enterprises, human-beings or computer systems;
- *Organisational barriers*: are related to human and organisation behaviours. Indeed, when two enterprises have different organisational structures (e.g., hierarchical vs. matrix authority) and decision processes, mappings are likely to be needed before the two enterprises can interoperate. Compared with conceptual barriers (centred on information problems) and technological barriers (associated with computational problems),

organisational barriers often originate from human-related issues but have an impact on the interoperation of ICT systems.

2.1.4.3. Interoperability Approaches and Solutions

As evidenced by the multiple frameworks enumerated in the previous chapters, several authors are conducting research and implementations following their directives to address several interoperability concerns and covering some of the identified barriers (e.g. Boudjlida & Panetto 2007; Chen & Doumeingts 2004; Franconi 2004; Jardim-Goncalves, Grilo, et al. 2006; Jardim-Goncalves, Figay, et al. 2006; Jardim-Goncalves et al. 2009; Lubell et al. 2004; Missikoff et al. 2003; Panetto et al. 2006; Ray & Jones 2006). Most rely on internationally accepted methods, reference data models, ontologies, open platforms and services to enable e-business, product data management, product lifecycle management, and design and simulation applications to interoperate.

Still, so far the deployment of EI has been based strongly on the support of a "big bang" transition to a more efficient, enterprise-wide "best way" of working. However, this new best way of working is defined by a relatively small group of dedicated expert analysts (industrial engineers, business engineers, information analysts etc.) and implemented and maintained top-down. According to this approach, most people in the enterprise (i.e. those not part of the group that is dedicated to the design and automation of the work processes) have limited influence on the deployment of EI. Therefore, this approach is leaving bottom-level knowledge in enterprises largely unexploited and solutions are sometimes not getting the expected adherence. Under the current conditions of fierce competition for knowledge, this under-utilisation of experience and expertise is not an effective strategy (Informal Study Group on Value Proposition for Enterprise Interoperability, 2008).

Due to this, and also to the limited budget available for the "big-bang" transitions, many organisations, namely SMEs, have been adopting precarious solutions that move away from the multidisciplinary holistic vision. Nonetheless, one must move away from monolithic, centralised architectures and software paradigms that focus only within the single enterprise and not on the business partners and customers. As envisaged in the EI research roadmap, developments on enterprise interoperability should target a newer vision to interoperability describing it as a *"utility-like capability that enterprises can invoke on the fly in support of their business activities with specific ICT functions being delivered as services that are cheap, fast, reliable, and without major integration efforts"*. The overall aim is to make ICT become *"a transparent and invisible part of the business operation"* (Enterprise Interoperability Cluster, 2008). Thus, development approaches should not follow "big-bang" transitions, but progressive ones with holistic concerns selecting and evolving if necessary through the three approaches described in the ISO standard 11354 (ISO TC184/SC5, 2011):

- *Integrated*: a common format shall be used to ensure a syntax and semantics that are recognised in the same manner by partner enterprises participating in shared businesses. This common format is not necessarily an international standard but needs to be agreed by participating enterprises so that is sufficiently expressive to capture the details that affect

interoperability and enables to build systems accordingly. This approach is suitable when designing and implementing new enterprise systems but is less satisfactory for reengineering existing ones because of the high development costs;

- *Unified*: a common meta-level structure applicable for the participating enterprise shall be identified and detailed. Unlike the integrated approach, this structure is not an executable entity to be directly embedded in the enterprise systems. Instead, it shall provide a means for semantic equivalence to allow mapping between entities. Using this meta-level structure, a translation between the constituent entities of the different enterprise systems is then possible. However, that translation might involve the loss of some information because the participating enterprises can have different extensions or instantiations of the same meta-level structure.

  The unified approach is particularly suitable when developing interoperability for collaborative or networked enterprises. To be interoperable with networked partners, a new company maps its own model or system to the neutral meta-format without the necessity to make changes on its own model or system. This approach has an advantage over the integrated approach because of the reduced efforts, time and cost in implementation. It is also suitable for the situation where a large company needs to interoperate with SMEs since these organisations normally work with more than one large company and this approach can facilitate coordination without requiring conformance to potentially conflicting processes or environments;

- *Federated*: parties shall accommodate and adjust their operation dynamically without the need for a common implementable format or mediation meta-model. While there can be a common understanding between the partners, in this approach no partner imposes their own models, languages and methods of work. Interoperation can be supported by employing agents to discover the needed information at run time and map corresponding input and output information of the entities and identifying inconsistencies to be later solved manually.

  The difference between this approach and the unified approach is that meta-models are not pre-defined; they are established dynamically through negotiation. Therefore, the federated approach is particularly suitable to virtual enterprises where diverse companies combine their resources and knowledge to manufacture a product for a limited duration.

All three approaches allow the establishment of interoperability between enterprise systems. The federated approach is considered as the most challenging for achieving interoperability, but the ideal choice depends on the context and requirements of interaction. When the need for interoperability arises from a merge operation or long-term collaboration, the unified approach is a possible solution since the expenses of establishing interoperability are amortised over the entire collaboration period. The federated approach will yield the most satisfactory results for a short-term collaboration, and the integrated one for new systems development.

2.1.4.4. Holistic Framework

Having understood the needs and concepts presented before, a holistic framework for EI focuses on requirements to enable communication rather than defining the communication itself, and is thus independent of specific technologies. As explained before, such a framework has been introduced by the reference research projects on enterprise interoperability, INTEROP NoE (2006c) and Athena IP (2007b), and later complemented in the ISO standard 11354 (ISO TC184/SC5, 2011). Figure 2.4 illustrates its composition, evidencing the complementarity between the three dimensions of levels and concerns, barriers, and approaches, where specific solutions are only relevant to the resolution of interoperability difficulties if they contribute to overcome one or more barriers.

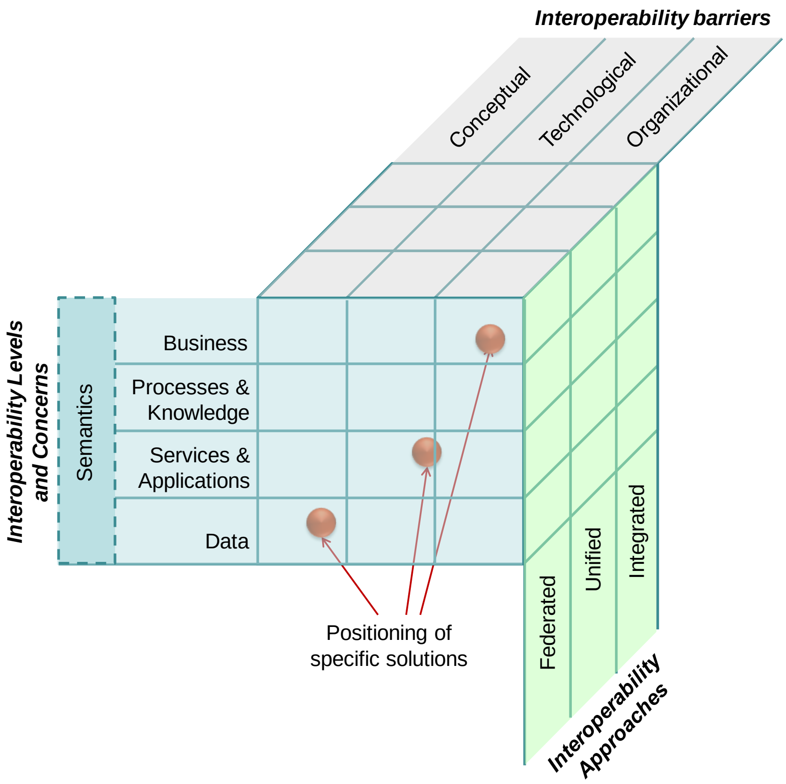


Figure 2.4: Holistic Framework for EI (adapted from Chen et al. (2008))

2.2. Information Modelling, Standards and Ontologies

Analysing the holistic view for EI, namely the three interoperability approaches, it has become evident that formats, models and meta-models are a core asset to interoperability. Defined by the construction of computer-based symbol structures (e.g. items, entities and relations) which are able to capture and express the meaning of information and knowledge about a system, organising it in a precise form that makes it understandable by people, and executable by systems, information modelling can amplify the productivity of skilled programmers (Mylopoulos, 1998). It enables them to address more complex and challenging problems. Indeed, modelling serving as a core technology for information Systems (IS) engineering is a ubiquitous and precious resource.

Modelling and model simulation are quickly becoming the primary enablers for complex system design, since they can represent knowledge in an intricate and complex way and at various abstraction levels allowing automated analysis (Mosterman & Vangheluwe, 2004).

2.2.1. Models and Meta-Models

A **model** is a definition of some slice of reality which is being observed and interpreted. It is constructed through the use of abstract elements and relationships in order to match corresponding real-world elements. In some contexts, e.g. Model-Driven Development (MDD) and Engineering (MDE) (as analysed in section 2.3 "Model-Driven Engineering"), the reality or object in study is designated as the System Under Study (SUS). Models can represent different aspects of one reality, derive from different natures or be created using varied languages, paradigms, concepts and formalism levels (INTEROP Partners, 2005a).

As evidenced in Figure 2.5, models must be written in a well defined modelling language since the symbols and relationships that are used to model a SUS should be described both syntactic and semantically in a fixed and coherent form. The modelling language, in its turn, is described by a *meta-model* – a model specifying constructs and relationships used in a given modelling language, which makes solid defined statements about what can be expressed in a valid model using that particular language. Hence, a valid model is only conformant to its meta-model when it does not violate any statement and constructs inherited or deducible from its language. On the other hand, a meta-model is also a model, thus must also be written in a coherent language – the meta-language, which is responsible to describe modelling languages in the same way of the meta-model/model relation (Tony Clark et al., 2002).

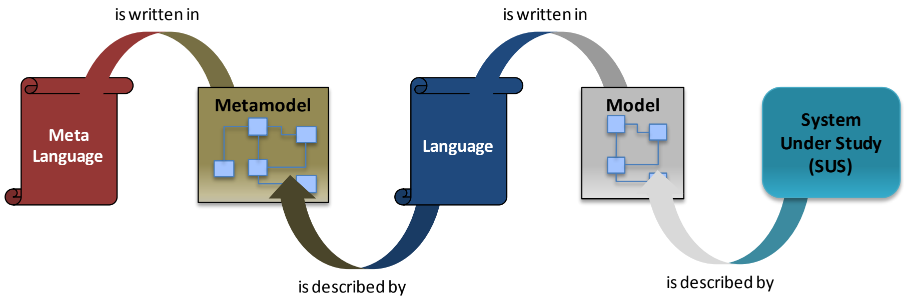


Figure 2.5: Engineering Modelling Concepts' Relationship

When talking about models it is useful to classify them according to the meta-modelling level (INTEROP Partners, 2005b; Frankel, 2003):

- *Level 0* - is the level commonly know as the instance level since it is not possible to instantiate any further (e.g. a XML document);
- *Level 1* - is the model level, which has to be instantiated to obtain data (e.g. XML Schema);
- *Level 2* - describes the language itself, commonly called "meta-model" (e.g. XML language);
- *Level 3* - is the base model for generating different languages (meta-models) (e.g., SGML).

Normally is not common to consider more than 4 meta-modelling levels, such as in OMG's[15] Meta Object Facility (MOF[16]) framework, but other approaches can only consider 3 levels. Nevertheless, a reflexive meta-model prevents the indefinitely increase of abstraction layers (model, language, meta-model and meta-languages layers), since it is expressed using the minimal set of elements of the modelling language to express the statements of the meta-model. This way, a meta-meta-model is a self-describing model which self-conforms to its own semantics. A few examples are MOF which is the top layer of the Model-Driven Architecture (MDA) (explained under section 2.3.1), and Ecore which has been introduced with the EMF[17] (Seidewitz, 2003).

Either being used in the form of traditional Entity-Relationship (ER) databases, architectural models, or domain ontologies, models can be characterized according to four dimensions (INTEROP Partners, 2005b):

- *Meta-model* - corresponding to the language for modelling, as described above;
- *Structure* - corresponding to the topology associated to the model schema;
- *Terminology* - the labels of the model elements that don't refer to modelling primitives; and
- *Semantics* – which given a "Universe of Discourse", are the interpretations that can be associated with the model.

This way, model operations can be classified as acting on any of these dimensions. However, despite the rigid formalisation, models can be generated from several different modelling languages (from different meta-models) with different syntaxes, expressive power, and semantics. Thus, if different systems expressed by different models need to communicate (e.g. using a unified or federated interoperability approach), it will be difficult to achieve a lossless expressiveness "link" between them.

2.2.2. Ontologies and Folksonomies

The term ontology has its origin in philosophy. Nevertheless it has been adopted by computer science to define a special type of information model. It produces a common language for sharing and reusing knowledge about phenomena in a particular domain, thus it is a model of knowledge. Ontologies are used in information systems design to standardize terminologies, map requirements, and promote information exchange using an agreed specification of how to describe all the concepts (e.g. notions of objects, people, processes, relationships, transactions, etc.) of a particular domain of interest (Gruber, 1993; Gruber, 1995; Edgington et al., 2004). Thus, by defining concepts and their relationships, ontologies provide a common understanding of the same, that before may have had different views and interpretations from the different practioners of that domain (Berners-Lee & Fischetti, 1999; Guarino, 1995; Guarino et al., 2009).

Following very simple modelling principles, an ontology uses classes, properties and relationships to define a hierarchical view of the world (designated by taxonomy). It is engineered by members of

[15] OMG – Object Management Group (www.omg.org/)
[16] MOF – Meta Object Facility (www.omg.org/mof/)
[17] EMF – Eclipse Modelling Framework (www.eclipse.org/modeling/emf/)

a domain which try to represent a reality as a set of agreed terms and logically-founded constraints on their use (Mika, 2005). It is also a way of specifying the structure of domain's knowledge in a formal logic designed for machine processing, thus systematizing the description of complex systems. This way it is acceptable to conclude that ontologies enable the creation of a kind of "*Lingua Franca"* for common understanding and exchanging (Barata, 2003).

Nevertheless, since there are normally multiple domain experts which may not all be involved in the process of creating a single ontology, one might find several ontologies for the same area of knowledge. Moreover, in those cases they tend to be described in very specific and often eclectic ways which makes them often incompatible. That problem is even more evident in ontologies created for the social dimension. Due to the dynamics of the community itself, with members leaving and entering, and with their commitments changing new contributions may invalidate knowledge already codified in the ontology. To address this problem of an ontology drift, several authors expect that the individual interactions of a large number of rational agents would lead to global effects that could be observed as semantics emergence (Aberer et al., 2004). Ontologies would thus become an evolving part of the system as opposed to be a fixed, limited contract of the majority, as is in standards.

Nowadays, ontologies are also the enabling technology for the Semantic Web. They are a means for people to state what they mean by the terms used in data that they might generate, share, or consume. In fact, an emergent concept is growing called Folksonomy, arising from data about how people associate terms with content that they generate, share, or consume (Mika, 2005). In contrast with taxonomies, that limit the dimensions along which one can make distinctions, folksonomies are massively dimensional and there is no global consistency imposed by current practice. Recently the two ideas (ontologies and folksonomies) have been put into opposition. However, they have distinct roles: ontologies, namely domain ontologies are more appropriate for systems interoperability enabling automatic reasoning on classified data, while folksonomies are more appropriate for massive loads of information with a high degree of liberty like in the social media (Gruber, 2007).

2.2.3. Standards

Standards are of key importance to enable enterprise interoperability. They are great enablers to the agreement of terminology, thus allowing communication and cooperation between software components, processes, organisation units and humans (D. Chen & Vernadat, 2002). Standardisation initiatives, supported by standardisation bodies (such as ISO[18], IEC[19]), developed by industrial communities (e.g. IEEE[20]) or by European projects, have been trying to contribute towards data exchange and systems communications. However, each focuses on one particular aspect of interoperability without aligning their enterprise knowledge and skills for taking advantage of seamless cooperation (Panetto, 2007).

---

[18] ISO - International Organisation for Standardization (www.iso.org/)
[19] IEC - International Electrotechnical Commission (www.iec.ch/)
[20] IEEE - Institute of Electrical and Electronics Engineers (www.ieee.org/)

A large number of standardization groups exist, supported by local governments and international communities. Nonetheless, even among them they have replication of efforts, while the full potential benefits could only be achieved if interoperability is underpinned by a coherent set of open, and internationally accepted ICT standards (Mason 2007; Ray & Jones 2006).

2.2.3.1. Product Data Standards

Data standards are fundamental to exchange information data effectively among contractors, subcontractors, customers and suppliers, regardless of the applications and software used by each one of them. They provide a means for enabling and facilitating compatibility between systems and sub-systems, supporting interoperability between models, as well as enterprise collaboration throughout the product life cycle.

Specific data standards are available for industrial use, namely the ISO's standard for the exchange of product model data (STEP) (ISO TC184/SC4, 1994). With more than forty standard Application Protocols (APs) for product data representation, reflecting the consolidated expertise of major industrial worldwide specialists working together for more than twenty years, STEP covers the main product data management areas for main industries, thus having a distinct advantage over similar technologies (Jardim-Goncalves, Figay, et al., 2006; ISO TC184/SC4, 1994). The objective of ISO 10303 (STEP) is to provide means of describing computer-interpretable product data throughout the life cycle of a product, independently from any particular system. Hence, the nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing product databases and for archiving data (Pratt, 2001). The standard is many times embedded within computer software associated with particular engineering applications, thus transparent to an end user.

However, the acceptance of STEP technologies has been facing difficulties when applied to sectors primarily composed by SMEs. The main problem is that this technology is unfamiliar to most application developers, requiring a wider tool support, and SMEs don't have the resources that larger companies do to hire or educate specialised personnel (Farinha et al., 2007; Jardim-Goncalves, Agostinho, et al., 2007; Lubell et al., 2004; Krause & Kaufmann, 2007). APs are described using ISO 10303-11, most commonly known as EXPRESS, which is the STEP modelling language (ISO TC184/SC4, 2004). It combines modelling methods from the entity-attribute-relationship family of modelling languages with object modelling concepts. EXPRESS provides general and powerful mechanisms for representation of inheritance among the entities constituting the AP standard model, and it also encloses a full procedural programming language used to specify constraints on populations of instances. Nevertheless it is relatively unknown, especially since the increasing usage of open technologies from the web community (e.g. W3C[21]) which are more intuitive and tool supported. In the same framework, additional advantages may result from the use of completely open standards, such as an easier integration of multiple data formats, applicability to a wider range of domains and a more efficient contribution to design by specialist departments.

[21] W3C – World Wide Web Consortium (www.w3c.org/)

2.2.3.2. Modelling standards and frameworks

Among the huge panoply of standards, not only the product data standards can be of relevance for enterprise interoperability, but also plain data encoding standards as the XML language, communication protocols as TCP/IP[22], etc. More importantly, modelling standards and frameworks can provide a valuable contribute by regulating the way developers should use available technologies to model systems, software, and interoperability solutions addressing all levels of the enterprise, from the business/organisational view, through the processes, services and data itself as envisaged in the holistic framework for EI.

Concerning EI, one of the most important standards available is the new ISO 11354 (ISO TC184/SC5, 2011) that proposes a framework for enterprise interoperability (explained before, on section 2.1.4.4 "Holistic Framework"). Nevertheless others closely related to EI exist, e.g. UEML[23] a *defacto* method for enterprise modelling, OMG's MDA, ISO/IEC 11179 (ISO/IEC, 1999) ruling the specification and standardization of data elements, among others. The next figure illustrates a condensed view on some of the interoperability related standards (the yellow ones are considered the most relevant and complementary to ISO 11354), as proposed by Zelm & Kosanke (2007).

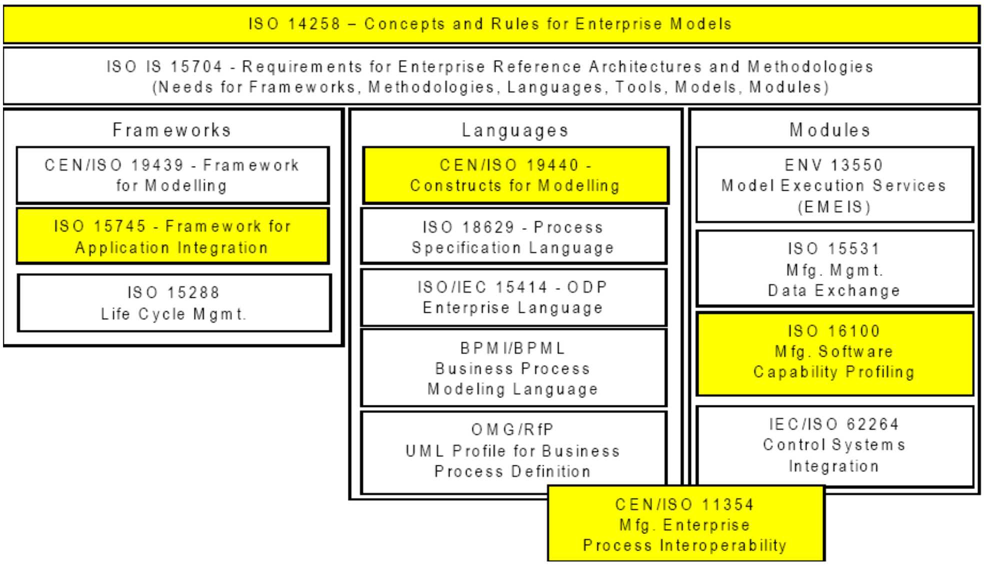


Figure 2.6: EI related standards (Zelm & Kosanke, 2007)

2.3. Model-Driven Engineering

Model-Driven Engineering (MDE), sometimes also referred as Model-Driven Development (MDD), is an emerging practice for developing model driven applications. It represents a promising software engineering approach to address systems complexity, both by simplifying and formalising the various activities and tasks that comprise an information system life cycle (i.e. from design, to construction, deployment, operation, maintenance and modification). Given today's increase of technology complexity, models are becoming more and more attractive as a powerful mechanism

[22] Transmission Control Protocol
[23] Unified Enterprise Modelling Language (http://www.cimosa.de/Modelling/UEML02.html)

to precisely describe problems in a way that avoids delving into technological details, thus allowing developers to focus on more abstract tasks and increasing productivity rather than to computing concepts. MDE is meant to maximize compatibility between systems, simplifying the process of design, and promoting communication between individuals and teams working on the system (Selic, 2003).

In MDD's vision, the primary artefacts are models. When compared to the most popular programming languages, models are easier to specify, verify, understand and maintain, thus widening the creation of new systems to domain experts, instead of only computing specialists. One key premise behind MDD is that code can be automatically generated from these models, elevating, the level of abstraction at which developers operate, reducing both the amount of development effort and the complexity of the software artefacts (Hailpern & Tarr, 2006; Van Gorp, 2008). Since the past two decades, the level of software abstraction has been raising, for example, by using expressive object-oriented languages (JAVA, C#, C++, etc), rather than less abstract one (Fortran or C) (Schmidt, 2006).

MDD/MDE's vision goes even further, invoking the unification principle, which states that *"everything is a model"* (i.e., platforms, components, legacy software, services, etc, are all models), similar to the basic principle in object technology stating that *"everything is an object"* which was most helpful in driving technology towards simplicity, generality and power of integration (Bézivin, 2005). Another key prerogative in MDE is the support of models at different levels of abstraction, from high-level business models focusing on goals, roles and responsibilities down to detailed use-case and scenario models for business execution (Frankel, 2003). These models are developed through extensive communication among product managers, designers, and members of the development team, and as they approach completion, enable a fast development of product and systems. However, despite obvious potential capabilities for closely matching the EI holistic levels, yielding major productivity and reliability benefits, there is not consensus about its technology readiness (Ambler, 2003; Czarnecki & Helsen, 2006).

2.3.1. Model-Driven Architecture

Among the several realisations of the MDE/MDD principles that exist, such as Agile Model Driven Development (Ambler, 2003; Ambler, 2011), Domain-oriented Programming (Thomas & Barry, 2003), Microsoft's Software Factories (Greenfield et al., 2004) and Model Driven Architecture (OMG, 2003; OMG, 2011a), MDA is perhaps the most prevalent at the moment. It is an OMG initiative, presenting a new vision of how application systems should be developed and managed, based on the MDD principles. Since it was launched, in 2001, MDA has been having a major impact on the software development community, and presently, there is a large landscape of tools available for its support.

MDA has as its foundation on three complementary ideas: direct representation, automation and open standards. The first makes use of abstract models to represent ideas and concepts of the problem domain, reducing the semantic gap existing between domain-specific concepts and the technologies used to implement them. The second uses model transformation tools to automate

the translation process from high level specifications and formal descriptions of the systems, to the bottom levels and implementation code, therefore increasing speed, code optimisation and avoiding human errors in the process. Regarding the last foundation, MDA enforces the usage of open standards to specify the high level models, and the features of the target implementation platforms. In addition, the usage of standards promotes interoperability among the entire ecosystem of tool vendors addressing its many different aspects (Delgado, 2008; Frankel, 2003).

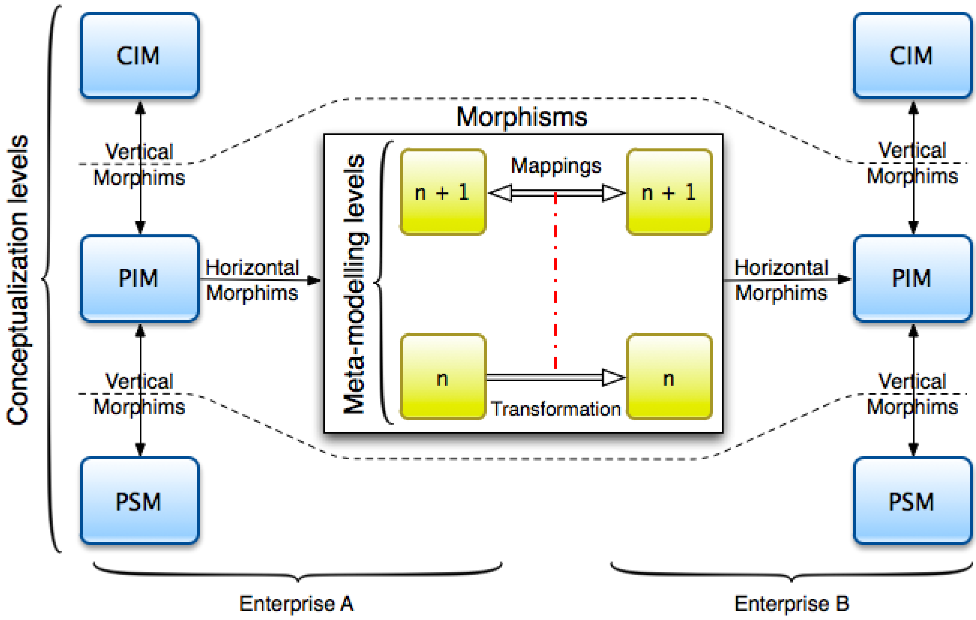


Figure 2.7: MDA's Conceptualization Levels and Transformation Types

An MDA system can be observed and analysed from different points of view (see Figure 2.7), and in order to support the supra-cited foundations it defines a hierarchy of models at three different levels of information abstraction (Berre et al. 2009; Grangel et al. 2008; OMG 2003):

- *Computation Independent Model (CIM)*, which specifies the requirements for the system and the environment where it will operate. It's also called a domain model since it's meant for the domain practitioners and it's based on the vocabulary of the specific target domain. In this sense, it's useful, not only as an aid to understand a problem, but also it plays an important role in bridging the gap between domain experts and the development experts that will build the system and/or services;
- *Platform Independent Model (PIM)*, which is the formal specification of the structure and functionality of the system that abstracts away technical details. More concretely, it focuses on the operation details while hiding specific details of any particular platform in order to be suitable for use with several different platforms. As an analogy, a PIM can be viewed as a model intended to be executed in a virtual machine (e.g. Java Virtual Machine) defined independently of any specific platform and which are realised in platform-specific ways on different platforms;
- *Platform Specific Model (PSM)* that combines the specification in the PIM model with details that specify how the system uses a particular type of platform. For instance, in the

MDA, the term platform is used to refer to technological and engineering details that are irrelevant to the fundamental functionality of a software component. Thus, a PSM adds to the PIM, technical details and implementation constructs that are available in a specific implementation platform, including middleware, operating systems and programming languages (e.g. Java, C++, EJB, CORBA, XML, Web Services, etc).

While CIM specifies the requirements, both PIM and PSM specify respectively the system design and its implementation. However, both must follow strictly the definitions of CIM (Asnina, 2009).

Based on MOF enabled transformations, the MDA unifies every step of the development of an application or integrated suite from its start as a CIM of the application's business requirements through PIM defined functions and behaviour, one or more PSMs, to generated code and a deployable application. The PIM remains stable as technology evolves, extending and thereby maximizing software return on investment. Portability and interoperability are built into the MDA architecture, which also introduces the distinction between vertical and horizontal transformations (evidenced in Figure 2.7). The earlier implies a change on the abstraction level of the resulting model, e.g. going from PSM to PIM implies a generalisation transformation, and from PIM to PSM implies a specialisation transformation. In the case of the horizontal transformation (e.g. refactoring of individual models, language translation, or even joining different models), the level of abstraction remains unchanged, leading to solutions for interoperability problems at the same enterprise level (INTEROP Partners, 2005b).

Both input and output models must be an instance of a well-defined meta-model, and have to be classifiable according to the meta-modelling level they belong to (see section 2.2.1 "Models and Meta-Models"). Due to that, greater interoperability benefits but also harder complications are expected in horizontal transformations, since at the time of the transformation specification (mapping), one has to be concerned with different language-related specificities. In fact, different languages might enable to describe the same objects with different detail levels (e.g. properties, constraints, etc.).

2.3.1.1. Horizontal transformations

Horizontal transformations are a current practice to enable interoperability among two organisations. With them, companies can specify P2P mappings to translate any data from one format to 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 Figure 2.7, the idea is that when performing a transformation 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, e.g. ATL[24] (Eclipse foundation, 2011) and the QVT[25] (OMG, 2008) (see section 2.4.2 "Executable Transformation Languages"). This is valid either for CIM, PIM or PSM models.

---

[24] ATL – Atlas Transformation Language (www.eclipse.org/m2m/atl/)

This type of transformations are normally static processes that once defined can be repeated any number of times achieving the same results.

2.3.1.2. Vertical transformations

As explained before, MDA specifies a standard-based architecture for models, organised around three different abstraction levels that can be created based on vertical transformations, for example specialisations as in CIM to PIM or PIM to PSM. However, since the details of each modelling level are different, to perform such transformations, it is necessary to define a consistent set of rules that provide extra meaning to components from one abstraction level to another, and ultimately into executable code. Thus, most of the MDA vertical transformation tools provide a mechanism to perform model annotations and means to customize the transformation rules according to the user need at the different abstraction levels (Figure 2.8). Additionally, they provide pre-defined PSMs, along with their respective annotation stereotypes for the most commonly used programming languages (e.g. Java, C#, etc.).

The amount of generated code depends on both the code generator and also the level of detail represented in the PSMs (i.e. how well the PSM captures the details of the physical platform). Ideally, only small portions of missing code should have to be added by the human developer in order to ensure that the generated code and auxiliary files are ready for compilation, linking and deployment.

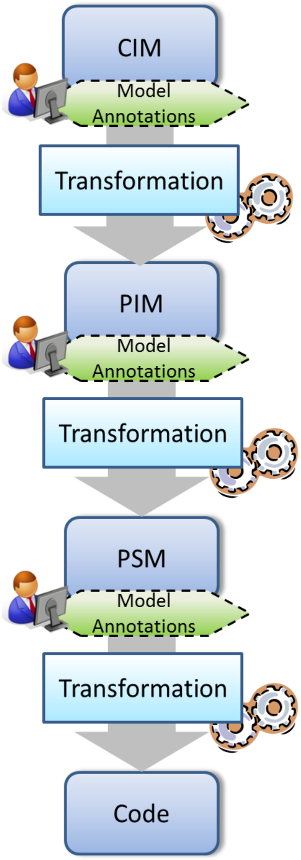


Figure 2.8: Vertical Transformation Process

To support this process, the OMG Unified Modeling Language (UML) already provides annotation constructs that largely enhance the PIM to PSM and PSM to code transformations. Being the core standard used to develop PIM and PSM models, UML models can be represented both textually and graphically (OMG, 2011c). Besides its powerful modelling mechanisms, it has other features that are essential in an MDA environment, such as extension mechanisms – the UML Profiles. These are packaged stereotypes, which extend and annotate the UML language to accommodate new constraints, syntactic elements, or even to restrict it, thus enabling to add specific knowledge to existing models. This mechanism allows transformations to become more efficient using more complete mappings, or even to define a new language without the need of creating it from scratch (Fuentes-Fernández & Vallecillo-Moreno, 2004). Once the profile is well defined, and PIM or PSM models annotated, an executable transformation language can be applied to it and achieve morphism automation from a model conforming to the defined stereotyped meta-model.

However, today, MDI vertical transformations are still an open issue. Incomplete applications have been developed from CIM to PIM due to the lack of efficient tools and methods for transformation (MSEE Partners, 2012). For this purpose, new concepts, methods and tools are necessary. Among

---

[25] QVT – Query View Transformation (www.omg.org/spec/QVT/)

the confirmed enterprise modelling methodologies which can bring interesting concepts, we can highlight GRAI (Doumeingts et al., 2006), 'House of ARIS'[26], or Leapfrog[27], but all need to be provided a suitable service engineering reference model to properly annotate CIM business models and enable vertical transformations to PIM. When starting from the same models, interoperability among systems developed using this method is seamless.

2.3.2. Model-Driven Interoperability

Model-Driven Interoperability (MDI) method is another MDA-based approach. It was introduced by the ATHENA and INTEROP research projects to solve interoperability problems between enterprises not only at the application and software systems level, but also at the business level with an ontological support (Athena Partners, 2007; Elvesæter et al., 2005; INTEROP Partners, 2006c). As the MDE/MDD for the software development, this method aims at improving the enterprises performances on communications, and it is supported by the MDI conceptual framework through the extensive use of models in vertical and horizontal integration of the multiple abstraction levels defined (Berre et al. 2009; INTEROP NoE 2007). This method, as detailed in Figure 2.9, introduces different conceptualization levels to reduce the gap between enterprises models and code level during the model transformation of MDD and MDA sub-domains, and uses a common ontology to support the transformations and to solve semantic interoperability.

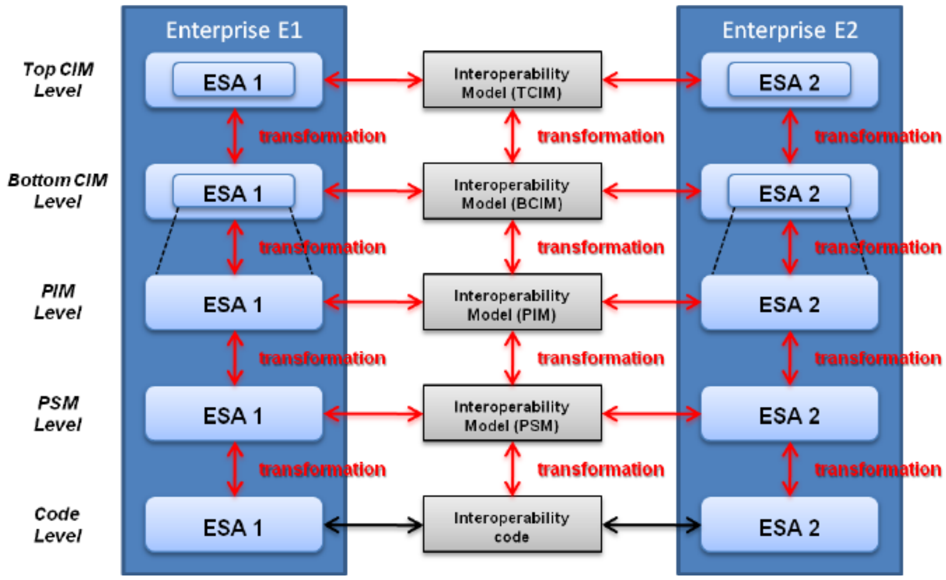


Figure 2.9: Reference Model for MDI (INTEROP Partners, 2007a)

The definition of the several levels in the reference model for MDI is based on the MDA, which defines the three classic levels, CIM, PIM, PSM but divides the CIM level into two sub-levels, the Top CIM Level (TCIM) and the Bottom CIM Level (BCIM). This reduces the gap between the CIM and PIM:

[26] http://www.ids-scheer.com/international/en
[27] http://www.leapfrog-eu.org

- *Top CIM* is used to represent a company from the "holistic" point of view, i.e., its domain, business strategy, etc, on a high level of abstraction without any detail of the software applications features;
- *Bottom CIM* is the representation of the Top CIM, since it needs to be implemented on some computer system, but without linking it to any kind of technology or implementation in specific.

While the main objective of MDA is to separate the functional specifications of a system from the implementation details related to a specific platform, MDI method's objective is to start at the highest level of abstraction and derive solutions from successive transformations, instead of solving the interoperability only at the code level. Therefore, the interoperability model has been defined at the various abstraction levels, solving the horizontal interoperability problem between enterprises.

To emphasise the systems engineering perspective of MDI, Figure 2.10 maps the three MDI/MDA modelling levels to system life cycle phases. Note that the design phase is concerned with the three modelling levels but each level has a particular focus: CIM model aims at representing end-user requirements and mainly deals with preliminary and end-user oriented design; PIM model supports technical and detail design; and PSM model focus and implementation/realization oriented design.

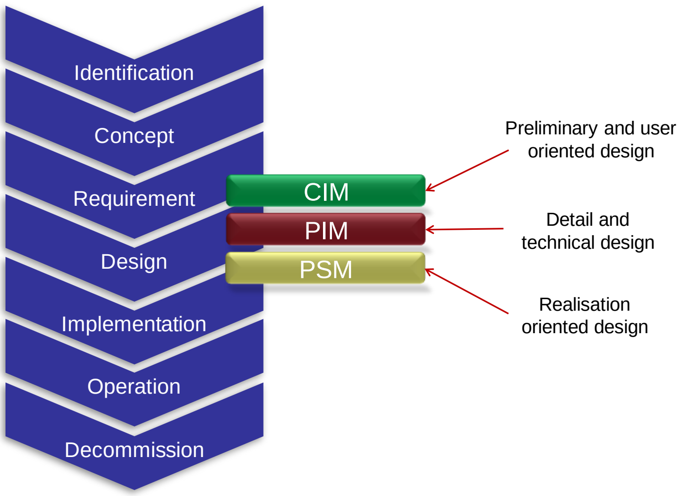


Figure 2.10: System Lifecycle and Modelling Levels (adapted from MSEE IP (2012))

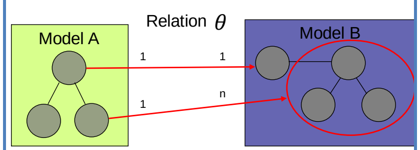
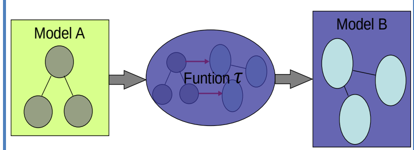
2.4. Model Morphisms (MoMo)

In mathematics, a "Morphism" is an abstraction of a structure-preserving map between two mathematical structures. It can be seen as a function in set theory, the connection between domain and co-domain in category theory, or the edge that connects two nodes within a graph. Recently, this concept as been gaining momentum applied to computer science, namely to systems interoperability (INTEROP Partners, 2005b). In this new usage, "morphism" specifies the relation (e.g. mapping, merging, transformation, etc) between two or more information model specifications (*M* as the set of models). Indeed, according to Agostinho et al. (2007), MoMo is now used to

describe a model operation as the ones addressed in MDA/MDI (Agostinho, Sarraipa, et al., 2007) and can be classified within two classes of operation types (Table 2.2):

- *Model non-altering morphisms*, which are based on the concept of traditional model-mappings, i.e. given two existing models (source *A* and target *B*), a mapping θ is created relating each element of the source with a correspondent element in the target, leaving both models intact. These mappings define the space of all the relations (1-to-1, 1-to-n, or m-to-n), and the process of constructing the mapping, called "mapping discovery", should find semantically related elements in both models;
- *Model altering morphisms*, where the source model is transformed using a function that applies a mapping to the source model and outputs a target model (Delgado et al., 2006). The main objective behind model transformation consists in transforming a source model *A* into a target model *B*, by means of modifying the first one by a function τ. This function can be expressed by a simple table, relating multiple or single constructs from both meta-models but once it is created should be later implemented by using more formal and executable languages (such as ATL). There are several techniques for achieving model transformations at various levels, such as the top level "model-to-model" and "model-to-text" techniques (Czarnecki & Helsen, 2006; Van Gorp, 2008). Other relations, such as the merge operation, can also be classified as model altering morphisms, however they are not here detailed (refer to INTEROP NoE (2005b) for more information).

Table 2.2: Classes of Model Morphisms

	MoMo	Formalisation	Classification	Graphical Representation
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	

MoMo is an attempt to formalise model based operations, however, 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. Technologies and formalisation methods will be analysed hereafter concerning their usability towards that goal and as a support for describing morphisms in the future EI science.

2.4.1. Suitable MoMo Formalisation Techniques

As introduced by INTEROP NoE (2005b), graph theory has been originally used as a basis for MoMo formalisation, but other theories can be considered to achieve the envisaged goals.

2.4.1.1. *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 2.2 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 (Delgado, 2008; INTEROP Partners, 2005b). This abstract view of models allows formal reasoning on their properties and on the ones 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 (Dauben, 1979). 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 and can be used to express model relationships.

2.4.1.2. *Mapping as a model: Model Management* (Bernstein, 2003)

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 $M2$). Thus, each object "m" in the mapping can relate to a set of objects in $M1$ and 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 Figure 2.11).

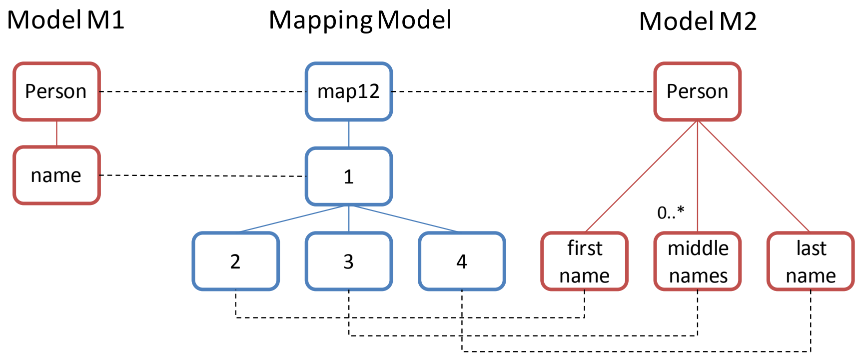


Figure 2.11: Mapping as a Model "map12"

Using concepts from classical mathematics, this approach enables to define complex algebra to describe model operations such as *match*[28], *compose*[29], *diff*[30], *model gen*[31], or *merge*[32].

[28] *Match* - takes two models as input and returns a mapping between them.
[29] *Compose* - takes a mapping between models A and B and a mapping between models B and C, and returns a mapping between A and C.
[30] *Diff* - takes a model A and mapping between A and some model B, and returns the sub-model of A that does not participate in the mapping.
[31] *ModelGen* – takes a model A, and returns a new model B based on A and a mapping between A and B.
[32] *Merge* – takes two models A and B and a mapping between them, and returns the union C of A and B along with mappings between C and A, and C and B.

2.4.1.3. Mapping as a complex tuple: Matching (Giunchiglia et al., 2007)

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 (Blaha & Premerlani, 1996), and semantic matching (analysed under section 2.6.2 "Semantic Mismatches").

2.4.2. Executable Transformation Languages

To complement the formalisation of the morphisms with a practical dimension, a number of transformation languages with support for automatic model transformation execution can be used, e.g. ATL and the QVT (Eclipse foundation, 2011; OMG, 2008).

QVT is a hybrid declarative and imperative transformation language that defines a standard way to transform source models into target models, which is sustained by the four levels of OMG's meta-modelling architecture. It also supports bidirectional model to model (horizontal) transformations conforming to any MOF 2.0 meta-model. This means that model to text, whatever the text is (XML, Code, etc.), or vice-versa, is simply not supported. On the other hand, ATL is not so rigid and enables association with other methods to accomplish that. The main difference between them is that it can only be used to do unidirectional syntactic and semantic translation. An ATL 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").

The above languages are based on the Object Constraint Language (OCL) (OMG, 2010a), but others exist: Xtend/Xpand[33] which is a JAVA-alike language, UMLX, AToM3, MTL, just to enumerate some (Czarnecki & Helsen, 2006). ATL is one of the most used transformation languages, having a large user base and being very well documented as well as a good JAVA integration, nevertheless it is neither a standard nor a simple language to use (Bézivin et al., 2003).

[33] Xpand Language (www.eclipse.org/modeling/m2t/)

2.4.3. MoMo recommendation system

To describe unambiguously any tool that implements model morphisms, a reference ontology designated by MoMo ontology was designed by D'Antonio et al. (2006) (see Figure 2.12 [34]). Using it, becomes possible to describe interoperability solutions related to model processing operations, and when properly instantiated, the ontology will provide a valuable knowledge-base for the INTEROP NoE (2007b) MRS (MoMo Recommendation System) to reason and make decisions and suggestions. Indeed, the MRS is able to assist any user in the resolution of mapping/transformation problems, by analysing the ontology instances and recommending the most appropriate computational method(s) or tool(s) suitable for specific model morphism tasks (Agostinho, Dutra, et al., 2007).

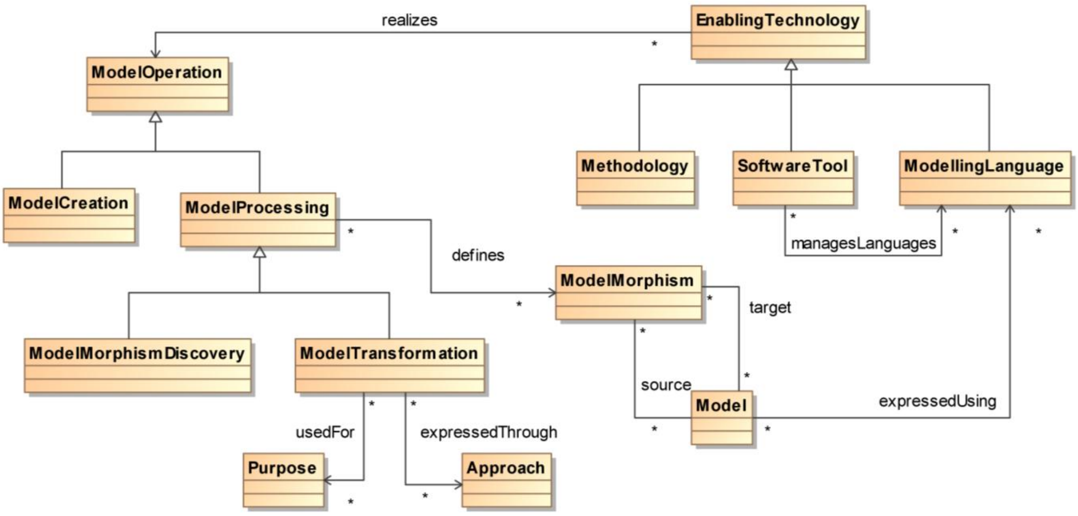


Figure 2.12: MoMo Ontology

The ontology presents "ModelMorphism" as a generic concept that defines a "ModelTransformation" or a "ModelMorphismDiscovery" which are not here concretized but could be formalised using any/a set of the techniques presented before. A possible example of the complete cycle of a very simple "MoMo recommendation" process may be the following:

1. A user finds a demand for a MoMo, e.g. needs to rename all the occurrences of a specific entity in an OWL ontology (is information system repository);
2. Formalises the query to the MRS;
3. Receives the recommendation of which tools to use, e.g. the combined use of two different morphism tools that will perform the task.

2.5. Cloud and Service-Oriented Interoperability

2.5.1. Service Oriented Architectures (SOA)

The definition of the service oriented interoperability concept is directly connected with the one of service oriented architectures, namely with the planning and customization of services to achieve

[34] UML notation was used to increase readability

software interoperability. In software, architectures perceive the building elements and structures as "black boxes" used through a known interface that hides the implementation details. This concept has been growing in complexity, evolving from functions, to classes, components, and web services thus enabling the development of systems with greater functionality in less time by reusing existing assets (Baïna et al., 2004; Gorton, 2006; Papazoglou, 2003). In fact, today, it is possible to take advantage of any kind of “black box” technology for systems development. Nonetheless, if one needs to integrate heterogeneous systems from different companies, it needs to understand the way each system is accessed and the encoding of the messages in order to build proper wrappers for the integration. For that, service to service interoperability is required, isolating the application logic from the communication details and relying on services, namely SOA (Athena, 2005).

SOA combines the capability of invoking remote objects and functions (services) using specific tools for dynamic service discovery as if they were local, placing emphasis on interoperability. Therefore, it uses the “black box” concept applied to remote services, isolating the developer from differences in operating systems, implementation languages, and network protocols used between the various hardware and software components. Clients (e.g. developers) look for specific services in a registry, which points to previously published provider services (Papazoglou, 2003; W3C, 2004).

Today, services and SOA are really about designing and building systems using heterogeneous network addressable software components, e.g. agents and web services (WS), which have given the term service more importance in the last years thanks to the support from the software community (Athena, 2005). They are designed to support interoperable machine-to-machine interaction over a network, describing client and provider interfaces in a machine-processable format (WSDL[35]) and enabling information exchange using SOAP[36] messages, typically conveyed over the WWW with an XML serialization (W3C, 2004). WS have well defined standards to describe the services, manage the security and transactions, and represent the requests and responses. However, standards do not define the "middleware" for SOA, i.e. the WS standards support a simple P2P model but in a more widespread environment, a many-to-many scenario is likely to occur. Thus, the idea of “Grid services” appeared as a new paradigm of services that envisages the flexible and secure integration of multiple resources among dynamic sets of individuals, applications and organisations (Juric et al., 2007).

Grid computing has developed addressing the need of high performance computing in distributed systems, reusing underutilized hardware resources and parallel processing (Foster et al., 2002). Grid and Web services share a great deal of requirements and specifications, such as interoperability, service discovery, etc. Thus, the concept of Grid based e-business appeared sharing the essential goals with SOA, although the specific focus rests on integration of existing services to build up new high performance functionalities, enhanced with business specific capabilities (Schubert et al., 2010).

---

[35] WSDL - Web Service Definition Language (www.w3.org/TR/wsdl)
[36] SOAP - Simple Object Access Protocol (www.w3.org/TR/soap/)

2.5.2. Cloud Computing and Interoperability of Clouds

Though the concept of "clouds" is the youngest among the ones discussed above, it is undisputable that it has proven to be a major commercial hit over recent years and will play a large part in the ICT domain in the future, as systems will further exploit the capabilities of managed services and resource provisioning (Schubert et al., 2010). A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers dynamically provisioned and presented as one or more unified computing resource(s), which are linked using service-level agreements established through negotiation between the service provider and consumers (Buyya et al., 2009).

One might think clouds are just a combination of classical resource service provisioning and Grids. However, this is not the case. Cloud computing is a general concept that incorporates trends in technologies, and relies on the Internet for satisfying the computing and networking needs of the users and customers (Weiss, 2007). Cloud platforms possess characteristics of both WS and Grid, namely SOA and hardware virtualization. With its own special attributes and capabilities, clouds support resources virtualization, dynamic composition of services using WS interfaces, high availability, and elasticity with the capability to add more capacity to a running system by deploying new instances of each component and shifting load to them. They are infinitely scalable and can be used for every purpose from disaster recovery to business continuity through a fully outsourced ICT service. This makes cloud computing very attractive to small companies (Schubert et al., 2010).

Within the cloud computing paradigm, there are some variations on what service is included. The most common reference is SaaS (Software as a Service) which is the capability to use the provider's applications running on a cloud infrastructure, but is also linked intimately with those of: IaaS (Infrastructure as a Service) to provide the user processing, storage, networks, and other fundamental computing resources in which there is the possibility to deploy and operate any software system, including operating systems and applications; PaaS (Platform as a Service) to deploy directly onto the cloud infrastructure of applications using development tools supported by the cloud provider; and collectively *aaS (Everything as a Service) (Vaquero et al., 2009).

It is foreseen that it will become easier to move data from one cloud computing system to another, with providers developing ecosystems that are able to seamlessly communicate and share data among clouds. In order to ease the use of cloud computing systems and enable the migration of applications between clouds of different providers, there will most likely be the need for standardized cloud APIs. Still, cloud computing can become a major facilitator for efficient EI as interoperability between companies' applications within the same cloud is assured by the usage of the same standards. Furthermore, as clusters of companies emerge in clouds, it is expected that different clouds need to become interoperable, and subsequently, so will companies from different clouds.

2.5.3. Internet of Services (IoS)

The Internet of Services is a vision where everything that is needed to use software applications is available as a service on the Internet, such as the software itself, the tools to develop the software,

and the platform (servers, storage and communication) to run the software (European Comission, 2011). Hence, all variations of cloud computing imply an IoS service-oriented architecture where companies can explore different aspects of the service sector to determine which services can be managed through ICT and, being combined with others, transformed into value-added services and business models (Juric et al., 2007; Schroth & Janner, 2007; SAP Research, 2009).

Service based application provisioning is part of the Future Internet (FI)[37] as such, and therefore a similar statement applies to cloud and Internet of Services as to cloud and Future Internet. In combination with Web 2.0 technologies, IoS is expected to improve service innovation. Additionally, by bringing events from the real world into the services realm, the Internet of Services will become a cornerstone for the Internet of the next generation - Web 3.0 (SAP Research, 2009).

2.6. Semantic-Based Interoperability

Being defined as the study of meaning, semantics are essential for systems interoperability (Merriam-Webster, 2011). In fact the second part of the IEEE definition of interoperability (IEEE, 1990) - *"use the information that has been exchanged"* - already suggests that semantics are important. As a negative example, consider two persons who do not share a common language: despite speaking to one another and recognizing that data is being transferred (probably they can also parse individual words and distinguish the beginning and end of messages due to pauses in the speech), the meaning of the message will be unintelligible (Komatsoulis et al., 2008). Moreover, if they were to write down the words spoken and listened, they would come up with different pieces of text. A similar situation would also happen even if they shared the language but not the vocabulary (e.g. technical, slang, etc.)

Semantic interoperability requires that any two systems derive the same inferences from the same information. It needs syntactic interoperability as a pre-requisite, so that the approach for processing the information will be interpretable from a known structure. However, once the syntactical correctness has been verified, semantic interpretation, which goes beyond syntax or structure, must be understood and unambiguously defined based on the context. The task of achieving semantic interoperability among IS requires the use of a means to assure that. Hence, if there is any context sensitivity to the way terms are used, it must also be specified as part of the information using those terms.

With this, in a broad definition, semantic interoperability is the ability to automatically, meaningfully, and accurately interpret the information exchanged in order to produce useful results as defined by the end users. To achieve semantic interoperability, an accepted approach is that both sides refer to a reference knowledge model (e.g. ontology) so that the content of the requests for information is unambiguously defined: what is sent is the same as what is understood (Jardim-Goncalves, Silva, et al. 2007; Missikoff et al. 2003; Sarraipa et al. 2010).

---

[37] www.future-internet.eu/

Semantics are recognised as an important area for EI alignment. Besides the CEN/ISSS levels of interoperability (CEN/ISSS, 2006) or the EIF (ISA, 2010), also the Athena Interoperability Framework (AIF) (Athena IP, 2006) identified them as one of the levels of interoperability to consider within an enterprise. Wenguang Wang et al. (2009) place semantic interoperability in the middle of the LCIM scale towards a seamless conceptual interoperability, where more than an agreement between all systems on a set of terms, companies need a shared understanding of the system's conceptual models. However, a general observation shows that these models, in their present state, are often not exploited to assess interoperation. Indeed, their semantics are, due to their heterogeneity (tools, methods), often implied by implementation requirements and not expressed as dependent on the business modelling (Yahia et al., 2012).

2.6.1. Semantic Heterogeneity of Conceptual Models

As analysed by Yahia (f2011), the heterogeneity of conceptual business models, verified at both syntactic (as studied in section 2.2.1 "Models and Meta-Models") and semantic levels, is caused by the differences in the interpretations of the real world by modellers, when designing information systems (J. Park & Ram, 2004). Moreover, the majority of these models have been developed by different experts, each with their own modelling experience. Therefore, the semantics conveyed by the different models is more or less explicit, and the process of conceptualization can produce several representations syntactically different but semantically equivalent. These differences relate to the context and use of data and generate semantic conflicts that can occur in patterns and/or data (schema-level, data-level).

The conflicts can be identified as semantic mismatches, leading to either *lossless* or *lossy* relationships depending on the properties of the related model elements (see next section): in lossless cases, the relating element can fully capture the semantics of the related and no interoperability problems are verified; while in lossy mismatches a semantic preserving mapping to the reference model cannot be built. This uncertainty motivates that semantic interoperability should be defined according to the property of non-symmetry, i.e. *"S1 can be interoperable with S2 without the latter being fully interoperable with S1*", thus tolerating *maximum potential interoperability* (S1→S2) and *minimum effective interoperability* (S2→S1) (Yahia et al., 2012).

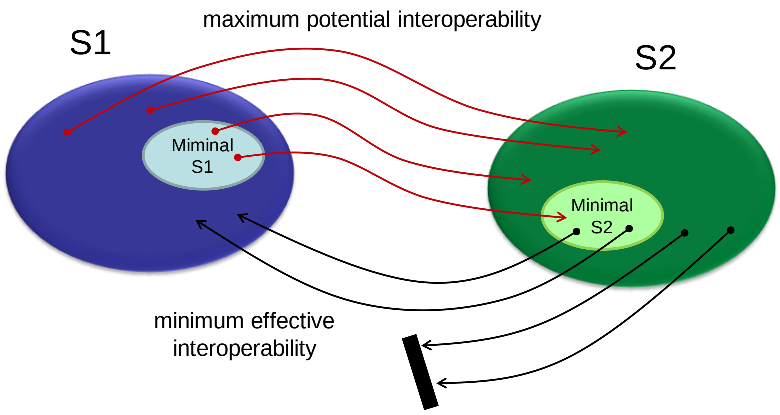


Figure 2.13: Non-symmetry of Semantic Interoperability (adapted from Yahia (2011))

2.6.2. Semantic Mismatches

Mismatches are inconsistencies of information that result from “imperfect” mappings. Due to the differences among models (e.g. language, terminology, granularity, etc), almost in every case, a MoMo towards achieving interoperability leads to a semantic mismatch (see Table 2.3).

Table 2.3: Semantic Mismatches (adapted from INTEROP NoE (2006a))

Mismatch		Description	Examples	
			System 1	System 2
Lossless	Naming	Different labels for same concept of structure	Person -name : string (Naming)	Person -personName : string
Lossless	Granularity	Same information decomposed in or composed by (sub)attributes	Person -name : string (Granularity)	Person -firstName : string -lastName : string
Lossless	Structuring	Different design structures for the same information	Person -name : string (Structuring)	Person -hasId Indentification -name : string
Lossless	SubClass-Attribute	An attribute, with a predefined value set (enumeration) represented by a subclass hierarchy (or vice-versa)	Person -name «enumeration» GivenNames +Jose +Carlos (SubClass-Attribute)	Person Jose Carlos
Lossless	Schema-Instance	An attribute value in one model can be a part of the other’s model schema (or vice-versa). In the example on the right, “Jose” (schema) could be the “name” of a “Person” (instance)	Person -name : string -weight : int (Schema-Instance)	Jose -weight : int
Lossless	Encoding	Different formats of data or units of measure for the same attribute	Person -weight_Kg : int (Encoding)	Person -weight_Lbs : int
Lossy	Content	Different content denoted by the same concept. In the example on the right, “Person” has different information associated	Person -name : string (Content)	Person -address : string
Lossy	Coverage	Absence of information	Person -name : string -weight : int (Coverage)	Person -name : string
Lossy	Precision	Accuracy of information. In the example on the right, a predefined value set does not allow the same accuracy of values as the domain of an integer number.	Person -weight : int (Precision)	Person -weight «enumeration» WeightCategory +heavy +medium +light
Lossy	Abstraction	Level of specialisation, e.g. “Ford” is a specific type of “Car”	Car (Abstraction)	Ford

This notion of mismatch can bring a semantic meaning to the type of the relationship being established in the mapping and lossy mismatches should be avoided whenever possible.

2.6.3. Knowledge and Related Concepts

Finally, when speaking about semantics, it is important to clarify some knowledge related concepts that normally tend to raise some confusion, i.e. data, information, and knowledge:

- *Data* can exist in multiple ways, independently of being usable or not. In the raw format, it does not have meaning in and of itself.
- *Information* is data that has been given meaning by way of relational connection to a context (Breiter, 2004). Still, in information, this "meaning" can be useful for some, but not necessarily to all. Information embodies the understanding of a relationship of some sort, possibly cause and effect. Thus, people might "memorize" information (as less-aspiring students often do), but still be unable to understand it since it requires a cognitive and analytical ability, i.e. knowledge (Bellinger et al., 2004).
- *Knowledge*, which as defined by Nonaka et al. (2001) can be sub-divided into: 1) Tacit knowledge: that people carry in their minds, which provides context for people, places, ideas, and experiences; and 2) Explicit knowledge: that has been or can be articulated, codified, and stored in certain media. In an ideal semantic based interoperability framework, both should be addressed and processable in order to achieve more advances stages of knowledge, such as understanding and wisdom (Bellinger et al., 2004; Jardim-Goncalves et al., 2009; Syed & Shah, 2001).

The major research challenge nowadays is to gather the tacit knowledge domain stakeholders hold, in interpretable knowledge bases, thus transforming it to explicit knowledge stored in a structured organised way. This way, syntax and lexical semantics would be integrated and automatically processable (Pustejovsky, 1993). For reaching that purpose, literature suggests the usage of several knowledge representation technologies such as (Figure 2.14):

- *Dictionaries* - whether in the form of natural language, technical, or domain, they have been found to lessen a great deal of miscommunication. Dictionaries provide users with information about terms and abbreviations available in a central location, where definitions are used differently or in a very specific way within the domain;

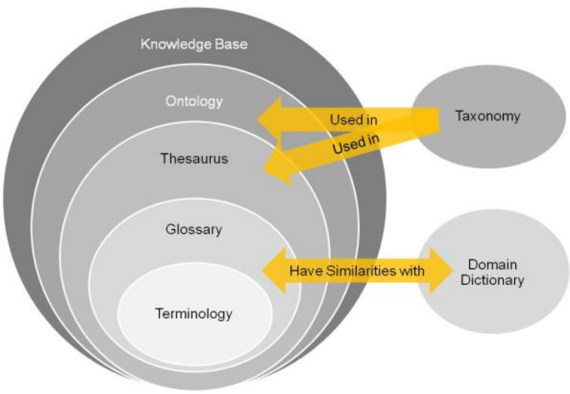


Figure 2.14: Knowledge Base (Sarraipa 2009)

- *Glossaries* – a list (mostly in alphabetic order) of specialised terms, each of which, composed by its corresponding description. A glossary can be used when communicating information in order to unify knowledge sharing;
- *Taxonomies* - a classification system that categorizes all the information in a class/subclass relationship, representing a simple tree where each child is a subset of the parent concept;
- *Thesaurus* – whose main objective is to establish a formal lexicon of a specific domain. A thesaurus is a structure that manages the complexities of terminology and provides conceptual relationships like a taxonomy of domain concepts composed by its meanings;
- *Ontologies* – already introduced in section 2.2.2 “Ontologies and Folksonomies”, uses most of the above technologies to define a common language for sharing/reusing knowledge about phenomena in a domain;
- And *knowledge bases* – which are the instantiation of the ontology with knowledge expressed using the formal knowledge representation language.

2.7. Sustainability of Systems and Interoperability

The concept of “sustainability” is tightly connected with improving the quality of human life while living within the carrying capacity of the supporting eco-systems (Munro, 1991), but it is not exclusive to it. As explored in this dissertation, it can be associated to ICT and other domains that ultimately can create an impact on the quality of being.

That association was made more evident only at the 2005 UN World Summit (United Nations General Assembly, 2005), where it was noted that sustainability requires the reconciliation of environmental, social and economic demands, thus forming the "three pillars" of sustainability (Figure 2.15), which are not mutually exclusive but can be mutually reinforcing. This strategy has many comprehensive intervention areas that cross-cut domains from biology to engineering and manufacturing, e.g. ozone layer, bio-diversity, renewable materials, energy efficiency, health, employment, etc. With the proliferation of the term associated to Earth’s preservation, soon it became common also in industrial domains. In fact, policy makers have constantly been encouraging industrial organisations to reconcile economic interests with social and environmental needs (Adams, 2006).

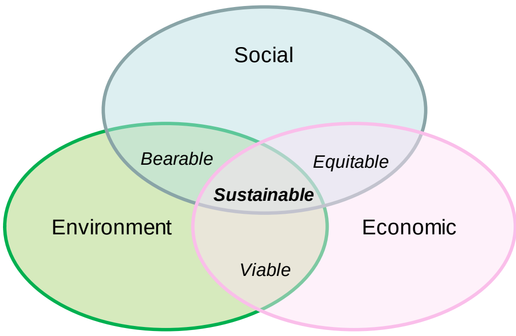


Figure 2.15: Three Pillars of Sustainability

2.7.1. Sustainable manufacturing

Indeed, following that strategy, sustainable manufacturing is nowadays one of the key areas in systems research, supporting the creation and distribution of innovative products while minimizing resources consumption (inputs such as materials, energy, water, and land), eliminating toxic substances, and producing zero waste across the entire lifecycle of products and services (NIST, 2011). In this framework, sustainability is envisaging the development of new infrastructures to support information models, standards, metrics, tools, and interoperability.

The Model Based Enterprise (MBE) project was one of the first initiatives to target this strategy in the ICT domain (MBE project, 2009). It provides a fully integrated and collaborative environment grounded on 3D product definition shared across the enterprise; to enable rapid, seamless, and affordable deployment of products from concept to disposal. MBE is a systems engineering process of reusing 3D CAD (Computer-Aided Design) models by all downstream customers, instead of recreating or re-entering data. Models contain all of the information needed to define the product, in an annotated and organised manner, that enable automatic extraction of information by non-CAD users, thus replacing traditional paper drawings and producing not only an economic impact, but also a social and environmental one.

2.7.2. Sustainable ICT

Apart from manufacturing domains, other sectors, as well as research, are becoming concerned with the sustainability of their work, even if not yet crossing the three pillars directly, e.g.:

- Research fields like textual scholarship, literary and documentary research have seen their tools and data shift more and more towards a digital environment in the last decades. This has revolutionised the possibilities of collaboration, sharing and analysis. However, technical approaches until now have been focused on the collection of data and are presenting problems for sharing and self-sustaining tools and data (Interedition, 2008).
- A similar case is happening in the natural language processing community, where a substantial effort has been put in the creation of computational lexicons and compendia of semantic information (e.g., framenets, ontologies, knowledge bases) together with *language corpora* annotated for all varieties of linguistic features. However, the lack of a thorough, well-articulated longer-term vision has engendered the creation of a disjointed set of language resources and tools, which are simply not interoperable (Ide et al., 2009).
- On the Web domain (services), a problematic case is found on WS, and more specifically mashups, where the relevant policy issue is not how to establish interoperability (which exists by definition), but rather how to ensure that this level of interoperability is maintained so that the current rate of innovation continues. Currently, survival of mashups depends not only on interoperable systems but also on the continued willingness of data providers to give meaning and utility to those interoperable systems (Gasser & Palfrey, 2007).

Apparently these examples from the ICT industry might not address directly the sustainability targets as envisaged in Adams (2006) and United Nations General Assembly (2005), but the

advances in these domains have a strong economic and social impact due to the influence ICT has in the modern society. The ICT network helps businesses foster new relationships that are essential to growth and overall productivity, while taking costs out of traditional business operations to free resources and support further innovation. Environmental impact is less visible depending on the areas, but definitely they can be related, reducing the paper waste, energy consumption (e.g. simulation instead of prototyping) and more (Sorensen, 2009).

2.7.3. Sustainable Interoperability

Technology is critical to the transition from the old economy (fossil fuel, automobile throw-away) to the new economy (reuse, recycle, new energy) (Bosin et al., 2005).

Today enterprise interoperability, 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. Research projects are still launched for analysing the domain, proposing solutions, and for making breakthrough research. However, all these attempts should be coordinated to comply with the visions of the EI roadmaps (Enterprise Interoperability Cluster, 2008; FInES Research Roadmap Task Force, 2010) and most importantly, rely on scientific foundations, axioms and theories that should be set, in order to homogenize the various approaches, validate research activities and guarantee scientific excellence based on accepted and valuable results of the past (ENSEMBLE CSA, 2011). In other words, today the domain of EI has matured through the actions of the past, and in order to evolve further, needs sustainability, i.e. to be build upon a science base, if not a science by itself, capable of providing solid grounds for dynamicity, as well as repeatability of processes and solutions in multi-domain networks. As in the ICT domain, EI can impact the three pillars of sustainability. A good example of this new concern to interoperability is the realization of the 1st workshop on Model Driven Interoperability for Sustainable Information Systems, held in conjunction with the Conference on Advanced Information Systems Engineering, in 2008 (InterOP-VLab, 2008).

"Self-sustainable interoperability" or "**sustainability of systems interoperability"**, are new concepts proposed in this thesis to addresses the above needs and concerns.

2.8. Background Analysis and Identified Challenges

As information systems in enterprises and organisations evolve and become more complex, the need for interoperable operation, automated data interchange and coordinated behaviour of large scale infrastructures becomes highly critical (Chen & Doumeingts 2004; INTEROP NoE 2006c; Ray & Jones 2006). In fact, lack of interoperability would disturb creation of markets and diminish innovation and competitiveness. Therefore, interoperability appears as a key enabler towards unlocking the full potential of organisations, processes and systems in the public and private sector. Since its inception by the DoD (1977), and through the years, interoperability tends to obtain a broader, all-inclusive scope of a repetitive, well organised, and automated feature of organisations.

Apart from being a technical issue, interoperability is also important at other enterprise levels, as business, organisational and semantic level, underlying the need for solutions that support not only the seamless cooperation among ICT systems, but also among organisational structures and people (Jardim-Goncalves, Grilo, et al., 2006). EI suggests that interoperation becomes a native capability throughout research development of focal areas, removing barriers to communication, fostering a new networked business culture, and transferring and applying the research results in industrial sectors.

Challenge 1: *Interoperability should be a "transparent" capability supporting businesses, assured by ICT services but not perceived to users, i.e. working in the background of enterprise systems.*

However, up to now, the principal tools for targeting EI have been the various standards, in particular product data standards, that seek to govern IS development and operation. The adoption of such standards has not been as widespread as desired, since they are usually linked with specific market sectors, application areas, or technology trends, thus having a limited time span or a static nature, and quite often different interpretations by technology vendors (Jardim-Goncalves & Steiger-Garcao, 2002; Jardim-Goncalves, Figay, et al., 2006).

Challenge 2: *Stakeholders should be able to implement the several standards required in their businesses, independently of the technology they use, and without prejudice to the network harmonization and the subsequent interoperability effect provided by the standard.*

Nevertheless, standardisation policy is a major global and regional tool in the EI context (e.g., ISO, CEN). Standards point out to be perfect and completely clear but they must be implemented by the market. To reach globalization objectives for EI, they must be submitted to robust feedback methods to receive input from implementers, interested communities and from the market itself, guaranteeing dynamic improvement and standards maintenance. In fact, the Informal Study Group on Value Proposition for Enterprise Interoperability (2008) advocates that *"disruptive innovation at the enterprise level needs to be matched by disruptive innovation for enterprise systems of the future"*. This also means that IS need to evolve and adapt to the market requirements and enterprise trends, and standards might even have to incorporate additional data, thus making versioning more common.

Challenge 3: *Information systems, reference data models and standards should be gradually updated to match disruptive enterprise and market innovation.*

It is highly likely that the Future Internet will give rise to *"new opportunities of creativity and innovation, enable new forms of participation, and further catalyse the formation of networked enterprises and communities that span the world, thereby ushering in a new generation of enterprise systems requiring a reappraisal of interoperability between those systems"* (Informal Study Group on Value Proposition for Enterprise Interoperability, 2008). In the past, it was said that EI was not achieved until the interaction could take place at the different layers of the enterprise

(Athena IP, 2007). Today, it is believed that EI will not be fully achieved until the benefits brought by the new technology paradigms are reaped, including the paradigms for FI.

Challenge 4: *Enterprise systems and networks need to support rapid evolution of technology and applications, and enable automatic or on-demand reprocessing, recompiling or fixing of components or processes.*

Thus, scientific methods to assess the sustainability, suitability, impact of evolutions and radical changes, as well as the extension of the adoption and relevance of standards in the EI domain must be rigidly adopted. Still, in spite of the research developed, no scientific foundations for EI have yet been established: a deficit recognised by the EI community, disabling the generalisation and reuse of the methods and tools developed, and preventing a self-sustainable interoperability (Enterprise Interoperability Cluster, 2008; FInES Research Roadmap Task Force, 2010).

Challenge 5: *EI needs a scientific foundation that assures repeatability of processes with similar results for similar problems and enables the evolution of ICT without prejudice to existing systems.*

2.8.1. Interoperability within Enterprise Networks

Global business networks are suffering from the delocalisation phenomena, with suppliers and manufacturers moving their production networks to countries with cheaper human efforts or skilled competences. There are an increasingly number of cultures, backgrounds, and systems within a network, which are becoming to be characterized by non-centralised decision making. This, increases the autonomy of hub organisations, enabling different rules and procedures for decision making within the same supply chain, but decreases the effectiveness in terms of integration and interoperability.

Challenge 6: *Enterprises need to be able to achieve interoperability with its partnering organisations, independently of the business, geographical and cultural requirements.*

Paraphrasing Steve Ray, former Division Chief at NIST *"Supply chains are plagued with uncertainty (…) Most goods are still handed-off through faxes, phone calls, paper documents, and a wide range of proprietary electronic exchanges"* (Watson, 2008). With this reality, and the associated expenses and inefficiencies, policy-making organisations have been encouraging an increasing focus of research and development (R&D) in the domain of interoperability, through multiple public and private funding. Investments in infrastructures to support global supply chains, including hardware and software standards, information languages and protocols, and financial accounting and clearing systems, must increasingly be made across sectorial and national boundaries (European Comission, 2006; Enterprise Interoperability Cluster, 2008; INTEROP Partners, 2006b; Mason, 2007; W. J. White et al., 2004).

Challenge 7: *Public and private organisations need to be motivated to continue reinforcing the funding for interoperability research and technology development.*

As presented, different solutions to assist in achieving interoperability have already been proposed and categorized following different scales. Analysing the available literature (summarized in this chapter), it is possible to identify common characteristics and to point out the strengths and weaknesses of each. By and large, interoperability is a broad and complex subject and the development of specific solutions is difficult. However, it is acknowledged that assessing the interoperability with well-chosen measures is essential to identify priorities to consider when establishing business networks and partnerships.

System's design and network interoperation is seen as a "black box", but concerning evaluation, and following the work of Yahia (2011), interoperability needs a more detailed and accurate "white box" analysis to see what is happening inside of the network and systems in terms of MoMo. Yet the existing maturity models and levels do not focus EI practices in terms of efficiency of the network.

Challenge 8: *EI needs a "white box" view of collaboration networks to support the analysis of model morphisms and enable the measurement and evaluation of the network efficiency.*

Similarly to what is suggested in the ISO standard 11354, by recognizing their position within an interoperability frame, companies are able to evolve their systems and the networks they belong to, following progressive transitions with holistic concerns. This network interoperability evaluation helps:

- (*a posteriori* measurement) To determine whether a specific business collaboration network, and the systems that composes it, should improve to newer and better interoperability approaches/solutions;
- (*a priori* simulation) Companies planning to set up collaboration partnerships to decide, depending of their IS and the actual number of business partners, the type of network that better fits their needs;
- (*a priori* simulation) Newly formed companies, guided by the specified "white box" evaluation parameters, to identify suitable interoperable software systems to implement their business models.

Challenge 9: *Enterprises should be advised and informed on how to recognise ICT systems that might be inefficient.*

Challenge 10: *Interoperability related issues, which are damaging a more effective outcome of the collaboration network, should be identified and mitigated enabling companies and the networks they belong to, to evolve according to different scales of interoperability maturity.*

2.8.2. Need for a Self-Sustainable Interoperability

Being a novel concept suggested in this PhD work, little literature has been found related to enterprise/systems sustainable interoperability. Following the last challenges, it is only natural that after analysing its own interoperability status, companies would want to improve and do things

differently. Moreover, doing the parallelism with many of the problems raised on other domains, and as introduced in section 1.1.3 "The Problem of Sustainability of Interoperable Solutions", EI is strongly influenced by the people within enterprises: their decisions and complex behaviour influence market tendencies and consequently the evolution of systems and their interoperability within business networks. Despite thinking that one is improving, it might not be the case, since the evolutionary actions at the enterprise level could be deteriorating the level of interoperability at the overall network level, and thus, indirectly also diminishing the efficiency of its own activities.

As evidenced in the EI research roadmaps, today enterprise interoperability, has evolved from a technical business systems interconnection problem to a larger domain with multiple dimensions and multidisciplinary issues, that needs to be addressed using a more systemic and holistic way. This way, sustainability of systems interoperability appears as a natural requirement of the increasingly complexity of the EI domain. On a first perception, one could think that this sustainability has little to do with the "Earth's preservation" sustainability definition. However that is not entirely true. Being much closer to technology and the economic pillar due to the cost reduction that interoperability brings to businesses, it can also have a strong social impact motivated by the lifestyle changes provided by unrestrained (interoperability-wise) ICT innovation, and an indirect positive environmental feed.

2.8.2.1. The Grand Challenge

The Grand-Challenge: *Innovation and developments on manufacturing, service and ICT-related enterprises should envisage the development of new methods and infrastructures to* ***sustain*** *information models, standards, tools, and interoperability, thus aligning economic interests with social and environmental needs, and supporting their eco-systems.*

This challenge is designated as the grand challenge since it somewhat supersedes the previous ones and is directly related with the principal research question stated in this thesis plan. Despite all the others challenges are mostly technical, they are fundamental to address the grand challenge which cannot be dissociated from them.

3. Complexity and Dynamic Environments

Over the past decades complexity theory has become a broad ranging subject appreciated in a variety of areas. The study of complex adaptive systems has become the ultimate interdisciplinary science, focusing its modelling activities on how microstate events, whether particles, molecules, humans, or enterprises, self-organise into an emergent aggregate structure (Mckelvey, 1999)*. With roots in many disciplines such as evolutionary biology, non-linear dynamical systems, and artificial intelligence, models of CAS characterize the interplay between a system and its environment and the co-evolution of both* (T. Choi et al., 2001)*.*

In some aspects, complexity displays a clear proximity with the study of Enterprise Interoperability. In fact, models of CAS can be used to explore how large-scale (macro) order is bent by the features of local (micro) interactions, namely, how patterns of local relations and organisation's adaptive actions, impact the overall network behaviour (Wilkinson & Young, 2002)*. Hence, in the context of collaboration networks, ideas from complexity could provide contributions for a self-sustainable interoperability.*

3.1. Concerning Systems

To understand the complexity that might be hidden behind a system, the best is to define it. Following Mandel's (2011) claims, in a simple sentence, "*a system is a family*" with many constituents. However, there are just so many kinds of systems one could say almost everything is a system. As an antagonism, a "thing" by itself is not a system, but no one knows exactly what a "thing" is. Consequently, due to these difficulties the best approach is to describe how a system behaves instead of defining it: "*a system is what a system is doing"* (Mandel, 2011).

More formally, in its General System Theory, Bertalanffy (1976) defines a system as a "*set of elements standing in interrelations among themselves and with environment"*, i.e. a set of interoperable components combined in such a way as to perform a given function under specified conditions. Indeed, according to his view, a machine is an example of a tangible and clear cut system, but other exist which are not as tangible, e.g. systems involving people (e.g. cooperating in a workshop or within an enterprise) (Ho, 1998; Senge, 1994). The concept of system is recursive and may be composed of other sub-systems which are tangible or intangible interacting elements, thus one may generalise that a system is a view on the whole: a set of interacting entities performing a given activity that can be influenced by the surrounding environment (see Figure 3.1) (Couture, 2007).

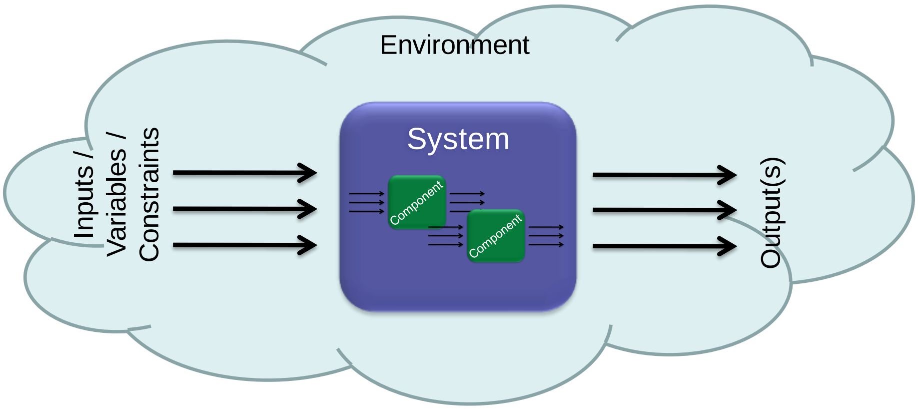


Figure 3.1: View of a System

The system's communication with its environment allows all necessary exchanges through the system's boundaries, even though transformation and production mechanisms are influenced by internal rules, beliefs, constraints, culture and internal models (Holland, 1996). Despite that, most systems share a common number of characteristics, namely the behaviour with respect to the inputs, variables and conditioners. Moreover, they are evolving with respect to time, thus are not considered as static sets of elements. For example, a static hardware part may potentially become a system when put in action with other elements such as humans or software processes (Couture, 2007). This dynamical aspect is essential to support concepts of complexity theory as in CAS.

3.1.1. Classification of Systems

There are multiple ways of characterizing systems (e.g. Ackoff (1971); Freetutes.com (2007); G. Lewis (2007); Selik et al. (2010); etc.), yet, classification is hardly complete or perfect for all purposes. As explained before, the most common method for approaching systems is regarding their function, but one can also look at systems concerning their tangibility, since they can be abstract and conceptual (e.g. formulas, representations, or models of real systems) or even physical if they are tangible entities that can be touched. In the last case, they can also be seen as static or dynamic in nature, and natural or designed, depending whether they are the product of human creation. Finally, regarding the interaction with the environment, systems are either open, or closed where no interaction with the environment is envisaged.

It is also possible to obtain a simple classification by analysing the properties of a system's behaviour. In this case, they can be time continuous, discrete, or neither; they can be linear or nonlinear; time invariant or time varying; as well as stable or unstable. Also the causality might be important to consider when analysing a system. All "real-time" systems must be causal because output depends only on current or past inputs, but not future. With these variations, one can conclude that depending on the purpose of the analysis, systems can be classified differently. In fact, authors have developed along the years many classifications for systems, but among the most known are the classifications summarized in Table 3.1.

Table 3.1: Some Classifications of Systems

	Classification	Description	Example
Ackoff (1998)	1. Deterministic/Mechanistic	Neither parts nor whole has choice	Machines
	2. Animated	The whole has choice, but not the parts	People
	3. Social	Both parts and whole have choice	Corporations
	4. Ecological	Some part have choice, but not the whole	Earth
Miller & Miller (1982)	1. Cells	Free-living or colonial forms	-
	2. Organs	Carry out organism processes	-
	3. Organisms	Multi-cellular plant and animal life forms	-
	4. Groups	Two or more organisms interacting as systems	-
	5. Organisations	Two or more echelons in their decider structures	-
	6. Societies	Self-subsistent system	-
	7. Supranational systems	Two or more societies	United Nations
Boulding (1956)	1. Frameworks	Static	Bridge
	2. Clock-works	Simple dynamics with predetermined motion	Engine
	3. Thermostat	Transmission and interpretation of information	Thermostat
	4. Open systems	Self-maintaining structure	Cell, River
	5. Genetic Societal	Division among cells to form a cell-society	Plant
	6. Animal	Increased mobility, theological behaviour and self-awareness	Birds
	7. Human	Intrinsic purpose or motivation, and self-consciousness	Human beings
	8. Socio-cultural systems	Distinguished by the role of humans in the environment	Families
	9. Transcendental systems	Unknowable's	-
Gharajedaghi (2005)	1. Passive	One structure and one function in the environment - No choice on both	Linear tools
	2. Reactive	Different structure but a single function on the environment - No choice on both	Self-maintaining systems
	3. Responsive	Several structures and functions on the environment - Has choice of means, not ends	Goal-seeking systems
	4. Active	Several structures and functions on the environment - Has choice of means and ends	Purposeful systems

Finally, important distinctions have also been made between hard and soft systems, where:

- *Hard systems* are associated with areas such as systems engineering, operations research and quantitative systems analysis; and
- *Soft systems*, which are commonly associated with methods such as action research and emphasizing participatory designs.

It is often stated that "hard" systems thinking is appropriate in well-defined technical problems (systemic) and that "soft" systems thinking is more appropriate in fuzzy hill-defined situations involving human beings and cultural considerations ("*the world is fuzzy but the enquiry is systemic"*) (Checkland, 1999). In this line, the creation of information systems (IS) is usually a relevant, and often a core, concern of hard systems whose functionalities are emphasised by the technology that supports them. However, following a soft system methodology, it is important to separate the two (systems and technology) to encourage more appreciation for the human aspects of information systems, i.e. the human-interaction.

Continuing with this premise, organisations and their subsystem IS (which are open systems that interact with their environments) must consider human activity subsystems as part of the overall system modelling process, and therefore classification of information systems cannot be focused only on functional/transactional processing (Freetutes.com, 2007). Management and decision support must be also considered, where they can be treated as responsive systems according to the Gharajedaghi (2005) classification (see summary Table 3.2).

Table 3.2: Analysis of Systems

	Information Systems								
Gharajedaghi (2005)	Passive	Reactive	Responsive	Active				Responsive	
(Ackoff, 1998)	Deterministic/Mechanistic		Animated	Social				Ecological	
Miller & Miller (1982)		Cells	Organs	Organisms	Groups	Organisations	Societies	Supranational systems	
Boulding (1956)	Frameworks	Clock-works	Thermostat	Open systems	Genetic Societal	Animal	Human	Socio-cultural systems	Transcendental systems

3.1.2. Cybernetics and Systems Thinking

Systems theory or science is the interdisciplinary study of the abstract organisation of phenomena, independently of substance, type, spatial or temporal scale of existence. It investigates both the principles common to all complex entities, and the (usually mathematical) models used to describe them (Heylighen & Joslyn, 1992). According to systems theory, all systems, however complex they are, have some kind of organisation which is often independent from the specific system or domain

(Frei et al., 2007). In this sense, rather than reducing an entity (e.g. the human body) to the properties of its parts or elements (e.g. organs or cells), systems theory focuses on the arrangement of the whole, i.e. holism (Bertalanffy, 1976).

Cybernetics is a closely related discipline that treats the aspects of communication and control by focusing on circular feedback mechanisms in complex systems (Frei et al., 2007). It has been established by Wiener (1965) as the science of communication and control in the animal and the machine, and focuses on how anything (digital, mechanical or biological) processes and reacts to information while changing itself. This approach of “Systems Thinking” is fundamentally different from that of traditional forms of analysis (Checkland, 1999). It is an attempt to avoid the reductionism of natural science, which focuses on separating the individual pieces of what is being studied. Systems’ thinking, in contrast, focuses on how the object of study interacts with the other systems of the same environment. This means that instead of isolating the smaller and smaller parts of the system being studied, systems thinking works by expanding its view to take into account larger numbers of interactions with same-level systems.

Conclusions obtained following a systems thinking analysis are sometimes strikingly different than those generated by the traditional forms of analysis, especially when what is being studied is dynamically complex or has a great deal of feedback from other sources. Examples of areas where systems thinking has proved relevant, include (Aronson, 1996):

- complex problems that involve helping many actors to see the big picture (e.g. management);
- recurring problems or those that have been created by attempts of solving them;
- issues where actions affect the surrounding environment.

Cybernetics and systems thinking are well established disciplines in management literature and could provide, in principle, interesting theories to certain Enterprise Interoperability problems. However, they are limited in scope to the boundaries of a system and by the perspective of holism. Cybernetic systems operate at the level of basic processes that are relatively undisturbed, while systems’ thinking is an approach to problem solving when substantial changes occurred in processes, viewing disruptions as parts of an overall system. However, neither can deal with major environmental changes of collaborative networks (Schary, 2007). In fact, the information age has highlighted the complex nature of systems, which move between ordered and disordered states, and networked enterprises are an emerging logical form of organising (Black & Edwards, 2000).

3.1.3. Complex Systems

Complex systems are a particular type of systems that cannot be described by a single rule. They originally have multiple objectives but still exhibit unexpected features not contained within their specification (Couture, 2007). Therefore, complex systems have rich dynamics with patterns and fluctuations on many scales of space and time, along with the absence of equilibrium. In fact, a key ingredient of a complex system, are the non-linear interactions among its constituents. Under special circumstances, they can give rise to coherent, emergent, complex behaviours with a very

rich structure. These cannot be attributed to single separate subsystems but rather to a collective effect, like in general systems theory, where the whole is seen as more than the sum of its parts (Bertalanffy, 1976).

Complex systems interrelationships are described by Gharajedaghi (2005) depending on the level and compatibility of their inner elements and finalities as:

- *Cooperation* – elements and finalities are compatible;
- *Coalition* – compatible elements but not finalities;
- *Competition* – compatible finalities but not elements; and
- *Conflict* – elements and finalities are incompatible.

All describe the capacity of a system to achieve its mission. Also, one can observe an intuitive judgment of the degree of complexity following Boulding's classification of system types (see Table 3.1). His types range from structures and frameworks at level 1 to transcendental systems at level 9 with new complexity characteristics emerging at each level, and any problem exhibiting the characteristics of a certain level cannot be solved with a lower level system (Boulding, 1956).

Most complex systems are computationally irreducible, thus the only way to decide upon its evolution is to actually let them evolve in time. This means that no trivial single configuration of the system will satisfy all interactions simultaneously. Whereas simple and complicated (mechanical) systems are described in terms of Euclidian geometry, complex systems are better understood by drawing upon fractal geometry and related concepts of recursion, feedback, and self-similarity (University of Alberta, 2011). Nevertheless, what is most striking is that complex systems that apparently have little in common, e.g. a collection of machines in a manufacturing plant, nodes in a business network, or even a group of human agents in an economic setting, often share remarkably similar structures and behaviour (Couture, 2007). Thus, many natural phenomena can be considered to be complex systems, and their study (complexity science - section 3.2) is highly interdisciplinary.

3.2. Complexity

Human life is frequently described as becoming more and more complex, and rightly so. It seems that the terms "complex" or "complexity" appear everywhere. In some part, this is because life really is complex, but this conclusion is also driven by the fact that over the last few decades, much has been learned about the nature of complexity and the role that complexity plays in our lives (Courtney et al., 2008). Complexity theory has become a broad ranging subject appreciated in a variety of ways and illustrated in many books such as Anderson et al. (1988), Holland (1996), Kauffman (1996), Mainzer (1996), Nicolis & Prigogine (1989), Papadimitriou (1994), or Waldrop (1992), just to point some of the most referenced.

Nowadays, it is viewed as a source of concepts for enabling the trans-disciplinary exploration of complex organisations in the networked economy and society, and for explaining the dynamics of networked systems at different levels of description (from the micro- to the macro-level). It offers a

powerful set of methods for explaining non-linear, emergent behaviour in systems. Being such interdisciplinary, complexity science is approached differently by three major schools (Mckelvey, 1999; Merali & McKelvey, 2006):

- *European:* which follows the theories of Nicolis & Prigogine (1989), as well as other recognised authors (e.g. Mainzer (1996)). It focuses on the discovery that certain levels of negentropy[38] in physical systems at the "edge of chaos", cause aggregate "dissipative structures" to emerge from a stochastic "soup" of microstates. According to the European school, theses structure evidence predictability, thus providing scientific explanations to complexity at the edge of chaos. It is a mathematical intensive school;
- *American:* which focuses on how new order arises in biological and social systems. Represented by the "The Santa Fe Institute[39]", it studies the so-called "edge of chaos" where order emerges when heterogeneous agents – such as biomolecules, organisms, people or social systems – are motivated, by a drive for improved fitness or learning, to initiate connections with other agents. Micro-systems operating at the vicinity of the edge exhibit creativity and produce new and novel behaviours at the whole system (Kauffman, 1996);
- *Econophysics:* Where the focus is on how the order creation actually unfolds once the forces of emergent order creation by self-organising agents are set in motion. Key parts of this third aspect are fractal structures, power laws and scale-free theory.

The science of complexity is, therefore, the study of emergent order in what are very disorderly systems. Spirals in whirlpools, funnels in tornadoes, flocks of birds, schools of fish, are all examples of orderly behaviour in systems that are neither centrally planned nor centrally controlled. In a sense, complex systems innovate by producing spontaneous, systemic bouts of novelty out of which new patterns of behaviour emerge. Understanding its influence on the performance of organisations could lead to major gains in driving businesses (McElroy, 2000).

Complexity theory offers a number of new insights, analytical methods, and conceptual frameworks, and it may offer a synthesis on how organisations adapt to their environments and population. The most exciting characteristic is, perhaps, the promise that complexity theory will help understanding how systems can learn more effectively, and spontaneously self-organise into more structured and sophisticated forms that are better adapted to their environments. Although most of these findings are yet preliminary and confined to computer simulations, management scientists are claiming to be able to apply complexity principles to bring enterprises to "the edge of chaos", enhancing creativity, learning, and adaptation (Lewin, 1999).

Following these findings, the application of complexity theory to IS design, implementation, testing, installation, and maintenance is under research as well. When looking at complexity in ICT, one cannot overlook the organisational structures that technology supports. It underlies a huge part of

---

[38] Also known as negative entropy - the entropy that a system exports to keep its own entropy low. In information theory and statistics, negentropy is used as a measure of distance to normality

[39] Santa Fe – Complexity Research Expanding the Boundaries of Science (www.santafe.edu/)

operations in modern organisations, so by extrapolation, the role of IS is under the scope of complexity science. As in general systems theory, the idea that the *"whole is greater than the sum of the parts"* is fundamental. Thus, complexity cannot be foreseen from an examination of the constituent parts of an IS. It is, instead, a characteristic that emerges after the parts are tangled in a way that subsequent separation would destroy the whole, creating integration and interoperability problems (McElroy, 2000).

3.2.1. Traditional Science, Complexity, and Systems Science: What is different?

Traditional science has tended to focus on simple cause–effect relationships, e.g.: Newton's well known formula in which force equals the product of mass and acceleration; etc. However, it is not surprising that early science focused on simple laws, since they are the easiest regularities to replicate, detect, control, and measure.

On the other hand, complexity science introduces a new way of studying. It posits simple causes for complex effects, with rules that typically determine how a set of artificial agents will behave in their virtual environment over time, including their interaction with other agents. Unlike traditional science, it does not predict an outcome for every state; instead, it uses feedback and learning algorithms to enable complex adaptive systems to adapt to its environment over time. The application of these rules to a large population leads to emergent behaviour that may bear some resemblance to real-world phenomena (Phelan, 2001).

Advocates of complexity theory see it as a means of simplifying complex systems. However, due to the absence of a single identifiable complexity theory, the actual practice is not simple. Instead, a number of theories concerned with complex systems gather under the general banner of complexity research (Manson, 2001). Complexity theory may be traced back to conceptual antecedents such as cybernetics, cellular automata, and also to general systems theory with the anti-reductionism and holistic appreciation of system interconnectedness (Bertalanffy, 1976). In fact, some literature claims systems theory and complexity theory are the same. Indeed, for some systems theorists, many of the central concepts in complexity theory are simply "old wine in new bottles". Several of the terms used carry virtually the same definition in both theories, including the notion of "system", "emergence", "dynamic", "nonlinear", "adaptive", etc.; and both theories also share a belief that there are universal principles underlying the behaviour of systems, but they are not the same (Manson 2001; Phelan 1999):

- One of the basic premises of complexity theory is that much of the apparently complex aggregate behaviour in any system arises from the relatively simple and localized activities of its agents. Systems theory instead favours simplification and parameterization of flows, a process that assumes that the system exists in equilibrium and therefore contradicts the need to examine relationships between system elements;
- Complexity concerns nonlinear relationships between constantly changing entities, while systems theory, in contrast, studies entities linked by linear relationships;

- Complexity research contends that systems have emergent or synergistic characteristics that cannot be understood without reference to sub-component relationships;
- Systems theory is predominantly focused on confirmatory analysis, seeking to identify relationships between elements in a system and then to optimise some objective function. A problem-solving perspective is present in hard system methodologies, as well as in much of the work in system dynamics and cybernetics. Complexity, on the other hand, is more focused on the exploratory nature of work, with researchers typically starting to look at an overall complex system, and proceed to conjecture relationships between agents or elements, in interaction, that may explain the system's aggregate behaviour.

In summary, complexity theory tends to focus on exploratory analysis and systems theory favours confirmatory analysis or "problem-solving" using feedback - based models. Complexity research employs techniques such as artificial intelligence to examine qualitative characteristics such as the symbolic content of communication.

3.2.2. Chaos and Edge of Chaos

Chaos theory was pioneered by Lorenz in 1963, who was studying the dynamics of turbulent flow in fluids. Its name comes from the fact that the systems described in the theory are apparently disordered, but chaos theory is really about finding the underlying order in apparently random data. It arises for example when trying to calculate the path of an object in the gravitational pull of two or more bodies. By observing the motion of a toy, a metal ball suspended over two or more magnets, it becomes evident that the ball traces a series of patterns that never exactly repeat themselves and yet are not totally random (Levy, 2000). The paradox here is that the motion of the metal ball is driven by the same Newtonian equations as the well understood case of a single gravitational attractor. If one knew precisely the original location, speed, and direction of the ball, it ought to be able to predict its path with a reasonable degree of accuracy.

Like systems theory, catastrophe theory is based on the idea of transitions between equilibrium states. The difference is that in catastrophe, the transitions occur abruptly. As in the metal ball example, it is possible to define a system in a perfectly deterministic way, by defining all the rules, and yet when the system is set in motion, it becomes totally unpredictable. This is the phenomenon of chaos, thus a deterministic system is chaotic if an infinitesimal change on the initial conditions leads to an entirely different series of events (Bentley & Maschner, 2007).

Mathematically, chaotic systems are represented by differential equations that cannot be solved, so computer modelling and simulation techniques need to be employed to follow the path of such a system. However, chaos was discovered with the realization that deterministic equations, when repeatedly applied, can lead to results that are as sensitive to the initial values of the parameters as to be unpredictable (Gleick, 2008). To show how this can happen, and following a simple example from Bentley & Maschner (2007), suppose that one wishes to model the population of algae in a pond. There are two tendencies acting on the population level, one for the existing algae to multiply, and other for some algae to die as the population reaches the capacity of the pond. The next eq. calculates the population of the pond Pt at a given time (t):

$$Pt = A * P(t-1) * (1 - P(t-1)) \quad (1)$$

A is the algae multiplication factor (a constant greater than one) and $P(t-1)$ is the population of the pond in previous year, which can vary from 0 (no algae) to 1 (pond at capacity). To arrive at the population in successive years, equation 1 must be iterated. Now, if the value of *A* is between 1.0 and 3.0 the pond converges to a stable population (left side of Figure 3.2), called an *“attractor”*, which can be seen as the result of an equilibrium process. However if the factor is above 3.0, the system evidences *chaotic behaviour* (right side of Figure 3.2).

Scientists have now moved beyond chaos and have come up with a further elaboration – complexity theory, where the phrase, “the *edge of chaos*” (Waldrop, 1992), is often used to describe complexity, although the “edge of stability” applies equally well. It is the zone between stability and predictability, on one side, and chaos and unpredictability, on the other.

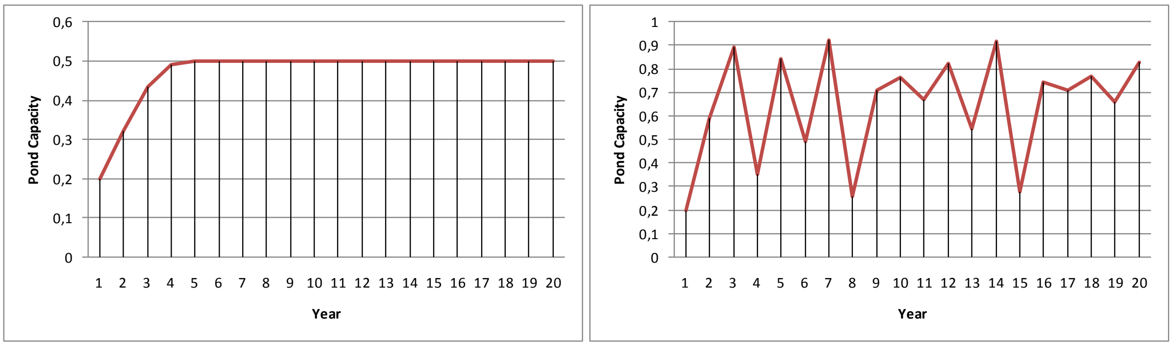


Figure 3.2: Example “Population of Algae in a Pond”: From Equilibrium to Chaotic Behaviour

In the complexity zone, systems adapt, learn and grow, while in stability nothing changes, and in chaos, there are too many changes happening for anything to be learned. Also, in enterprise systems, if there is too much change and freedom, then systems can tip over into chaos (e.g. social revolutions). On the other hand, too little innovation and systems become rigid, totally predictable and able to respond only through tried and established methods (R. Lewis, 1994). Therefore, whenever possible, a bit of “personality” needs to be added to the elements so that the interactions between agents and its neighbours can be cooperative, competitive, reactive or a mixture of them (Bentley & Maschner, 2007).

3.2.3. Complex Adaptive Systems (CAS)

CAS are complex systems that emerge over time into a coherent form, adapting and organising themselves without any singular entity deliberately managing or controlling them (Holland, 1996). The system is embedded in a changing environment, with which it exchanges energy, matter and information, exhibiting coherence under changes, via adaptive feedback loops (Figure 3.3).

Kauffman's research illustrates the concept of a CAS as a system made up of a large number of agents, each behaving according to its own rules or principles in response to local interactions with other system agents, adapting through a mechanism of self-organisation. What he discovered was the surprising fact that adaptation is optimised when the system exists at the edge of chaos (Kauffman, 1996). The system achieves a state of self-organised criticality without a blueprint or

control mechanism, evolving to the edge of chaos where it is optimised for adaptation. The number of interactions is great enough so that novel change can occur, and still not become totally unstable (Kirshbaum, 2002; Couture, 2007).

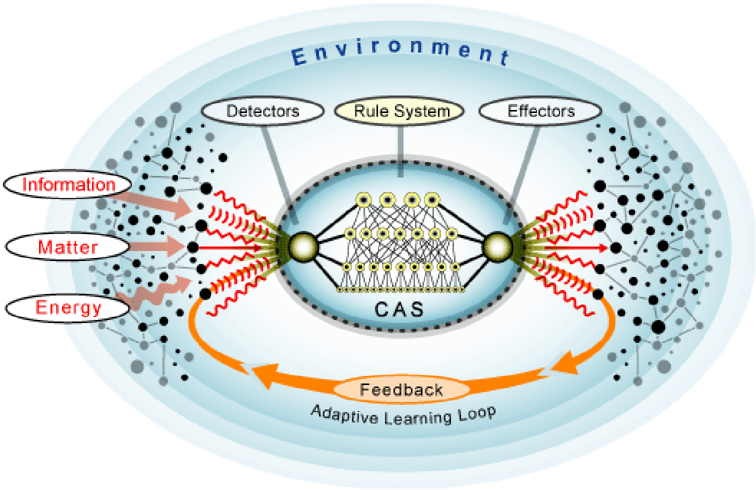


Figure 3.3: Complex Adaptive System (CAS) Model (Clemens, 2004)

Therefore, a complex adaptive system is said to be comprised of aggregation, nonlinearity, flows, diversity, tagging, internal models, and building blocks. Such systems are well placed to explore new niches, search their fitness landscape[40], changing their composition to fit the changing patterns they encounter.

What these mean in the context of information systems (IS) is still subject of analysis by researchers (Courtney et al., 2008). The dynamics of interaction between elements gives rise to a number of features that can be difficult to reconcile with some of the doctrines of the classical IS. However, machines based on these concepts will act like living cells, companies will transform from static into dynamic organisms, and societies will evolve into diverse ecosystems (Kirshbaum, 2002). CAS demonstrate intrinsic purpose, self-awareness, self-adaptation or motivation, thus would fit on levels "active" and "responsive" of Gharajedaghi's classification of systems (section 3.1.1 "Classification of Systems").

3.3. Complex Networks

The availability of large data sets has allowed researchers to uncover complex properties as large-scale fluctuations and heterogeneities in many networks. Until recently, these networked systems were considered as random sets of points and connections, but nowadays they are subject of study for understanding the effect of complex connectivity patterns and dynamics (Barrat et al., 2008; European Science Foundation, 2010; K. Park & Willinger, 2005; Suranaa et al., 2005; Wycisk et al., 2008).

The resilience and robustness of large infrastructures can be studied by filtering models in which one can progressively act on the network. For example, social behaviour may be often modelled

[40] way of visualizing a problem in optimisation or adaptation (see Table 3.5 "Formal Methods and Systemic Approaches Inherent to Complexity, Complex Networks and CAS" for more detail)

through simple dynamical processes and agent-based models. Therefore, a vast number of systems, from brain to ecosystems, power grids and the Internet, possess complexity properties and can be represented as large complex networks (Barrat et al., 2008).

3.3.1. Internet and WWW as a Complex Network

The Internet has been growing exponentially since its origins. With the network extent to segments of the population and emerging countries like China, India and Brazil, the number of users has increased more than 500% from approximately 70 million in 1998, to more than 2.2 billion[41] in 2011. The digital revolution and the advent of the Internet are transforming the way people work and spend their free time, as well as global communications and social relations. It is this all-pervading access and the different relationships that these connections represent that make the Internet one of our greatest tools for sustainable change, development, and innovation. Nevertheless, the relationship between Internet and society is complex and bidirectional, leading to a co-evolution of the two systems (European Science Foundation, 2010).

The complexity of the current Internet structure and its future developments cannot be understood without taking a full multi-disciplinary approach, which must necessarily be based on the science of complex systems, and in particular complex network theory. The Internet may be viewed as a system with diverse features and many components that can give rise to unexpected emergent phenomena, revealing much about its own engineering. In this line, discoveries of fractal or self-similar network traffic traces, power-law[42] behaviour, and WWW connectivity are instances of emergence (K. Park & Willinger, 2005).

Many people think of the network as simply the Internet. However, just as there are central and peripheral nervous systems in the human body, one can say there is the public Internet and then all of the other networks in the world, which range from simple, connecting only one or two devices to extremely complicated, connecting hundred or thousands of devices and resources. In a nutshell, the Internet is the foundation for nowadays world’s communication, and at higher level, it helps businesses foster new relationships that are essential to growth and overall productivity (Sorensen, 2009). However, when habited by selfish or greedy users, it behaves as a non-cooperative structure, thus systems stability is key to understanding the overall network and its dynamics. Fault-tolerance and robustness of large-scale network systems can exhibit spatial and temporal correlations whose effective analysis and management may benefit from rescaling techniques applied in certain systems (K. Park & Willinger, 2005).

At the infrastructure level, the Internet, defined as the world-wide collection of internet protocol speaking networks, is a multifaceted system with many components and diverse features. As in social level, it is also a complex system in the sense of a complicated system exhibiting simple behaviour due to a compendium of innate architectural features, as opposed to a simple system exhibiting complex behaviour. At its physical basis, the Internet is a flow network whose information

[41] www.internetworldstats.com

[42] When the frequency of an event varies as a power of some attribute of that event (e.g. its size), the frequency is said to follow a power law

transmission is governed by communication theory (Shannon & Weaver, 1998). Thus, approaching the Internet and the WWW with a complexity perspective, means using interdisciplinary tools and methods to tackle the subject area, e.g. self-similar traffic, power-law connectivity, non-cooperative network games, scalable traffic control, and organisational behaviour (K. Park & Willinger, 2005).

3.3.2. Industrial Networks

As one can guess by extrapolation of the text above that industrial networks are also target of similar considerations. From a business perspective, they should not be targeted as part of the WWW, but as a component of the Internet, thus themselves complex networks. Manufacturing networks from supply chains to dispersed networks, or multi-domain engineering networks such as collaborative product development, are among some of the industrial complex networks that have been target of analysis by the complexity community (T. Choi et al., 2001; A. Durresi et al., 2009; Estefan, 2007; Suranaa et al., 2005; J. X. Wang & M. X. Tang, 2006; Wycisk et al., 2008).

As explained, some of the key points of CAS are that they contain critical nonlinear features where a small input signal can be magnified to impact the entire system. In industrial networks strategy, the same applies with agents competing for resources and generating complex behaviour (Noori & Lee, 2009). For example, early in the life of a new industry, it is often witnessed a multiplicity of competing technologies (inputs).

Some researchers argue that population selection processes, reinforced by the feedback mechanisms, lead to the emergence of a single dominant form which, and from there on, evidences linearity and complexity free behaviour. This was the case of automotive industry at the beginning of the 20^{th} century, where manufacturers initially competed with electric, steam, and internal combustion engines. However, that theory proved to be wrong since nowadays automotive industry is again experiencing a similar competition for technologies as an alternative to fuel. Also, Brown & Eisenhardt (1997), in a study of the computer industry, have challenged this linear model and maintain that complexity is always present and motivated by rapid, continuous change. They argue that the ability to sustain continuous change, rather than a particular static configuration of resources and assets, is an important core capability of great strategic value for industrial networks (Levy, 2000).

3.3.2.1. Dispersed Manufacturing Networks (DMN)

One of the main characteristics of DMN is the ability to manufacture a high variety of products (i.e. mass customization) in flexible and geographically dispersed environments. This is a synonym for dynamicity and uncertainty, since it implies a higher degree of customization. On the other hand, as networks expand to provide wider product options, supply chains become nearly unmanageable to timely respond to consumer demands (Noori & Lee, 2009).

For these reasons, recent insights of complexity into the field of collaboration allow the full exploitation of the benefits it may yield in comparison with traditional organisation paradigms (Dekkers, 2009). Instead of making vain attempts to control these chaotic networked environments in a top down fashion, networks are being challenged to understand its own problems at micro

scale level (enterprise) and find solutions that can impact the entire system, thus enabling a self-organisation. As Noori & Lee (2009) suggest, although complexity has a strong mathematical basis (e.g. Kauffman 1996), certain qualitative heuristic rules can be applied to DMN business models. For instance, networks can work better in the presence of noise and overly rigid organisation structures are unnecessary and even counter-productive concerning efficient operation (*decentralized control*). Other useful heuristic is the free adaptability of companies belonging to the network, since those that cannot maintain an ever evolving standard of fitness will not prevail in nowadays global and competitive market (*dynamic internal adaptation*).

3.3.2.2. Systems Engineering Networks

As manufacturing which is more focused on the production side of a product, also systems engineering (SE) has been a target of multiple complex collaborative networks. It demands the combination of multiple processes, such as requirements elicitation, product design, modelling, teams' management, etc, that normally are performed in conjunction by different enterprises. The SE method recognises that each system is an integrated whole, composed of diverse, specialised structures and sub-functions that aid in the development and lifecycle of a product, using information and product models to describe and integrate the whole process (Chestnut, 1965).

Traditionally, SE are managed by large projects where organisations employ a document-based approach characterized by the generation of textual specifications and design documents, in hard-copy or electronic file formats, which are then exchanged between customers, users, developers, and testers. These documents represent the systems requirements and the design information, by which the systems engineers are responsible for controlling, ensuring validity, completeness, and consistency along the full engineering process (Ogren, 2000). This method is rigorous but has some limitations, like the consistency and relationships between requirements, design, engineering analysis, and test information. These are difficult to assess since the information is spread across several documents.

In fact, and according to Honour (2008), defining systems engineering in terms of an invariant set of best practices has not worked since each new standard or trend keeps changing the practices. It appears that a more appropriate approach is to define systems engineering in terms of the engineering of complexity. Thus, because of the limitations in SE and the need to improve the quality and access to information, techniques like model-based systems engineering (MBSE) have been introduced with the intention to model complex relationships/dependencies, and facilitate and reduce the complexity of SE activities (INCOSE & OMG DSIG, 2011) (see section 3.6.4).

3.3.3. Network Theory

Both chaos and complexity theory attempt to reconcile the essential unpredictability of nonlinear dynamic systems with a sense of underlying order and structure. Network theory is less concerned with underlying simplicity, since it tends to rely on huge computing power to model large numbers of nodes connected by simple logical rules. It is more interested on systems in the "edge of chaos" and in the emergent order and patterns in complex systems, than in an attempt to find a simple

mathematical formula for the system. Network models often try to capture the essence of interaction among the many agents in a system, whereas chaos theory generally attempts to model some resultant outcome, such as prices or investment. Also, network theory has relevance to brains - neural networks; to organisations - networks of departments and people; and to industries - networks of enterprises (Levy, 2000).

Being an area directly related to computer science, network theory also concerns the study of graphs as a representation of either symmetric relations or, more generally, of asymmetric relations between discrete objects. In this sense, classical networks are treated has graphs that may contain both active and passive nodes connected through edges (directed or not) (Diestel, 2010). However, some argue that the science of networks that has been taking shape over the last few years is distinguished from preceding work on networks in three important ways:

1. by focusing on the properties of real-world networks, it is concerned with empirical as well as theoretical questions;
2. it frequently takes the view that networks are not static, but evolve in time according to various dynamical rules;
3. and it aims, ultimately at least, to understand networks not just as topological objects, but also as the framework upon which distributed dynamic systems are built.

With the possible exception of certain types of random graph models, network analysis (e.g. in the social sciences) has largely avoided modelling, preferring simply to describe the properties of networks as observed in collected data (Weisstein, 2011).

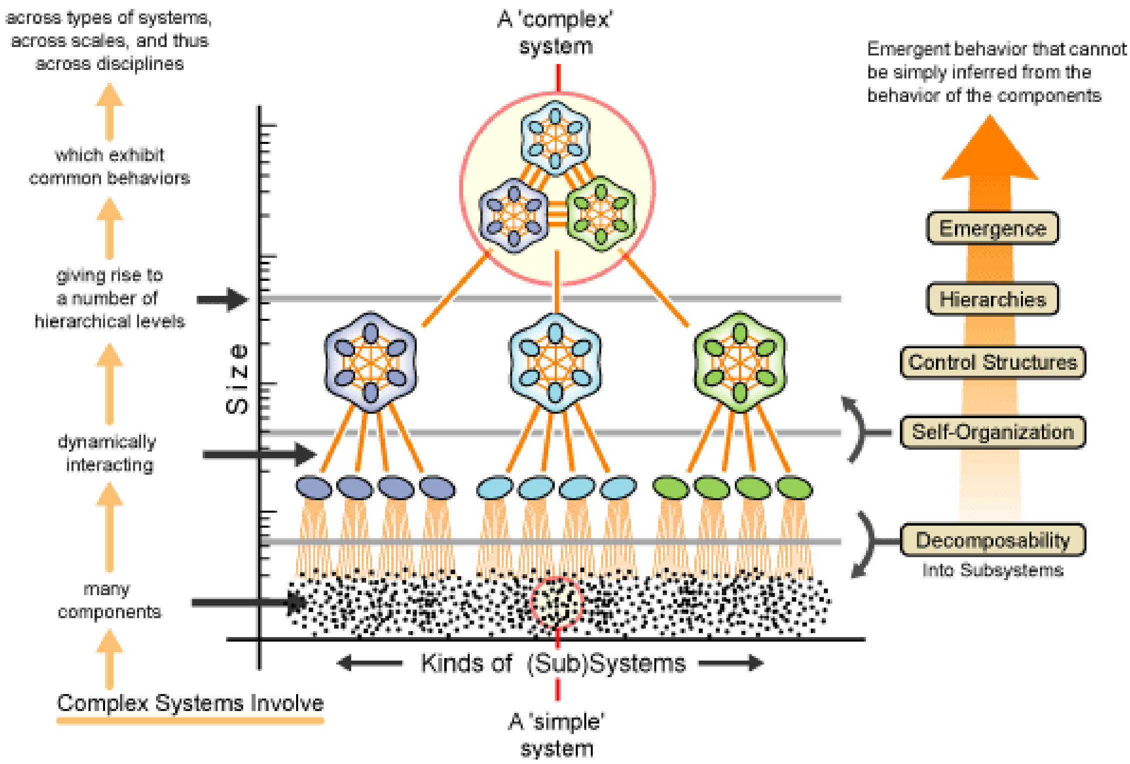


Figure 3.4: Complex Systems' Evolution (Clemens, 2004)

3.3.4. Behaviour and Properties of Complexity in CAS Networks

In the global market, companies and the networks which they are part of, tend to follow a dynamic and evolutionary behaviour as in CAS, exhibiting elementary and behaviour properties which are influenced by the environment (T. Choi et al., 2001; Wycisk et al., 2008). Complex networks are referred to as "adaptive" or "dynamic", because they are constantly changing their interrelationships based upon the needs of individual agents and environmental impacts (Couture, 2007). Figure 3.4 summarizes most of what has been described in this chapter concerning complex systems and networks: on the left their natural behaviour is illustrated, from multiple components dynamically interacting and giving rise to complex structures exhibiting common behaviour across multiple disciplines; on the right side, one could observe the phenomena of emergence; and on the centre, it illustrates the grouping of simpler systems into larger complex ones.

Table 3.3 complements that understanding of complexity in systems and networks, capturing some of the most important, and frequently interlinked, features which enable to indentify and characterize a complex adaptive system. To maintain readability and objectivity, it does not intend to be an exhaustive list and other terminology that also cross-cut much of the complexity literature has not been included. These include co-evolution, butterfly effect, dynamism, interaction, edge of chaos, symmetry breaking as described by several authors such as (Nicolis & Prigogine, 1989; Waldrop, 1992; T. Choi et al., 2001; Kirshbaum, 2002; Courtney et al., 2008; Wycisk et al., 2008)

Table 3.3: Complexity Core Features and Properties

Complexity Core Features	Description	References
Dimensionality	The dimensionality of a complex system is defined as the degree of freedom that individual agents within the system have to enact behaviour in a somewhat autonomous fashion. A system with high dimensionality evidences tends to be difficult to control. On the other hand, controls act as a form of negative feedback, effectively reducing dimensionality, e.g. rules and regulations or budgetary restrictions, ensure that an individual agent's behaviour is greatly limited, thus, changing the complexity and helping systems to behave more predictably and cybernetically.	(T. Choi et al., 2001)
Non-linearity	In a complex system, it is often true that the only way to predict how the system will behave in the future is to wait literally for the future to unfold. Non-linear relationships are relationships in which a change of given magnitude in the input to the system is not matched in a linear fashion to a corresponding change in output.	(T. Choi et al., 2001)
Decomposability	Modularity or near decomposability shows up in many scientific disciplines in a matter of degrees, e.g., it may refer merely to the ability to delineate one system module from another (recognizing the boundaries of the module), or to the ability to actually separate components from one another. This decomposability also enables the complexity of a system (or process) to be reduced into more controllable agents that work together to produce emergent properties.	(Simon, 1962; Simon, 1995; L. Chen & Li, 2004; Honour, 2008)
Reflexivity	As given by the degree of dimensionality, the only thing constant in complex networks and systems is "change". Each agent acts, and its	(Maturana & Varela, 1980; T. Choi et al.,

Complexity Core Features	Description	References
	actions have impacts on the other agents around it. A common way for changes to occur is by altering the boundaries of the system, which change as a result of including or excluding particular sub-system agents and by adding or eliminating connections among them, thereby changing the underlying patterns of interactions. However, other agents act in response to the change in their own environment, thereby changing the environment for the original agent. Furthermore, the environment can impose new rules and norms. On response to change, adaptation includes structural, physiological and/or behavioural response that increases the expected long-term success of a system and its reflexive nature. In the case that systems are not adequately adapted to their environment they will either have to move out of the habitat or eventually die off.	2001; Honour, 2008; Wycisk et al., 2008)
Ability to Learn	In such a structure of agents and reflexive interactions, a complex system is not beholden exclusively to the environment, i.e. it operates on local information rather than global, actively shaping, reacting, anticipating, and remembering through the persistence of internal structures. Therefore, systems are able to adapt by modifying their individual capabilities and change their rules of action so as to improve their performance as experience accumulates. In doing so, system agents can deal with truly novel situations, searching the so-called "building blocks" for a set of plausible rules enabling them to interact within a CAS.	(Holland, 1992; Holland, 1996; Honour, 2008; Wycisk et al., 2008)
Emergence	A process by which a number of lower level systems (considered as agents) self-organise into a higher lever autonomous complex system (a whole) with transcendent macro behaviours. Emergent properties are those behaviours that are perceptible only at the system level (macro) and cannot be perceived or even predicted from the behaviours of the agent parts (micro). Emergence occurs between the edges of order and chaos via processes of interaction and self-organisation.	(Bak et al., 1988; Holland, 1998; Manson, 2001; Phelan, 2001; Kauffman & Clayton, 2006; Honour, 2008)
Self-Organisation	Allows a system to change its internal structure in order to better interact with its environment. This process results from the autonomous interaction of single agents within a system giving rise to bottom-up (new) order creation by a system itself, as opposed to structure and process imposed on the system by outside entities (controllers). Self-Organisation is often linked to emergence.	(Nicolis & Prigogine, 1989; Manson, 2001; Wycisk et al., 2008)
Quasi-equilibrium	Equilibrium is a state of a system that, if not subjected to perturbation, will remain unchanged. Related with the non-linearity feature, normally a CAS tends to oscillate between a state of equilibrium and non-equilibrium due to the action of attractors that are sensitive to change as the CAS is pulled away from quasi-equilibrium state to a far-from-equilibrium state. The term "self-organised criticality" also refers to the ability of complex systems to balance between randomness and stasis. Instead of occasionally weathering a crisis, a system can reach a critical point where its internal structure lies on the brink of collapsing without actually doing so.	(Bak et al., 1988; T. Choi et al., 2001; L. Chen & Li, 2004)

3.4. Relationship to Enterprise and Systems Interoperability

It is possible to examine complexity on a discipline-by-discipline basis, and according to Manson (2001), breaking complexity research into three major divisions affords a more coherent understanding of complexity theory. He divides it into (1) *Algorithmic complexity*, in the form of mathematical complexity theory and information theory, for describing the complexity of internal system characteristics; (2) *Deterministic complexity* dealing with chaos and catastrophe theory, for studying how the interaction of two or three agents can create sudden discontinuities in stable systems; and (3) *Aggregate complexity* concerning how individual elements work together to create systems with complex behaviour.

Complexity Science, namely the research available under aggregate complexity has been important when applied to economics and organisational management, and can also be relevant to the study of enterprise and systems interoperability. In fact authors as in Merali & McKelvey (2006) believe that the IS researchers of the 21st century should conduct a paradigm shift towards a more dynamical theoretical and methodological framework better suited to support idiosyncratic aspects of IS, business and organisational domains. In that sense, aggregate complexity can be the answer, since:

- It contains mature concepts and methodologies focused on accessing the holism and synergy, resulting from the interaction of system components, as occurs in enterprise networks;
- If one considers that a system is a broad concept that can enclose software, services, enterprises, and even the entire collaborative network, then it can be closely related to many of the interoperability areas, understanding the interactions between different organisation's information models and schemas; business processes; software applications and services; ontologies; clouds of services and enterprise ecosystems; people; etc.

Aggregate complexity illustrates how relationships are more important than the attributes, in defining the nature of components. This domain of Complexity addresses a number of inter-related concepts, as summarised in Table 3.3, which are now used to attempt identifying more concrete relationships with the concepts and issues of EI (see Table 3.4) (ENSEMBLE Partners, 2012).

Table 3.4: Complexity Science Relationship to Enterprise and Systems Interoperability

Complexity Core Features	Relationship with the Topics of EI and Systems Interoperability
Dimensionality	Legislation and enterprise rules, regulations or budgetary restrictions, ensure that a system's behaviour is greatly limited, thus, changing the complexity and helping systems to behave more predictably according to different organisation structures.
Non-linearity	Non-linear systems are unpredictable thus their interoperation cannot be controlled externally. Enterprises should have internal mechanisms capable of measuring non-linearity to avoid entering in non-profitable relationships.

Complexity Core Features	Relationship with the Topics of EI and Systems Interoperability
Decomposability	When software systems or data models are divided into modules and components, decomposability is being used in the separation of concerns. Decomposed parts evidence lower complexity values, but the aggregate systems need to manage all the relationships.
Reflexivity	Enterprises are frequently changing to respond to new market requirements or internal needs for evolution. Frequently, when that happens they might need to realign their business processes, service compositions, and sometimes even software systems which, in turn, may impact their relationships with other enterprises. These may have the need to adapt as well, and might cause a reflexive behaviour.
Ability to Learn	The learning ability of physical entities requires the newest information and communication technologies (e.g. multi-agent-based models, RFID tags, etc.), and knowledge intensive data repositories (e.g. ontologies) that retain past knowledge used in the learning process.
Emergence	The degree of freedom that individual agents are allowed in information system is rather reduced, but still they can emerge into higher level systems, e.g. mashups created using automatic service discovery and composition; or cloud federation. Nevertheless, if the system depends on a human activity, it tends to exhibit positive feedback and enable complex phenomena, as social networks emergence or enterprise ecosystems.
Self-Organisation	Self-organisation within EI is related to the capability for a system to adapt automatically to external changes, based on a certain degree of intelligence and the previous acquired knowledge and semantics.
Quasi-equilibrium	Is closely related to monitoring, feedback and self-organisation capabilities. Only using them, is possible to detect when a system is approaching non-equilibrium and prevented from collapsing into randomness. This way, in one sense quasi-equilibrium is an issue for all EI areas, but on the other hand, it is software systems (applications) that categorize conformance and interoperability testing which can detect when systems are approaching dangerous situations.

3.4.1. Formal Methods and Systemic Approaches

Among the several theories associated to the science of complexity, complexity theory itself, network theory, and chaos theory appear to be the most relevant to the EI and systems interoperability areas addressed in this dissertation. They define formal methods and significant approaches that can contribute towards the hypothesis specified in section 1.2.4. Nevertheless, also systems theory, catastrophe theory, and fractal theory propose some of the methods enumerated in Table 3.5, below. This table does not intend to be exhaustive. It highlights a possible distribution and some EI applications of methods and approaches used throughout the several complexity domains.

Table 3.5: Formal Methods and Systemic Approaches Inherent to Complexity, Complex Networks and CAS

Complexity Core Features	Comment	Methods & Systemic Approaches	Description
Dimensionality	In EI, cooperation is fundamental and companies benefit from participation in cooperative networks where they can maximize their return of investment. Nevertheless, enterprise systems with high dimensionality levels tend to be too uncooperative while on the opposite side too much control tends to diminish innovation and chances of succeeds. This way, complexity can provide methods and approaches that help measuring and maintaining enterprises freedom in efficient levels.	Non-cooperative games	In game theory, a non-cooperative game is one in which players make decisions independently. It is not defined as games in which players do not cooperate, but as games in which any cooperation must be self-enforcing and within the contracts modelled in the game (Shor, 2005).
		Fitness Landscape	If one rates every option in state space by its achievement against some criteria then he can plot that rating as a fitness value on another dimension, i.e. a height that gives the appearance of a landscape. The result may be a single smooth hill (a correlated landscape), many smaller peaks (a rugged landscape) or something in between. If high fitness corresponds to high locations in the landscape, and if changes are mapped to movements in the landscape, then evolution will tend to make systems move in an uphill direction on the fitness landscape. According to R. Lewis (1994) and Kirshbaum (2002), fitness landscapes can be seen as the hills and valleys of change.
		Organisational Management methods	Management in all business and organisational activities is the act of getting people together to accomplish desired goals and objectives using available resources efficiently and effectively. The structure and pattern of communications within an organisation has a significant influence on the accuracy of decisions, the speed with which they can be reached, and the satisfaction of the people involved.
		Decentralized control	Decentralized systems are based upon distributed control in which individual components react to local conditions in real time. These components are linked to neighbouring components forming a network that can exhibit the desired adaptive behaviour. However, this approach does not necessarily require complexity at the level of the individual components/agents: the complex adaptive behaviour is an emergent property of the system of connections (i.e. network) (C. Anderson & Bartholdi, III, 2000).
		Scalable traffic control	The ability of a network not only to function well in the rescaled/dynamic situation, but to actually take full advantage of it.
		Intelligent and Heterogeneous Agents	Agents can be distinguished by different “rules” defining and/or governing abilities, fitness, goals, patterns of actions, rules of actions, etc. Due to those differences,

Complexity Core Features	Comment	Methods & Systemic Approaches	Description
			most agents composing a CAS are heterogeneous. Lower-level agents may be single physical entities within an enterprise, such as piece goods, machines, containers, or materials (Wycisk et al., 2008). In artificial intelligence, an intelligent agent is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals (see more on section 3.6.2 "Systems' Learning and Artificial Intelligence").
Non-linearity	As explained before complex systems are many times non-linear and unpredictable. Whereas this can be an important characteristic at business strategy level insuring some degree of surprise to competitors, it is disastrous at IS level invalidating attempts to cooperate and interoperate. Complex systems that are not erratic (e.g. CAS), manage to control non-linearity by constantly monitoring their status and the environment, maintaining themselves in a quasi equilibrium status.	Monitoring	Monitoring is the capability to collect information (characteristics and status) about resources (systems or system parts) (Deakin et al., 2006; Morjaria & Santosa, 1996; Snodgrass, 1988).
Quasi-equilibrium		Edge of Chaos	The zone between stability and predictability, on one side, and chaos and unpredictability, on the other (Waldrop, 1992).
		Feedback	Feedback is a circular process of influence where system actions have effect on the system itself, thus it is both an effect of a system's past activity and an influence on its future activity, contributing to reflexivity. A feedback loop is formed where an event occurs in an environment to which a system responds, and that response has an effect back on the environment. Simple mathematic equations of deterministic complexity allow for dynamic behaviour, by incorporating feedback in most of the systematic ideas about the actions of a system in its environment, i.e. learning, adaptation, evolution. A negative feedback loop keeps a system stable while positive is self-reinforcing, and results in one or more variables moving towards a point of no return (Waldrop, 1992; Manson, 2001).
Decomposability	EI is closely related to modelling. Either the full enterprise, products, processes or plain data, models intent to represent and reduce the complexity behind the major enterprise activities. Decomposability also enables the complexity of a system to be reduced into more controllable	Agent-Based Modelling	As described in section 3.5.1 "Agent Based Modelling (ABM)", ABM is used to study the dynamics of systems interactions and to reveal emergent structures and patterns of behaviour. ABM has characteristics that are particularly useful for studying embedded systems, since agent-based models deploy a diversity of agents to represent the constituents of the focal system (Courtney et al., 2008; Gilbert, 2007).
		Exploratory Analysis	Exploratory data analysis is an approach to analyse data sets and summarize their main characteristics in easy-to-understand form, often with visual graphs, without using a statistical model or having formulated a hypothesis (Tukey, 1977).

Complexity Core Features	Comment	Methods & Systemic Approaches	Description
	agents, thus methods applied may be very similar.	Self-similar traffic modelling	Self-similarity is a typical property of fractals which has important consequences for the design of computer networks, as typical network traffic has self-similar properties. Since the internet evidences self-similar properties, instead of developing completely new traffic models, a number of researchers have proposed to adapt traditional traffic modelling approaches to incorporate aspects of self-similarity (Khayari et al., 2004).
Ability to Learn	These complexity topics are very close coupled, thus the methods that apply to one, apply to the majority of them. Feedback is an example. Being one the most important processes used in complexity, it is both an effect of a system's past activity and an influence on its future activity, contributing to reflexivity and emergence	Heuristics	Refers to experience-based techniques for problem solving, learning, and discovery. Where an exhaustive search is impractical, heuristic methods are used to speed up the process of finding a satisfactory solution.
		Feedback	As described above.
Emergence	Deterministic complexity allows for dynamic behaviour, by incorporating feedback in most of the systematic ideas about the actions of a system in its environment, i.e. learning, adaptation, evolution.	Adaptive learning	Through processes of adaptation CAS are able to cope with environmental changes they are interrelated to (Wycisk et al., 2008). In Adaptive learning, the basic premise is that the environment or the system will be able to adjust to the agent's learning method, which results in a better and more effective learning experience that can be used to better achieve its goals
		Genetic algorithms	Genetic algorithms belong to the larger class of evolutionary algorithms, which generate solutions to optimisation problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover (Holland, 1996).
		NK and NKC models	NK models are described by its inventor Stuart Kauffman as a "tunably rugged" fitness landscape. NK model has found application in a wide variety of fields, including the theoretical study of evolutionary biology, optimisation, complex systems, and organisational theory, where it is used to describe the way an agent may search a landscape by manipulating various characteristics of itself. There is a subtle difference between the NK and NKC models since in the first, agents (organisations) are unaware of other agents' behaviour, i.e., they are all walking the landscape independently. In this case, the landscape is fixed and unchanging (although it may be too large to know and to explore fully). In the second (NKC model), the landscapes are coupled and a move by one agent affects the fitness of linked ones (Vidgen, 2008).

Complexity Core Features	Comment	Methods & Systemic Approaches	Description
Self-Organisation		Autopoiesis	Means "self-creation" and expresses a fundamental dialectic among structure, mechanism and function. An autopoietic machine is a machine organised (defined as a unity) as a network of processes of production (transformation and destruction) of components which: (i) through their interactions and transformations continuously regenerate and realise the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in space in which the components exist by specifying the topological domain of its realization as such a network (Maturana & Varela, 1980).
		Power Laws	When the frequency of an event varies as a power of some attribute of that event (e.g. its size), the frequency is said to follow a power law. Typically, power-law functions (e.g., Pareto, Fractals) are polynomials in a single variable, and are explicitly used to model the scaling behaviour of natural processes distributions. Power-law distributions (e.g. Kolmogorov–Smirnov) occur in many situations of scientific interest and have major consequences for our understanding of natural and man-made phenomena. Unfortunately, as defended by Clauset et al. (2009), the detection and characterization of power laws is complicated by the large fluctuations that occur in the tail of the distribution and by the difficulty of identifying the range over which power-law behaviour holds.
Reflexivity		Strange Attractor	Region in n-dimensional space, attracting variables moving according to the dictates of a dynamical system. It accounts for the system's bounded preferences. A strange attractor has zero measure in the embedding space and has fractal dimension (Weisstein, 2011).
		Decent. control	As described above.

3.5. Modelling Complex Systems and Networks

Modelling is sometimes mistakenly thought to enable prediction. It should rather be thought as a means of exploring possibilities, particularly those that would otherwise not be apparent (Rooke et al., 2007). In complexity, modelling has been used mostly in three different ways:

- Through the direct use of complexity concepts to capture the emergent dynamics of the changing state in continuous time. The complex systems approach of doing this is by describing state cycles using mathematical models and running simulations;
- Through Agent-Based computational Modelling (ABM), a branch of computer science that emerged as a methodology for studying complex systems, namely the dynamics of systems interactions, emergent structures and patterns of behaviour (Dekkers, 2009). ABM has characteristics that are particularly useful for studying embedded systems, since agent-based models deploy a diversity of agents to represent the constituents of the focal system (Courtney et al., 2008; Gilbert, 2007);
- And Artificial Neural Networks (ANN) models, inspired by the structure and functional aspects of biological neural networks, usually used to model complex relationships between system's inputs and outputs or to find data patterns.

This section's analysis is focused on the last two, namely on their application to computational systems.

3.5.1. Agent Based Modelling (ABM)

In ABM, the modeller defines the environmental parameters that are of interest as the starting conditions for the particular study, e.g. the instruction: "copy the actions of the neighbour agent".

Computational modellers assume that the aggregate adaptive learning intelligence, capability, or behaviour of an organisation may be effectively represented by millions of nanoagents in a combinatorial search space (like intelligence in our brains appears as a network of millions of neurons), each following a simple firing rule. Repeated runs of the simulated model reveal collective states or patterns of behaviour as they emerge from the interactions of entities over time (Mckelvey, 1999). Moreover, multi-agents systems have the ability not only to study and simulate an environment, but also to be deployed in real life acting as valuable dynamic resources capable of triggering actions within enterprise information systems (possibly agents themselves). Sharing this view and applying it to enterprise management, Benbya & Mckelvey (2006) consider business/IS alignment as a series of adjustments at individual, operational and strategic levels, and suggest several enabling conditions aimed at speeding up the adaptive co-evolutionary dynamics among them. Instead of focusing upon simple cause-effect deterministic logic, they suggest a chain of active-reactive causal dynamics.

Agent-based modelling allows the testing of hypotheses for complex systems in the social sciences and industrial networks, such as supply chains (Gilbert, 2007). The basis of ABM is that each agent acts based on local information, with some dependence on what other nearby agents are doing,

and what they have learned about their environment in a series of time steps. In early models, the rules by which agents acted were often based on ethnography and optimal foraging theory. However, in succeeding time steps, the general sequence of an agent-based computer model is based on the feedback assumption that local action causes environments change, which causes agent reaction (Bentley & Maschner, 2007).

3.5.1.1. Agent

This concept was introduced in computer science in the early 1980s, inspired by the agency theory of economics, as a result of developments in the area of Artificial Intelligence (AI) (Nwana, 1996). Due to being a technical expression tightly associated to novel technologies, defining an agent is not consensual and many different definitions can be found reflecting the views of the different research communities. While it is consensual that every agent should have some sort of autonomy, little agreement goes beyond this. For instance, for some research communities the ability of the agent to learn is a fundamental condition, for others it is almost irrelevant (Barata, 2003).

Today, the term agent is used to refer to anything between a simple subroutine and a conscious entity. There are "helper" agents for web retrieval and computer maintenance, robotic agents to venture into inhospitable environments, agents in economy, etc. Intuitively, for an object to be referred to as an agent it must possess some degree of autonomy, and some kind of identity to be identifiable in its environment (Macal & North, 2010). For the sake of simplicity in this review, an agent is considered a computer-based program located within an environment. It is able to take flexible autonomous actions, based on its goals and perceptions from that environment, in pursuit of its own agenda (maybe by actively interacting with others) and causing changes in that environment (Barata, 2003; Gilbert, 2007).

From this definition, there are three key concepts that are associated to agenthood (N. R. Jennings et al., 1998): (1) *Situatedness*, which means the agent is situated in some environment, from which it receives sensory input and where it can perform actions that affect the environment in some way; (2) *Autonomy*, so that the agent is able to perform its actions without the (direct) intervention of other entities; (3) *Flexibility*, in the sense that the agent should be able to respond to changes that occur in the environment and exhibit opportunistic behaviour. Still, the concept is broad and several taxonomies to classify them exist, according to their attributes and goals, e.g. mobile agents, collaborative agents, holons, personal assistants, etc. (Narasimhan & Hector, 2004).

3.5.1.2. Multi-Agent Systems (MAS)

Although there are situations where an agent can operate usefully by itself, the increasing interconnection and networking of computers is making such situations rare, and in the usual state of affairs, the agent interacts with other agents as in a society of agents.

MAS, also known as collaborative agents or intelligent agents, are the best way to characterize or design distributed computing systems. Centralized solutions are generally more efficient; however, distributed computations are sometimes easier to understand and to develop when the information involved is distributed and resides in information systems that are large and complex. Also, planned information environments are too large, complex, or dynamic to be managed centrally or

via predefined techniques. Each niche of embedded intelligence is best thought of as an autonomous agent that finds, conveys, or manages information.

Due to the nature of environments, agents must be long-lived, adaptive and social (Huhns & Stephens, 1999). As illustrated by Figure 3.5, a MAS is based on the notion of interaction. Multiple agents acting within the scope of an organisation interact and collaborate to perform their predefined activities, however, like any intelligent agent, they have some degree of autonomy, which defines coherency as another essential property of MAS coordination. Coherency ensures that an agent or system behaves as a unit, and on the best interest of the organisation. Without this coordination, agents could be conflicting with one another, thus squandering resources and efforts, and consequently failing to accomplish their objectives (Durfee, 2004). Based on this coordinated interaction, they create impact in the environment under the so-called sphere of influence.

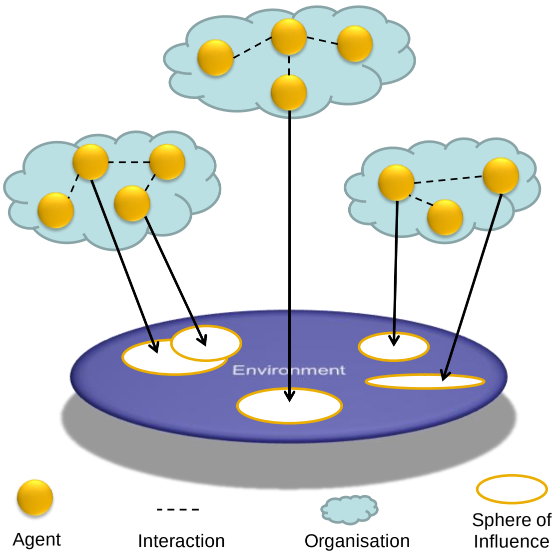


Figure 3.5: Typical Structure of a MAS (based on Wooldridge (2009))

<u>Agent Communication</u>

The study of animals and humans has shown that the most complex, long-lived communities are those that have developed complex communication mechanisms, allowing the establishment of complex interactions. The act of communicating can give to agents the necessary knowledge to execute their actions without being limited by their own viewpoint, as well as to synchronize themselves with other agents. Considering these aspects, it seems unquestionable that complex agent societies can only be created with the aid of Agent Communication Languages (ACLs) (Barata, 2003). However, an excessive use of communications can result in a society of bureaucratic agents, so they need to be efficiently managed. In this sense, an agent must represent its knowledge in the vocabulary of a specified ontology (Huhns & Stephens, 1999), and use agent communication protocols which are typically specified at several levels: the lowest specifies the method of interconnection; the middle level specifies the syntax of the information being transferred; and, the top level specifies the semantics of the information.

While traditional communication mechanisms exchange simple objects with no semantics associated, ACLs exchange propositions, rules, and actions, and at the same time are able to

implement the three levels of communication protocols. Among several ACLs the two most important are: KQML[43] and its many dialects, and FIPA ACL[44].

Concerning popular interaction protocols, the best known and most widely applied is the contract net protocol (Davis & R. G. Smith, 1983), an interaction protocol for cooperative problem solving among agents, providing a solution for finding an appropriate agent to work on a given task, even with the agents' roles not specified in advance. Here, any agent can act as a manager, by making task announcements, or can act as a contractor, by responding to task announcements.

Contract Negotiation

Negotiation is a process by which a joint decision is reached by two or more agents, each trying to reach an individual goal or objective. It is conducted using a common language for communication by the participating agents, a negotiation protocol, and the decision process that each agent uses to determine its positions, concessions and criteria for agreement. Thus, the resultant negotiation mechanism should (ideally) have the attributes of 1) *Efficiency* – the agents should not waste resources in coming to an agreement 2) *Stability* – no agent should have an incentive to deviate from agreed-upon strategies 3) *Simplicity* – the negotiation mechanism should impose low computational and bandwidth demands on the agents 4) *Distribution* – the mechanism should not require a central decision maker 5) *Symmetry* – the mechanism should not be biased against any agent for arbitrary or inappropriate reasons (Huhns & Stephens, 1999).

In the MAS paradigm, negotiation is divided in three consecutive phases: inviting, bidding, and awarding. A contract net protocol defines the basic messages in accordance with these phases and how each message is to be processed. Whenever a task comes to a manager agent, it is decomposed into subtasks and is subject for bid. Subsequently, the manager invites potential subcontractors or suppliers who possess the capability to solve the problem, and after collecting and evaluating all bids, selects the winner agents to execute the tasks (Rogerjiao et al., 2006). This represents a traditional view on negotiation which is based on P2P negotiation. Nowadays, many systems use agent-based collaborative negotiation, which differs from the traditional method in the sense that it enables multi-contract negotiation and coordination to improve overall performance.

Social Order

In a computational environment with interaction of multiple software agents there are elements which enable agents to communicate, cooperate and negotiate while they act in the interests of themselves or their society. Under these conditions it is necessary to consider the individual agent view that needs to interact autonomously with other artificial or human agents (individual benefit) and, at the same time, it is necessary to view the society of agents as a whole (collective benefit) (Barata, 2003).

Therefore, sociability is essential to cooperation, which itself is essential for moving beyond the somewhat rigid client-server paradigm to a true P2P distributed and flexible paradigm that modern

---

[43] KQML - Knowledge Query Manipulation Language (www.cs.umbc.edu/research/kqml/)
[44] FIPA ACL – IEEE FIPA (Foundation for Intelligent Physical Agents) Agent Communication Language (www.fipa.org/repository/aclspecs.html)

applications call for, and where agent technology finds its greatest payoffs. Sometimes it is required to convince the individual agents to sacrifice their interests over the group benefit. Norms and obligations, or incentives and sanctions are examples of mechanisms created by humans to help their interaction. In this context, an interesting aspect of norms is that agents are prepared to lose some autonomy for the benefit of society.

3.5.2. Artificial Neural Networks (ANN)

An ANN is an information processing model, whose design has been inspired by biological nervous systems, as the human brain (Stergiou & Siganos, 1996). It can be represented by a directed graph consisting of nodes as neurons, with interconnecting relationships as synaptic and activation links (Haykin, 2008):

- The basic computational element of an ANN is a neuron also often called node or unit.
- The synaptic links weight the neuron's respective input signals, and the weighted sum ($\sum w_{ij}$ as in Figure 3.6) of the input signals, also called the net input, defines the inducted local field.
- Finally, the activation link (f) flattens the induced local field of the neuron to produce an output (y). Such weights can be adjusted in a learning process to put the ANN answering in a pre-determined form, and its output can serve as input to other units.

A simulated ANN behaves like people. It learns by example, therefore it can be configured for specific applications as data classification or agents patterns' recognition to detect trends that are too complex to be noticed by humans or other computer techniques. However, there are various types of ANNs developed for specific objectives. They vary from those with only one or two layers of single direction logic, to complicated multi–input many directional feedback loops and layers.

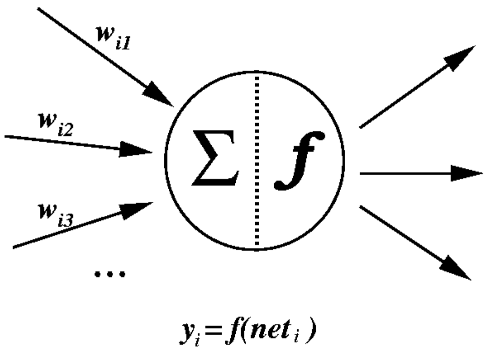


Figure 3.6: A Neuron function (Orr et al., 1999)

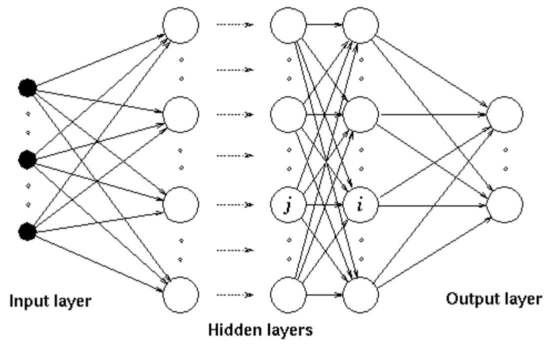


Figure 3.7: Neural Network Example

3.6. *Optimising Man-made Complex Systems*

As analysed, complexity is about the study of emergent order in disorderly systems. Using feedback and learning algorithms it enables systems to progressively adapt to the environment and respond quicker and better to non-linearity, to begin evidencing a certain degree of macro-equilibrium.

By definition, no external controller can be applied to regulate systems behaviour thus they need to be self-aware and responsive to the surrounding stimuli. Living organisms are adaptive by nature and interoperability is innate, but complex information systems need interoperability as means to ensure good feedback capable of triggering proper responses and avoiding erratic behaviour. Therefore, traditional complexity is analytic and does need optimisation; however, when being applied to other areas such as management, production or IS, several technologies can be embedded into systems design to enable optimised behaviour, e.g. monitoring processes, artificial intelligence (AI), change management, model-based systems engineering (MBSE), etc.

3.6.1. Systems Monitoring

A systems approach would find difficulties to model diversity at the level of each system organisation, and would also have trouble getting inside the systems black box to explain how decisions were made. However, the use of artificial adaptive agents enables these issues to be studied in depth.

In this context, monitoring is an important capability to collect information (characteristics and status) about resources (systems or system parts). Generally, monitoring processes include different stages from capturing the information to its elaboration and communication to the user, and are structured into specific components (generally the producer, the consumer and the registry) in order to meet a set of requirements (scalability, extensibility, data delivery models, portability and security). The whole process has to be carried out balancing extensibility and self-description capabilities on the one hand and compactness on the other (Deakin et al., 2006; Morjaria & Santosa, 1996; Snodgrass, 1988).

3.6.2. Systems' Learning and Artificial Intelligence

There is no doubt that one of the major drivers of complexity theory in information systems has been the widespread availability of AI methods, such as neural networks, ABM, and genetic algorithms. These techniques have enabled researchers to populate simulated worlds with multiple intelligent and idiosyncratic entities/agents, and study their behaviour (Davis & R. G. Smith, 1983). Since its conception in the mid-1950s, artificial intelligence with its great ambition to understand and emulate intelligence in natural and artificial environments alike is now a truly multidisciplinary field that reaches out and is inspired by a great diversity of other fields.

The field was founded on the claim that a central property of humans, intelligence, can be so precisely described that it can be simulated by a machine. Thus, machine learning has been central to AI research from the beginning, using both unsupervised learning algorithms to detect specific patterns within streams of input, and supervised learning with classification schemes and numerical regression algorithms to describe the relationship between inputs and outputs and predicting how the outputs should mould to different inputs (Russell & Norvig, 2003).

There is no established unifying theory or paradigm that guides AI research. Many consider it should simulate natural intelligence by studying psychology or neurology and apply a sub-symbolic approach such as neural networks, fuzzy systems and evolutionary computation, while others claim

that intelligence should be reproduced using high-level symbols, similar to words and ideas (Kelley, 2003). Nevertheless, it is agreed that intelligent agents are not limited to simply reacting in predetermined ways to inputs or environmental stimuli. They can learn, make inferences, and plan, thus optimising complex systems (Phelan, 1999).

AI systems can benefit from understanding in the field of artificial neural networks, but other application areas include pervasive/ubiquitous computing and autonomic computing (Saha & Mukherjee, 2003; Kephart & Chess, 2003). This pursues the idea of having networked communication systems able of being autonomously controlled, in a sense that it can manage themselves as a kind of administrators.

3.6.3. Systems Change Management

Change management is the process of identifying and implementing new features to an existing system in order to result in overall improvements and causing minimal disturbances. Throughout the process, the aim is to ensure that traceability of changes remain transparent while minimizing the change impacts on systems normal behaviour.

Among the different components of change management process, one can account for the identification of problems, costs feasibility analysis, analysis of change impact, implementation and testing, followed by a review of change benefits. This optimisation strategy is frequently used in organisational management, maintaining a record of the changes in knowledge repositories using traceability features and evaluating the potential benefits of proposed changes using simulation. This allows informed decisions to be made, which help to justify changes to a system before its implementation (CRESCENDO Partners, 2009a).

3.6.4. Network Optimisation: Model-Based Systems Engineering (MBSE)

Traditionally, in large engineering projects involving large networks, organisations employ a document-based systems approach characterized by the generation of textual specifications and design documents, in hard-copy or electronic file format, that are then exchanged between customers, users, developers, and testers. These documents and drawings represent the systems requirements and the design information that systems engineers control to ensure project's validity, completeness, and consistency (Ogren, 2000). However, despite being rigorous, this method has many limitations due to the inherent complexity of the processes (e.g. relationships between requirements, design, engineering analysis, and test information) and the network itself (relationships between stakeholders).

For these situations, the model-based systems engineering strategy has been introduced with intention of optimising network-centric complex systems' engineering activities using some of the procedures analysed above. MBSE is the formalised application of modelling to support cooperative engineering processes, namely requirements elicitation, product/systems design, analysis, verification and validation activities starting at the conceptual design phase and continuing throughout development and later lifecycle stages (INCOSE, 2007; INCOSE & OMG DSIG, 2011).

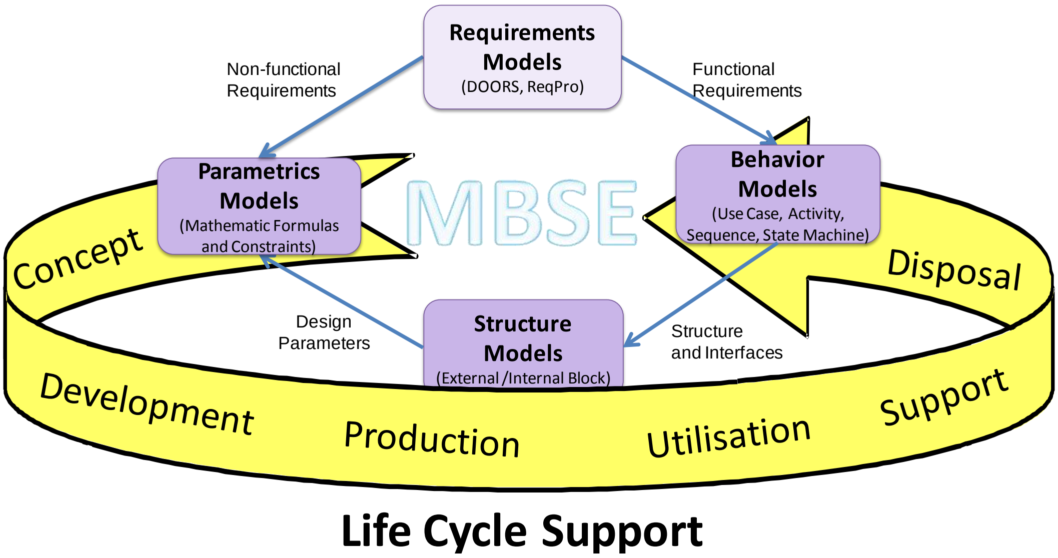


Figure 3.8: MBSE scope (adapted from (Nallon 2003 and Friedenthal et al. 2008))

Figure 2-1 illustrates a possible view on the MBSE process using multiple coherent models to describe a single system within a network of organisations, thus reducing not only potential problems on the project's validity, completeness, and consistency, but also enabling monitoring and change management. This vision merges the full product lifecycle, starting from the concept where the stakeholder's needs are identified until the disposal of the product, with the development of the following models:

- *Requirements models* that represent the relationships between user requirements and/or model objects. A primary benefit of modelling requirements is the opportunity this provides for analysing them with techniques such as requirements animation, reasoning, etc. (Nuseibeh & Easterbrook, 2000). Examples of tools used to construct this models are DOORS, ReqPro or even Excel;
- *Behaviour models* to represent the intended and unintended behaviour of a system of interest (e.g. a product), thus responding to functional requirements. Examples of behaviour models include UML activity, sequence or state machine diagrams;
- *Parametric models* to reply to the non-functional requirements representing the formal relationships and constraints of the system and its components. Tools used for parametrics models include mathematic formulas and constraints; and finally
- *Structure models* which describe the enterprise and system level contexts from both logical and physical viewpoints. These are represented with external and internal block shared by the network of cooperating companies.

In a nutshell, MBSE can provide: (1) enhanced communications, offering more understanding between the development teams and the other stakeholders; (2) reduced development risk, because requirements validation and design verification improve the cost-effectiveness in the development of a system; (3) improved quality, with a more accurate traceability between requirements, design, analysis, and testing; (4) increased productivity, having an more flexible and

easier readjustment of the existing models; (5) and enhanced knowledge transfer, since all partners share the same models (Friedenthal et al., 2008).

By the exposed, if properly integrated, MBSE framework can bring an added value to systems engineering enterprise networks, maximizing the efficiency of collaborations and stimulating interoperability via the models used along the system's and products lifecycle. However, the issues exposed on section 1.1.3 "The Problem of Sustainability of Interoperable Solutions" and also in the grand-challenge raised in the previous chapter (see section 2.8.2.1), regarding sustainability and dynamicity of networks and evolution of requirements, remain pertinent and unaddressed so far.

4. Self-Sustainable Interoperability

In the global market, companies and the networks which they are part of, tend to follow a dynamic and evolutionary behaviour like complex adaptive systems, exhibiting elemental and behavioural properties such as dimensionality and ability to learn (T. Choi et al., 2001; Wycisk et al., 2008)*. They are influenced by the environment, thus awareness of the known landscape, including market dynamism, tends to condition organisation's behaviour, affecting its internal elements, namely the wide variety of internal heterogeneous agents, the ability to interact, or even its autonomy to self-organise in response to external stimuli. When adapting themselves to the market demands, by introducing new requirements and corrections or by adopting new systems and applications, reflexivity arises, impacting back on the environment and establishing a positive self-fed loop among the system and the environment.*

Reaffirming the above statements, adaptation in complex systems also builds complexity (Holland, 1996)*. As a consequence, if not properly monitored, models and semantics can change chaotically, resulting in long periods of network harmonization breaking. This introduces a new dimension to interoperability research:* ***sustainability of systems interoperability****.*

4.1. Interoperability Efficiency Pyramid Model (IPyM)

Analysing the available literature, namely the existing interoperability models and levels (summarized in Table 2.1), it is possible to identify common characteristics and to point out the strengths and weaknesses of each. By and large interoperability is a broad and complex subject and the development of specific solutions is difficult. However, it is acknowledged that assessing interoperability with well-chosen measures is essential to identify priorities to be considered when establishing business networks and partnerships.

Traditionally, system's design and network interoperation is seen as a "black box", but concerning evaluation, and following the vision of Yahia (2011), interoperability needs a more detailed and accurate "white box" analysis to see what is happening inside collaboration networks and enterprise systems in terms of relationships. Yet the existing maturity models do not relate EI practices in terms of efficiency of the network (normally only the system is analysed), thus a different classification scheme with different evaluation parameters is proposed across five different levels of network **efficiency** (see Figure 4.1):

- *slack interoperability*,
- *unregulated interoperability*,
- *standard-based interoperability*,
- *semantic interoperability*, and
- *sustainable interoperability* which is the focus of this PhD's research.

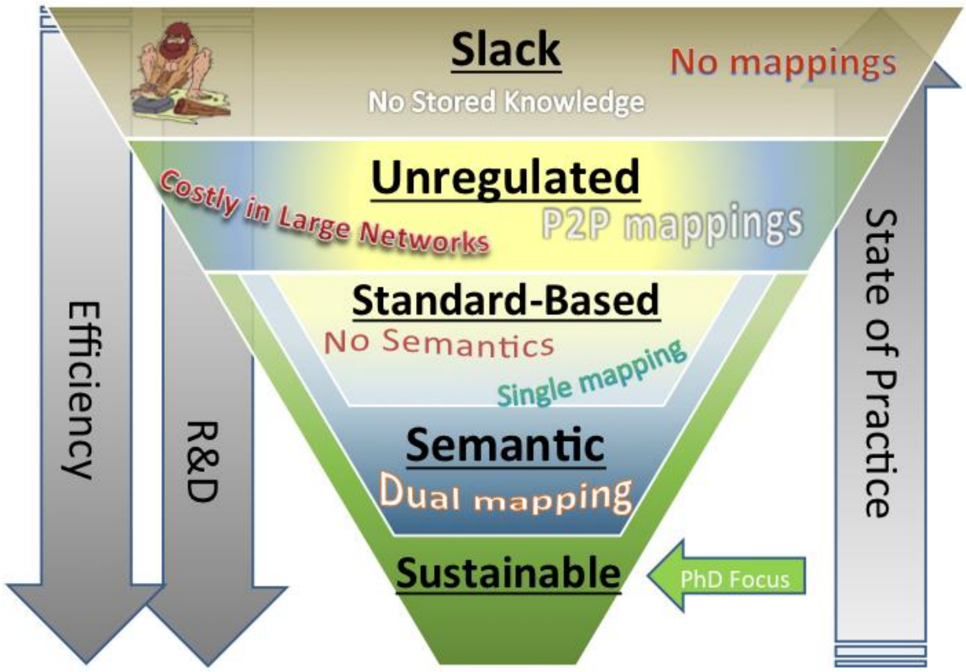


Figure 4.1: Interoperability Efficiency Pyramid

Efficiency describes the extent to which a resource, e.g., time or effort, is well used in the execution of a specific task to reach a goal. Although the term efficiency has varying meanings in different disciplines, the term is here used to evaluate the amount of time spent in enabling a specific system to become and remain interoperable with all the other systems in an enterprise systems' network. Hence, it relates to the degree of human intervention in the communications among different systems, i.e. the less time a system needs of human intervention to become enabled for exchanging data with another system, the more efficient the interoperability practice is. For instance, systems that need telephone interaction among their users to clarify what has been exchanged in a specific electronic message, evidence a low efficiency rate of interoperability when compared with situations where communication is performed seamlessly without any human intervention.

Unfortunately, as evidenced in Figure 4.1, the research and technological development that would allow that automation is inversely proportional to the industrial adoption (state of practice). Normally, research takes some time (frequently many years) to penetrate the market, thus the lower categories of the inverted pyramid, i.e. standard-based and semantic levels, can only be found at large or very mature companies. Also, being a novel concept, *sustainable interoperability*, as addressed in this dissertation, is still at academic level.

In the next sub-sections all levels are detailed.

4.1.1.1. Slack Interoperability

Most of the information being exchanged through the internet is not much more than untagged text, which might be acceptable for human reading or for large-scale electronic libraries, but is useless for e-business (Berners-Lee & Fischetti, 1999). In fact this is exactly the opposite of what is required to achieve systems interoperability. Bad cases are frequently reported, e.g. the one identified in Watson (2008) concerning the inefficiency of the automotive industry supply chains where "*goods are still handed-off through faxes, phone calls, paper documents, and a wide range of proprietary electronic exchanges"*, or the one in FunStep (1998) regarding the state of the art in European furniture sector describing even worse usage of computers and available ICT. Despite not being categorized as so in the examined literature, the term "slack interoperability" can be related to some form of LISI's "Connected Interoperability" (Table 2.1), and is proposed to encapsulate all the attempts organisations are making to use ICT as a support to their businesses, but with little efficiency in terms of e-business, and reduced return on investment.

Therefore slack interoperability designates all network communication sets (*Cs*) where there is no previous understanding between sender and receiver. All messages (*m*) exchanged between two enterprises require a corresponding "request for clarification" (*rc*) from the receiver side (eq. 2), a response from the sender (*r*) and, in some cases, a set of actions involving human intervention(*ha*).

$$\forall m \in Cs, \exists rc, r \in Cs\colon m \Rightarrow \mathrm{rc} \wedge \mathrm{r} \qquad (2)$$

<u>Becoming interoperable</u>: The "slack" level does not really enable an interoperable network. With no previous understanding between sender and receiver and no knowledge stored within the systems' internal processes for future communications, the effort to start a relationship is large and follows the same metrics as to maintain communications.

<u>Efficiency during communications</u>: All systems following this pattern are highly inefficient since the total time spent on the communications ($\Delta t(Cs)$) among two enterprises is given by the sum of four operands, and as expressed in eq. 3 (where *n* = total messages exchanged), is increased with the time spent on the clarifications, responses and human interventions. Slack interoperability does not improve over time, and subsequent communications between the same enterprises result in similar costs, i.e. only some tacit knowledge is kept and explicit knowledge is not stored.

$$\Delta t(Cs) = \sum_{i=0}^{n} \big(\Delta t(m_i) + \Delta t(rc_i) + \Delta t(r_i) + \Delta t(ha_i)\big) \qquad (3)$$

4.1.1.2. *Unregulated Interoperability*

Enterprises are traditionally focused on P2P relationships, thus disregarding the overall collaboration need of the network, and tend to use proprietary data formats and business rules, handling as many mappings ($mapOrg(x)$) as the number of direct partners (*x-1*, where *x* is the number of organisations) (see eq. 4). Unregulated interoperability is, therefore, a category proposed to classify these P2P interoperability practices, where every e-message (*m*) exchanged between two different enterprises requires an initial effort to establish a model morphism (*MoMo*), namely a mapping (*map*) among the models used by both (eq. 5).

$$\forall x \in \mathbb{N} : mapOrg(x) = x - 1 \tag{4}$$

$$\forall m \in Cs, \exists map \in MoMo : m \Rightarrow map \tag{5}$$

$$\forall x \in \mathbb{N} : mapNetwork(x) = \begin{cases} 0, & x = 1 \\ mapNetwork(x-1) + (x-1), & x > 1 \end{cases} \tag{6}$$

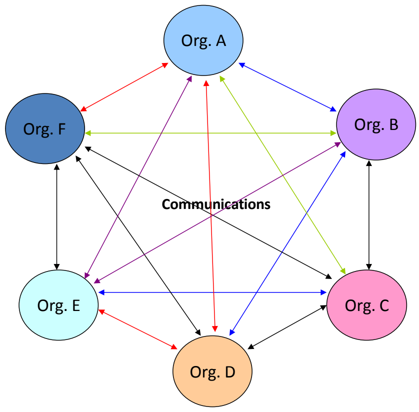


Figure 4.2: Unregulated Interoperability

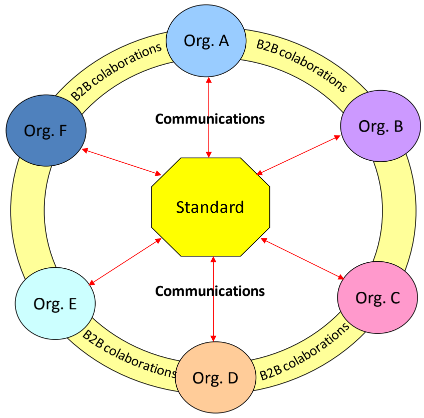


Figure 4.3: Standard-Based Interoperability

<u>Becoming interoperable</u>: Figure 4.2 illustrates an extreme case where all organisations need to communicate with all others, and each time a new enterprise enters the collaborative network causes an increase of the total number of mappings (*mapNetwork(x)*) as given by eq. 6. Nevertheless, unlike slack interoperability, explicit knowledge regarding the transformation (the mapping) is stored within the enterprise systems. Hence the major time consumption is concentrated at the first interaction, i.e. at the mapping establishment ($\Delta t(map(x))$).

$$\Delta t(map(x)) = \sum_{i=1}^{mapOrg(x)} \Delta t(map(x)_i) \tag{7}$$

Unregulated interoperability is quite efficient for networks of two or three enterprises, such as in NC3TA's RIM "Unstructured Data Exchange" or OIM's "Ad hoc" level (see Table 2.1). However, it becomes unbearable for large networks, demanding a considerable financial effort, and producing an excessive workload for the architects defining the mappings required for communications.

<u>Efficiency during communications</u>: Comparing the total time spent on regular communications (eq. 8) with the previous eq. 3, one might think that it has increased. However, that is not true since the request for clarifications (*rc*), corresponding responses(*r*) and human interventions (*ha*) are

significantly reduced, being '*k*' a limit lower than the total number of messages (*n*) in the communications among any two enterprises of the same network.

$$\Delta t(Cs) = \sum_{i=0}^{n} \Delta t(m_i) + \sum_{i=0}^{k} \big(\Delta t(rc_i) + \Delta t(r_i) + \Delta t(ha_i)\big), and\ k < n \quad (8)$$

4.1.1.3. Standard-based Interoperability

The use of reference models, as suggested in MDE (section 2.3.1 "Model-Driven Architecture"), and standardization rapidly became an evident priority, adding efficiency to unregulated interoperability. Several dedicated reference models covering many industrial areas and related application activities, from design phase to production and commercialization, have been developed enabling industrial sectors to exchange information based on common models (BuildingSMART, 2010; FunStep, 1998; ISO TC184/SC4, 1994; Mason, 2007).

When using standards or conceptual models as the reference format for information exchange (Figure 4.3 above), enterprises only need to be concerned with a single morphism (eq. 9), i.e. the one describing the mapping among its internal model and the reference one. Therefore, for each message (*m*) exchanged between two enterprises, a mapping (*map*) is still required (eq. 5), but comparing with unregulated interoperability, the major advantage concerns the total amount of mappings within the network. As expressed by eq. 10, *mapNetwork(x)* is given by the number of organisations in the collaborative network (*x*).

$$\forall x \in \mathbb{N} : mapOrg(x) = 1 \quad (9)$$

$$\forall x \in \mathbb{N} : mapNetwork(x) = x \quad (10)$$

<u>Becoming interoperable</u>: With standard-based interoperability, the collaboration effect is maximized and, when a new enterprise enters the network, it only needs to do a one-time effort of integration with the reference model. Following the previously defined eq. 7, the time spent with this operation only needs to tally the time for the definition of a single mapping, as $mapOrg(x)$ is always 1. Also, this mapping has the benefit of not requiring the involvement of the other network members, leaving them undisturbed.

<u>Efficiency during communications</u>: Comparing the total time spent within enterprise communications, given by eq. 11, with eq. 8 from unregulated interoperability, one might think that it is the same. However, that is not true since the request for clarifications (*rc*), corresponding responses(r) and human interventions (ha) are significantly reduced, being '*j*' a limit lower than '*k*', which is also lower than the total number of messages in the communication (*n*). This level can be compared with LISI's "Domain-based interoperability" (see Table 2.1), where all the clarifications are related with domain semantic issues, not with the syntax or data structures, since they follow the same format.

$$\Delta t(Cs) = \sum_{i=0}^{n} \Delta t(m_i) + \sum_{i=0}^{j} \big(\Delta t(rc_i) + \Delta t(r_i) + \Delta t(ha_i)\big), and\ j < n \quad (11)$$

4.1.1.4. Semantic Interoperability

It is at this level that becomes important to remind that interoperability is not a characteristic exclusive to ICT systems. On the contrary, it should be homogeneous throughout the network, crossing all enterprise levels, from the human knowledge, to business processes, down to plain data (Athena IP, 2007). Unfortunately, this is not completely accomplished yet, neither addressed by standard-based interoperability.

The authors refer to this envisaged level of interoperability as semantic interoperability, defined by two kinds of knowledge: tacit knowledge, which people carry in their minds, providing context for people, places, ideas, and experiences; and explicit knowledge, which has been or can be articulated, codified, and stored in certain media, e.g. a standard (Nonaka et al., 2001). It is because of the first, i.e. the human knowledge involved in businesses, that the previously described interoperability practices still require requests for clarification, and human actions need to be accounted in the total communications time. As an example, each stakeholder can have its own nomenclature and associated meaning for their business products. Therefore the information exchanged, even if sharing the same structure as in the standard-based interoperability, still may not be understood by all business partners.

Semantic annotation, semantic enrichment and knowledge mediation using domain ontologies are the current state-of-the-art research to address the above issues (Boudjlida & Panetto, 2007; Franconi, 2004; Jardim-Goncalves et al., 2009; Missikoff et al., 2003). Only when these challenges are accomplished, complemented with the usage of a standard, and applied in industry, seamless interoperability will become a reality. This way, the usage of semantic interoperability does not require the adoption of a common data exchange standard, but efficiency in communications is maximized if that is the case. That situation is the one considered for further analysis.

$$\forall m \in Cs, \exists map, ontomap \in MMs : m \Rightarrow map \wedge ontomap \quad (12)$$

$$\forall x \in \mathbb{N} : mapOrg(x) = 2 \quad (13)$$

$$\forall x \in \mathbb{N} : mapNetwork(x) = 2x \quad (14)$$

<u>Becoming interoperable</u>: Following the above premise, and as evidenced by equations 12, 13, and 14, the number of mappings within the network will be higher than in standard-based interoperability. This is because in addition to the morphism expressing the mapping (*map*) between the organisation's model and the standard model, there is also the need to map (*ontomap*) the organisation's semantics to a domain ontology, which is common to the collaboration network (Jardim-Goncalves, Silva, et al., 2007; Sarraipa et al., 2010).

$$\Delta t(map(x)) = \Delta t(map(x)_{standard}) += \Delta t\big(ontomap(x)_{ontology}\big) \quad (15)$$

<u>Efficiency during communications</u>: Despite the increase of time expressed in eq. 15, the total time spent on the communications between two enterprises (given by eq. 16) is heavily reduced when comparing with the previous practices, since having the domain semantics integrated, clarifications will no longer be required, and the communications set (*Cs*) will become fully automated. If

semantic integration is done with other type of interoperability than the standard based, then the equation below might not apply.

$$\Delta t(Cs) = \sum_{i=0}^{n} \Delta t(m_i) \qquad (16)$$

4.1.1.5. Sustainable Interoperability

Following the need organisations have to adapt themselves according to new market demands and to availability of new systems and applications, sustainability of systems interoperability appears as the natural evolution of the enterprise systems interoperability pyramid model since none of the above layers are concerned with the loss of interoperability after it has been established. It is a new research dimension that following the IPyM model presents itself as the successor of semantic interoperability in terms of R&D and efficiency and its predecessor in terms of state of practice due to the little research available.

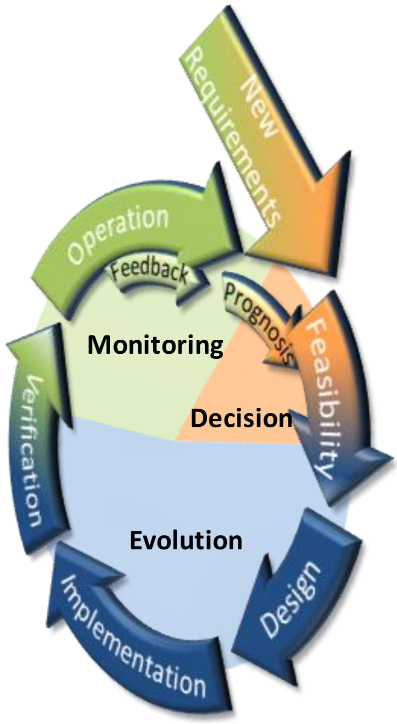


Figure 4.4: Adaptive Organisation's Lifecycle

The definition of "sustainability" has been closely aligned with improving the quality of human life while living within the carrying capacity of the supporting eco-systems. In fact, the term has proliferated in association with the Earth's preservation and soon became common in industrial domains, with policy makers constantly voicing the need to reconcile economic interests with social and environmental needs (the three pillars of sustainability (Adams, 2006)). However, the concept of sustainability applied to EI, as described in this level of interoperability, targets advances of a more technological sort, and categorizes enterprise systems and networks that are considering interoperability along the adaptive organisation's lifecycle illustrated in Figure 4.4, especially during the "operation" stages subsequent to an evolution cycle.

As explained before, complex systems are reflexive by nature, thus if the system evolution is not carefully addressed, the entire network can suffer the consequences endangering the collaborative effect. Figure 4.5 and Figure 4.6 help understanding this issue by looking at the network of organisations as a "molecule" of adaptive internal systems (nodes). Each of them has its own requirements and lifecycles and, depending on the nature of the evolution, can either continue interoperable with its business partners as it was, lose connectivity with some of the nodes (Figure 4.6 example) or, depending on the importance of the node on the whole network behaviour, even invalidate the network.

As in the case of semantic interoperability that does not require the adoption of a common data exchange standard, also sustainable interoperability does not require semantic interoperability, and can be implemented directly on top of unregulated or standard-based practices (see Figure 4.1).

Efficiency in communications is, however, maximized if that is the case, thus it will be the situation considered for further analysis.

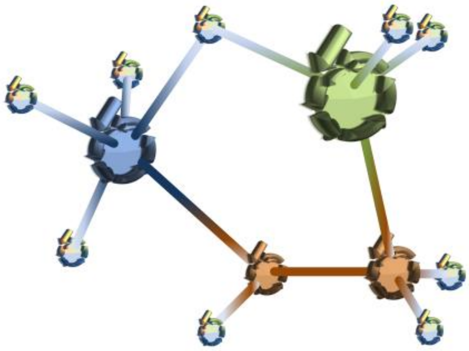


Figure 4.5: Network as a "Molecule" of Adaptive Systems

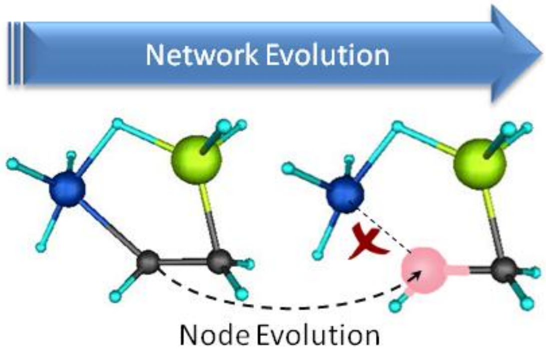


Figure 4.6: Example of a Network Evolution following a Node Evolution

Becoming interoperable: The equations defined for the semantic interoperability level are valid and applicable to sustainable interoperability. No more mappings are envisaged and the organisations entering the network will have to go through similar processes at similar efficiency costs as before. Only the methods to do so might change.

Efficiency during communications: So far this efficiency topic has been addressed only considering the normal flow of the collaborative network and, in that situation sustainable interoperability will also be as efficient as semantic interoperability (in the case it is implemented on top of it). In any other case, it will be equivalently efficient to the level upon it is implemented on.

The difference in the sustainable interoperability level is, when interoperability is compromised after a node evolution, it is more efficient since it is the only level considering a semi-automatic response to that situation. As developed in the next sections, less human intervention is envisaged and the following definition applies.

Sustainable Interoperability: *Interoperability that meets the needs of the present, without compromising the ability of future changes to meet new requirements, maintaining interoperability with adequate adaptation and suitable management of the transitory elements.*

4.2. Conceptual Solution to Enable Hypothesis

In a simplified form, the hypothesis presented in section 1.2.4 suggests that sustainability of systems interoperability could be achieved if a system node (organisation within a collaborative network) behaves like a complex adaptive system and self-organises in response to both internal and network-level changes. As argued during chapter 3 that happens almost innately. Organisations are emergent forms of several other complex agents, including humans, that acting together enable that response. However, they provide no insurances of timing, which by the exposed in the last level of IPyM, may have consequences to the overall collaboration network – even its dissolution.

The motivation behind this thesis is, therefore, to improve that innate capacity to an efficient level, complementing the organisations' information systems with extra intelligence in the form of expert agents dedicated to monitoring certain types of network-related data and triggering the appropriate actions. As hypothesised, if the model morphisms describing the system's relationships with other nodes were carefully monitored, interoperability harmonization breaking (see section 4.2.2 "Harmonization Breaking") could be detected and interoperability (re)established smoothly.

Figure 4.7 illustrates how the hypothesis has been translated to the form of an implementable solution, identifying the major processes, data flows, and research components. To highlight the commonalities, the same image used to demonstrate a CAS model in section 3.2.3, was used as a basis for the concept specification. Here, a system is represented within a networked environment composed of other interlinked systems that, from a macro perspective, are seen as "molecules". The internal CAS-like organisation depicts the process stages of the adaptive organisation lifecycle (also in Figure 4.4), going through:

- *Monitoring* processes, where during regular system operation, feedback is received from the networked environment;
- *Decision* processes, learning how human decision makers normally respond to unforeseen situations like new customer requirements or new market demands; and
- *Evolution* processes, where the organisation actually self-organises in face of the decisions taken, and produces an effect on the collaboration network it belongs to, initiating the *adaptive loop*, feeding the system new input information and triggering the adaptive organisation lifecycle again.

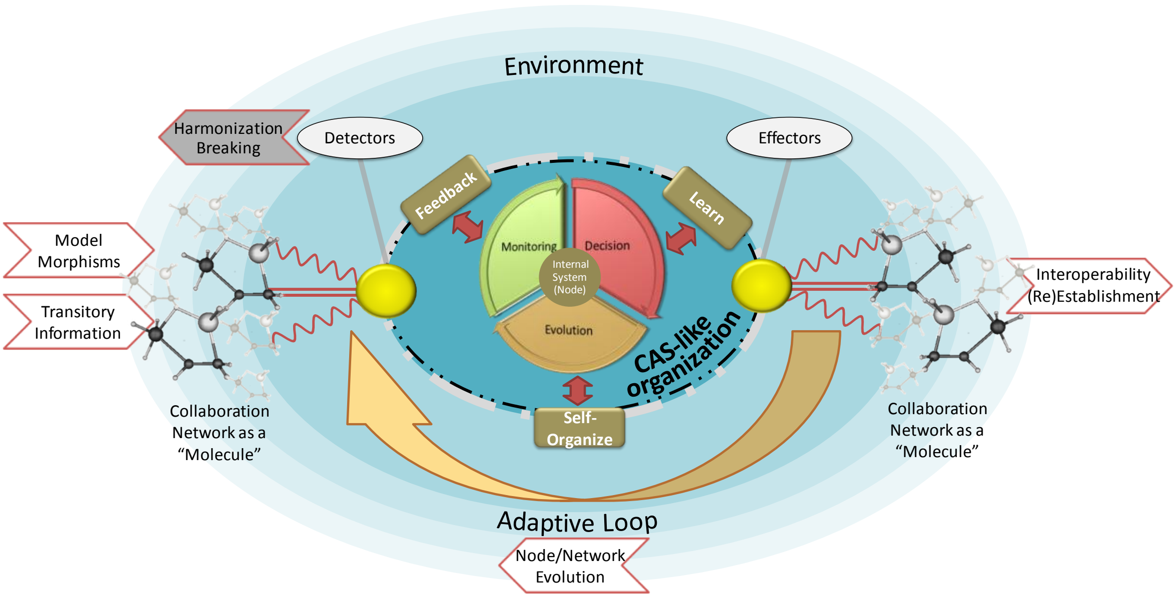


Figure 4.7: Self-Sustainable Interoperability Conceptual Solution

From a workflow point of view, the adaptive loop has actually two different entry points as evidenced in Figure 4.8. It can either be as described above, through a change in the environment (1) that leads the organisation to an unexpected adaptation (2), or through a change at system level (A) which can be consequence of the actions of (2) or not, that creates positive feedback into

the environment causing the remaining networked systems (also CAS) to react (B) accordingly. In this last case, the adaptation of the network as a whole is achieved by several sub-self-organisation processes triggered at system level after an interoperability harmonization breaking detection.

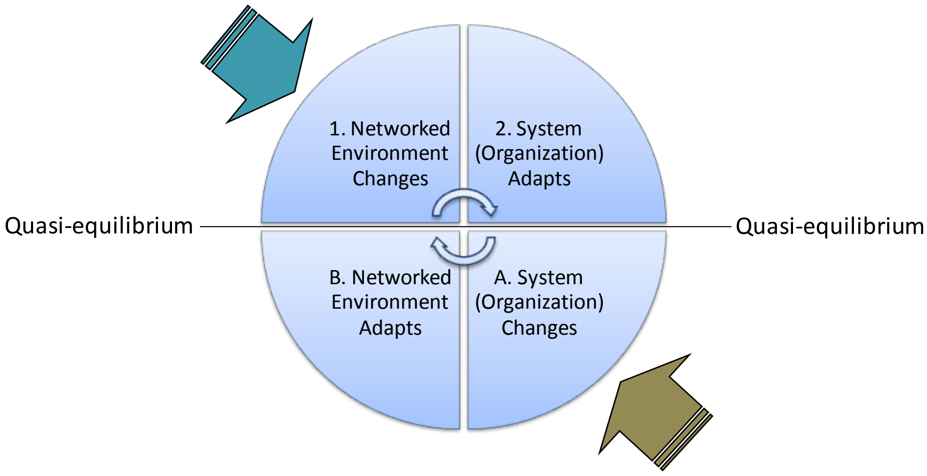


Figure 4.8: Adaptive Loop Entry Points

This oscillation among a state of network equilibrium, where organisations are operating normally, and non-equilibrium, characterized by non-linear behaviour, is designated within complexity as "quasi-equilibrium". With the conceptual solution proposed, the objective is to minimize the transitory and maximize the time networks are operating linearly profiting from a maximum efficiency from the interoperable relationships. Thus, with the existing mapping morphisms and the transitory information regarding the other nodes adaptive cycles, provided as system inputs (Figure 4.7 above), the intelligent expert agents proposed are able to accelerate the adaptive organisation lifecycle and sustain interoperability.

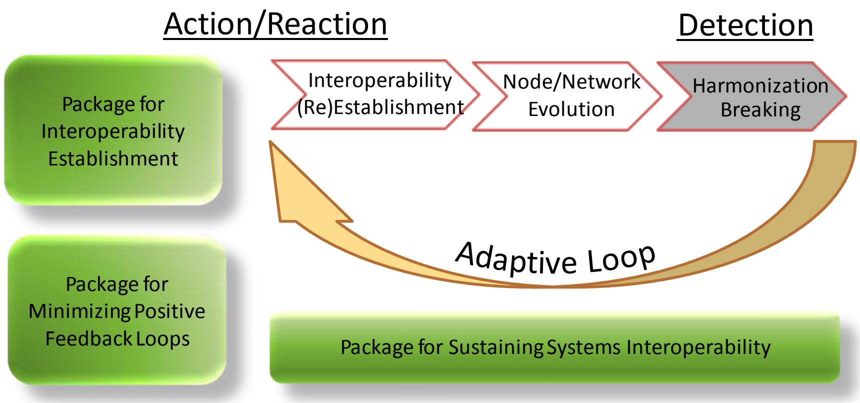


Figure 4.9: Packages composing the Framework for Self-Sustaining Interoperability

To conclude, the framework for self-sustaining interoperability is divided into three research and development packages that address the action, detection, reaction and processes described (Figure 4.9):

1. The *package for interoperability establishment* (action) enables the entire framework by providing the initial knowledge that supports the solution. This package helps defining the

necessary mappings for an organisation to enter a collaborative network and become interoperable;

2. The *package for sustaining systems interoperability* (detection and reaction) is responsible for managing the system's detectors and effectors, triggering the adaptive organisation lifecycle in response to harmonization breaking;
3. The *package for minimizing positive feedback loops* (reaction) is a decision support package that complements the activity of the previous one preparing auxiliary information (e.g. transient period, impact simulation) that decision makers can use to help in the choice of the best adaptive actions.

The harmonization breaking concept and all packages are contributions developed throughout this PhD work and will be further described in the following sections.

4.2.1. Interoperability Establishment

As analysed in the state of the art review, modelling is becoming the primary enabler for complex system design and engineering (Mosterman & Vangheluwe, 2004; INCOSE, 2007). Models represent knowledge in a uniform way at various abstraction levels allowing an automated and shared vision on the information systems. However, since standardization has not yet proliferated among all types of enterprises, namely SME's, different kinds of models and modelling technologies tend to exist in the same business environment which may lead to P2P integrations as described in the unregulated interoperability of the IPyM.

In this context, to enhance interoperability and business collaboration networks establishment through information model integration, organisations require mechanisms capable of abstracting the model from the technology (language) in which it is described. This happens because enterprises need to detach from technology details and focus on managing and planning of their business, i.e. defining the mapping morphisms. If that would be the case, more organisations could enlarge their business networks without having to make huge investments on specialised personnel and tools to manage technologies they are not aware.

To handle this situation while preparing ground for a self-sustainable interoperability, a modelling language independent framework for interoperability establishment is proposed for implementation, using MDA (Figure 4.10). With the language obstacle out of the way, organisations will become capable of establishing gradual P2P, standard or ontology-based mappings, on a need-to-serve basis, independently from the language their information models are described on, and from the number of business relationships within the collaboration networks they are part of. This means that organisations continue to use their legacy software and models (at the bottom of Figure 4.10), without needing to adjust internal structures to each organisation they want to seamlessly collaborate. Instead, the approach used in the framework relies on companies applying an interface to their output models, i.e. a meta-model profile which is achieved through a common contact modelling language harmonization layer (depicted at the centre of Figure 4.10) acting as a modelling language translator for all organisations' models. Hence, each enterprise does not need to know how to relate their models to the modelling language specificities of the other companies'

models - they just have to focus on how to correctly link themselves to the interface and generate harmonizable models, i.e. Language Independent Models (LIMs).

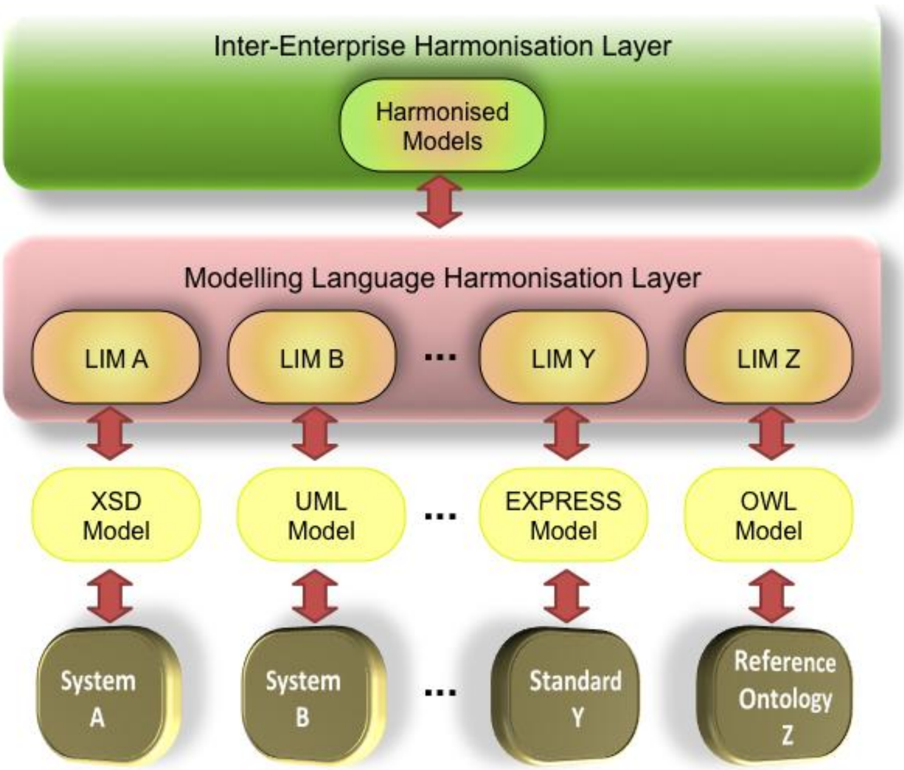


Figure 4.10: Modelling Language Independent Framework for Interoperability Establishment

Once these LIMs are generated, the inter-enterprise harmonization layer is responsible to establish another level of translation (top of Figure 4.10). Here, not only the models from different sources are mapped to obtain model transparency, but also semantics are analysed, adjusted and stored in dedicated morphisms knowledge bases to finalise the process of interoperability establishment in the preparation for sustainability. Once the models are in fact integrated in the top-layer of the framework, they can finally be exported down to any desired compatible source which is already connected to the framework. More information can be found at section 4.3 "MDA-based Package for Interoperability Establishment", where the package is technically detailed.

The proposal depicted in Figure 4.10 is inspired by ISO/IEC 11179 Metadata Registries (MDR) (ISO/IEC, 2004). It describes the standardising and registering of data elements to make data understandable, enabling the creation of a shared data environment in less time and with less effort than it takes for conventional data management methodologies. The interpretation of ISO/IEC 11179 here used, tried to simplify the enterprises' adoption process and maintain the overall time and money spent as low as possible since it does not intend to overcome the need for adopting product representation standards such as ISO 10303 – STEP for exchanging information. On the other hand, since this conceptual approach does not have a specific domain of action, it is possible to embrace several domains (e.g. aeronautics, furniture, automotive, etc), enabling different domain enterprises to communicate and collaborate.

4.2.2. Harmonization Breaking

It is a well-established fact that in classical sciences, like physics, certain phenomena can be described in exactly the same way even if experiments are carried out under different observational circumstances. This means that the laws describing the phenomena display similar results for similar inputs, i.e. symmetries, or symmetric behaviour. Yet, experiments have proven that in extreme circumstances small fluctuations acting on a system may cause it to cross a critical point and evidence an unexpected or chaotic behaviour. This disorder is designated as symmetry breaking (see Figure 4.11), and is a characteristic of complex behaviour (Nicolis & Prigogine, 1989).

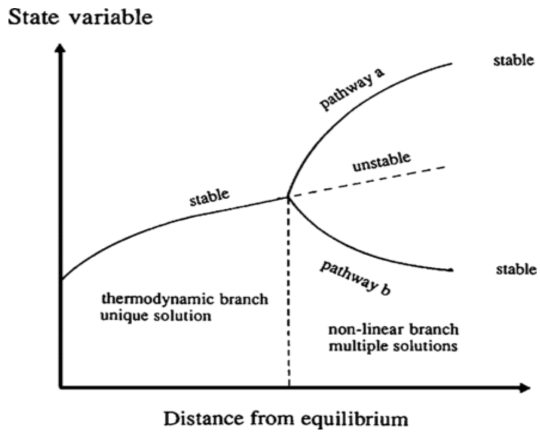


Figure 4.11: Symmetry Breaking (Nicolis & Prigogine, 1989)

Drawing a parallelism with the collaboration/business networks addressed in this PhD, the behaviour evidenced by organisations can be characterized similarly. After interoperability is first established (e.g. using the modelling language independent framework), the set of organisations within a network demonstrate stability exchanging e-messages following established laws (i.e. mapping morphisms). Therefore, at this point, networks display symmetry and organisations are operating regularly. However, as explained before, that may change according to the environment. If just one of the network members adapts to a new requirement, the harmony is broken, and the network begins experiencing interoperability problems. Therefore, “harmonization breaking” is a new term proposed for the interoperability domain, equivalent to the “symmetry breaking” term, from classical sciences.

4.2.3. Sustaining Systems Interoperability

By the previously exposed, complexity science has been largely used as an analytic framework for organisational management, and recently has also been acknowledged as a framework for the design of information systems (Courtney et al., 2008; Merali, 2006). It offers a powerful set of methods for explaining non-linear, emergent behaviour in complex systems, such as CAS which are presumed to be capable of autonomously adapt to environmental changes. However, some available literature makes very clear that CAS result in non-linear behaviour with some probability of butterfly events spiralling into positive and negative extremes (Wycisk et al., 2008). To avoid that, and ensure a rapid response to harmonization breaking, context awareness is demanded in

support of intelligence both at network and systems level (integration and interoperability intelligence layer).

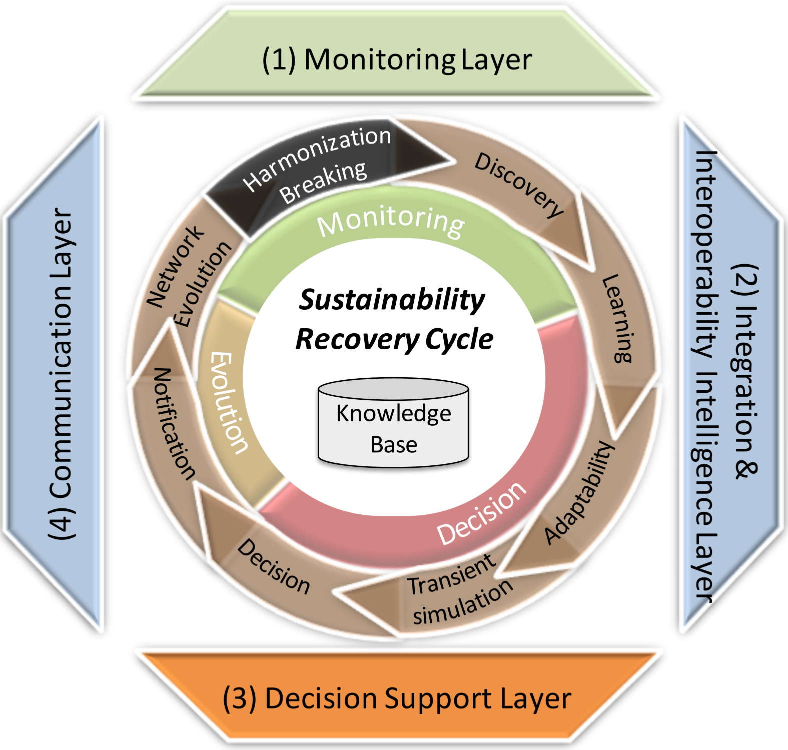


Figure 4.12: CAS-based Framework to Support Sustainable Interoperability (CAS-SIF)

Figure 4.12 presents a CAS-based framework to support sustainable interoperability. It envisages a sustainability recovery cycle along the adaptive organisation lifecycle (monitoring, decision, evolution), considering monitoring and decision support capabilities in the specification of a framework to implement the conceptual solution's second package. This framework must exhibit:

- *Discovery capabilities*, detecting when enterprise systems are updated in the network, driving into harmonization breaking;
- *Learning and adaptability*, i.e., after detecting the harmonization breaking a learning process should be triggered to acquire knowledge about the changes occurred and the nodes adaptation required, by means of morphisms knowledge bases defined at interoperability establishment. It should enable the adaptation of systems and the optimisation of the maintenance process, using knowledge representation technologies, applied to the model management domain, namely dynamic MoMo;
- *Transient simulation* (explained in detail in the third package for evaluating adaptations impact) *and decision*, to understand how a network, as an integrated complex system will suffer during the transient period, and how adaptations, if implementation is decided, affect the overall system and network behaviour;
- *Notification/Communication*, informing in what way should the network nodes react, so that they obtain information for their own needed adaptations and the information system, as

well as the entire network, evolve for the interoperable state (equilibrium) as swiftly as possible.

More information can be found at section 4.5 “MAS Package for Sustaining Systems Interoperability”, where this package is technically detailed using a multi-agent architecture.

4.2.4. Minimizing Positive Feedback Loops

As explained along Figure 4.8 “Adaptive Loop Entry Points”, any type of disturbances may move the system our of the “quasi-equilibrium” state, leading to costly and unexpected behaviour. Systems within collaboration networks are sometimes subject to on-line changes during finite periods of time, which occur when the environment is altered. On these occasions system parameters (e.g. models, interfaces, etc.) need to be reorganised in such a way that the system can remain interoperable. However, since enterprises are complex and adaptive systems, intervention strategies on the evolution of a network node (micro-level – enterprise system) affect the network interoperability sustainability (macro-level) (see Figure 4.8). Thus, before proceeding with application’s changes at information model and knowledge levels, as proposed in CAS-SIF integration and interoperability intelligence layer, organisations need to analyse such modifications carefully.

Swayed to factors that are making interoperability difficult to sustain over time, it is important to provide CAS-SIF with a methodology to simulate and analyse transient data, generated by harmonization breaking and corresponding solutions, in business collaboration networks. This would enable decision makers to evaluate the impact (including costs) of specific interventions for both enterprise and business partners, increasing the confidence that adaptations do not jeopardize potential benefits.

Transient analysis is a technique used to study the response of a system under the action of any time dependant disturbance. This type of analysis is normally used to determine the time-varying displacements, anomalies, stresses, or forces acting on the system under study, as it responds to any combination of static, transient and harmonic disturbances. Being a nonlinear differential equation, the solution is a waveform built upon an infinite-dimensional continuum of points, and the best one can hope for is to simplify for a finite-dimensional approximation to the actual solution, e.g. a finite sequence of points (Kundert, 2003). This way, in transient analysis, time is discretized into small time steps and the solution is simulated for each of them. Being frequently applied to electrical circuits, energy or to building structures for studying their behaviour when put on stress, the same concept could be adapted and applied to the problem under study (Calvano et al., 1999; N. J. Wang et al., 2004; Ledesma & Usaola, 2005), as elaborated on section 4.6 “Transient Evaluation Package for Minimizing Positive Feedback Loops”.

4.2.5. Conceptual Solution Coverage of the Identified Challenges and Research Questions

Table 4.1: Coverage of the Identified Background Challenges and Research Questions

Background Challenge		Coverage of PhD Research			Justification
		Addressed	Research Question	Alignment **(Full, Partial, Superficial)**	
Grand	Innovation and developments on manufacturing, service and ICT-related enterprises should envisage the development of new methods and infrastructures to **sustain** information models, standards, tools, and interoperability, thus aligning economic interests with social and environmental needs, and supporting their eco-systems.	Yes	RQ	Full	Research focus is fully aligned, addressing new methods and infrastructures to sustain EI: *Self-Sustainable Interoperability*.
1	Interoperability should be a "transparent" capability supporting businesses, assured by ICT services but not perceived to users, i.e. working in the background of enterprise systems.	Yes	Q1.2	Partial	Semi-automatic services to sustain EI are targeted, but human intervention is still needed (e.g. decision support layer of CAS-SIF).
2	Stakeholders should be able to implement the several standards required in their businesses, independently of the technology they use, and without prejudice to the network harmonization and the subsequent interoperability effect provided by the standard.	Yes	Q1.2 Q1.3	Partial	Primary research focus is not on EI establishment. But, due to the multidisciplinarity of possible evolutions, strategies to equip enterprise systems with new tools/methods to support multiple languages and technology are addressed: *package for interoperability establishment*. Also, transient analysis is envisaged to ensure minimal prejudice to the network harmony: *package for minimizing positive feedback loops*.
3	Information systems, reference data models and standards should be gradually updated to match disruptive enterprise and market innovation.	Yes	Q1.1 Q1.2	Full	Related to challenge 2 but focus is extended to include monitoring capabilities to know when to trigger the response (only information system models are addressed, though): *package for sustaining systems interoperability*.
4	Enterprise systems and networks need to support rapid evolution of technology and applications, and enable automatic or on-demand reprocessing, recompiling or fixing of components or processes.	Yes	Q1.2	Full	Closely related to challenge 2 and 3.

	Background Challenge	Coverage of PhD Research			Justification
		Addressed	Research Question	Alignment **(Full, Partial, Superficial)**	
5	EI needs a scientific foundation that assures repeatability of processes with similar results for similar problems and enables the evolution of ICT without prejudice to existing systems.	Yes	RQ	Superficial	Contributions to the EI science base are provided based on findings.
6	Enterprises need to be able to achieve interoperability with its partnering organisations, independently of the business, geographical and cultural requirements.	Yes	Q1.2	Superficial	Related to challenge 2 but more focused on semantic issues. Systems must be equipped with new tools/methods to support sustainability (establishment and re-establishment of EI): *package for interoperability establishment* and *package for sustaining systems interoperability*.
7	Public and private organisations need to be motivated to continue reinforcing the funding for interoperability research and technology development.	No	-	-	No direct contributions were envisaged in the research questions. However, the challenge is indirectly addressed through the dissemination actions.
8	EI needs a "white box" view of collaboration networks to support the analysis of model morphisms and enable the measurement and evaluation of the network efficiency	Yes	Q1.1 Q1.3	Full	Research focus is fully aligned, to include monitoring capabilities to know when harmonization breaking occurs: *package for sustaining systems interoperability*. Besides, an interoperability efficiency maturity model is proposed, providing a form of evaluating network efficiency, and used to support the simulation of the transient periods: *package for minimizing positive feedback loops*.
8	Enterprises should be advised and informed on how to recognise ICT systems that might be inefficient.	Yes	-	Partial	No direct contributions were envisaged in the research questions. However, the efficiency maturity model proposed enables to address the challenge partially.
10	Interoperability related issues, which are damaging a more effective outcome of the collaboration network, should be identified and mitigated enabling companies and the networks they belong to, to evolve according to different scales of interoperability maturity.	Yes	RQ	Full	High level challenge, closely related to the grand-challenge and the *Self-Sustainable Interoperability* developments.

4.3. MDA-based Package for Interoperability Establishment

In order to materialise the high level abstraction structure of the modelling language independent framework for interoperability establishment (depicted on Figure 4.10), a more complete representation of the proposal is depicted on Figure 4.13. It is based on the four levels of the model-driven architecture, relating meta-models, information models and data as described in section 2.3.1 "Model-Driven Architecture".

The left and right-hand sides of Figure 4.13 represent two different organisation's information systems with different internal legacy models, where information is presented following the model-language-meta-model hierarchy introduced previously in section 2.2.1 "Models and Meta-Models". The core of the architecture is focused on the middle part of the figure, and maps directly to the modelling language independent framework for interoperability establishment, enabling two complementary layers, i.e. the modelling language harmonization layer and the inter-enterprise harmonization layer:

- The first, with the pink background (boundaries shared with the enterprises), is focused on the definition of mapping morphisms at the meta-model level, i.e. the modelling language used in each information model. It is therefore the layer realizing the transformation of models from one language to the other, which in our case, is used as an intermediate step for interoperability establishment. Enterprise system models, standards or even reference ontologies are transformed to their abstract interfaces (and vice-versa) using metadata descriptions (the Language Independent Meta-Model - LIMM, presented in section 4.3.3) similar to the suggested in ISO/IEC 11179 Metadata Registries (MDR) (ISO/IEC, 2004).
- The last, with the green background (center), works sequentially after the first and is responsible for the model and semantics harmonization, defining mapping morphisms among the different abstract model interfaces (LIMs). The process includes storing this knowledge in a communication mediator knowledge base (detailed in section 4.4) replicated by the involved organisations, which serving as a standard during the mapping establishment will support the package for sustaining systems interoperability.

The architecture makes use of MDA's horizontal transformations (section 2.3.1.1) to support the harmonization of modelling languages, models and data levels, within a platform independent context.

4.3.1. Model Morphisms

Model morphisms are used across the multiple harmonization layers and throughout the MDA levels:

- Level 2 – language mapping;
- Level 1 – models and ontologies mapping, as well as the model transformation morphims;
- Level 0 – data transformation morphisms.

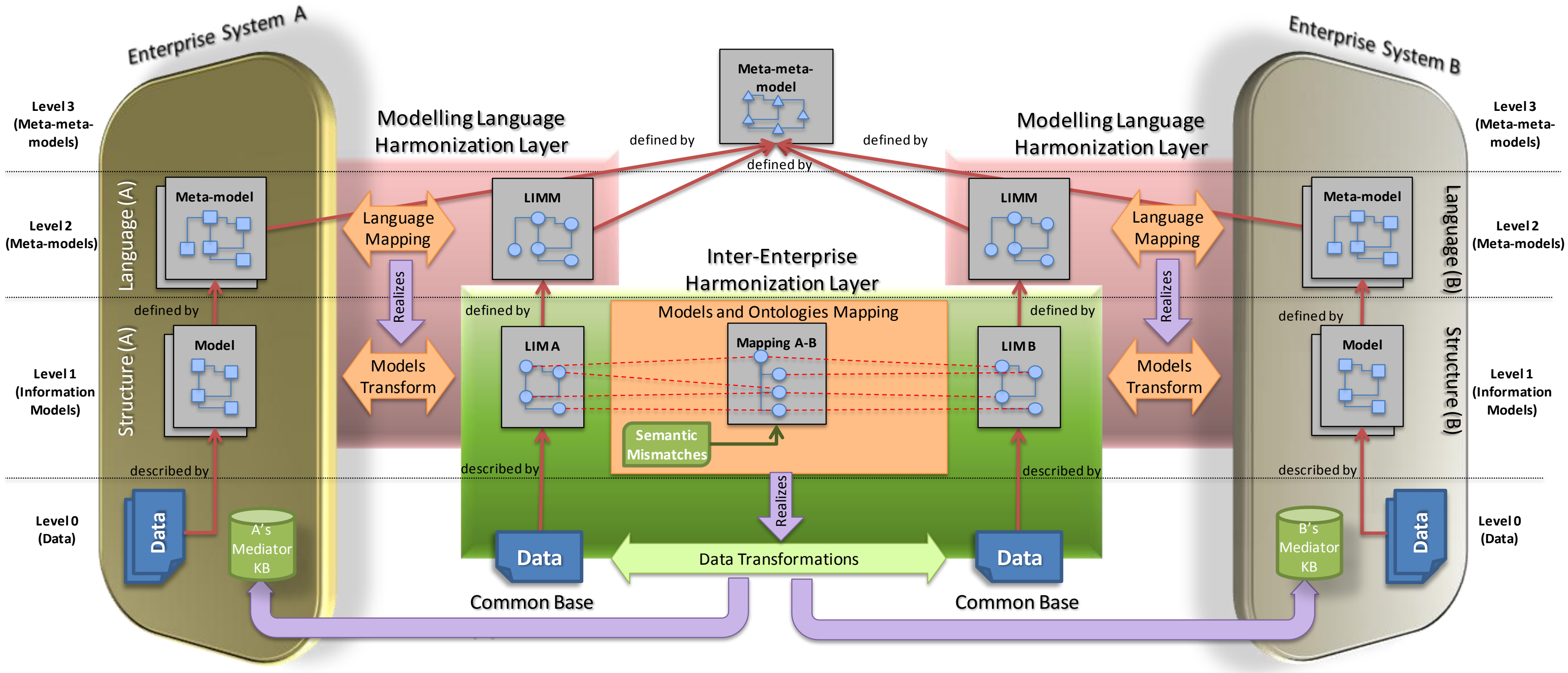


Figure 4.13: MDA-based Package for Interoperability Establishment

The MoMo's associated with the mappings are model non-altering "$\theta(A, B)$", which are described by mapping tables for each modelling language linked to the LIMM. These mappings are then implemented using an executable language, realizing the model altering morphisms (transformations "$\tau: A \times \theta \rightarrow B$") on the respective inferior level.

Since there are not so many modelling languages available, level 2 mappings are expected to be pre-defined and transformation scripts relatively static as changes in modelling languages specification is not common. They can be updated, but the mechanism for doing so is not envisaged to be as dynamic as the model and ontology mappings from the intra-enterprise harmonization layer. In that layer, for each information model there must be at least one transformation, and since models can be directly affected by the adaptive organisation's lifecycle, any mappings and transformations regarding particular models may have to be changed as models evolve, disappear or are added to the enterprise system. Therefore, unlike level 2, level 1 mappings need to be supported by a dynamic implementation of the architecture, enabling users to view existing mappings and change them to respond to new market requirements without compromising the network sustainability.

4.3.2. Modelling Language Harmonization Layer

As specified, this architecture layer is responsible for translating information models (see Figure 4.14). Mappings here defined are accomplished by establishing a correspondence, at the meta-model level (Level 2 of the MDA), between any specific language constructs and the language independent metadata, enabling bidirectional transformations at any enterprise information model (Level 1).

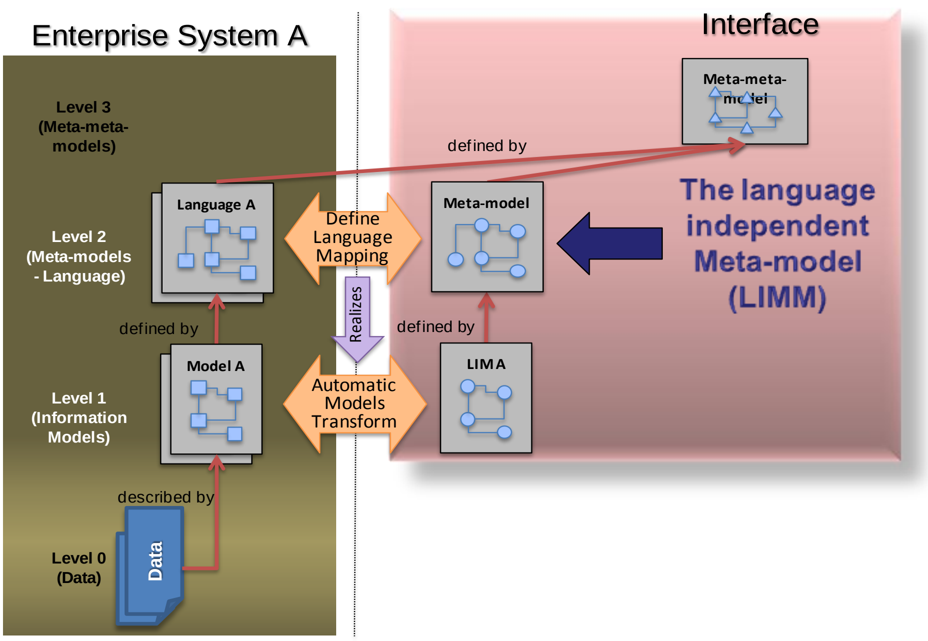


Figure 4.14: Detail of the Architecture Layer for Model and Language Independency

By being able to transform any given input back and forth to the LIM format (LIM meta-model - LIMM), the architecture accomplishes the objective of modelling language independency, helping

enterprises to further abstract from technology. To unleash it, executable rules can be applied to transform any N-1 level, according to the N^{th} level of the mapping. This way, one can represent multiple models according to LIMM (level 2) and, if there is a mapping defined between each input modelling language and the latter, multiple models from multiple languages can be represented by equal number language independent models (LIM).

The language mapping procedure is a manual process since meta-models must be analysed and mapped between them by experts, but the language transformations are always automatic and repeatable. Given that each language map is done only once independently of the number of times it is used or executed, it is an acceptable cost.

4.3.3. Language Independent Meta-Model (LIMM)

LIMM serves as an abstract interface on top of enterprises' information models. Through its usage, becomes possible to abstract the technology and implementation details associated with the different modelling languages, and thus, enlarge the scope of users involved in a traditional mapping definition activity. Having manager and domains experts involved in this process increases the quality of the mappings that will enable interoperable relationships. In comparison to most modelling languages, it is intended to enable as little loss of expressiveness as possible, but at the same time, be simple and generic to support multiple language mappings.

Also, LIMM resemblances with ISO/IEC (2004) standard are not by fortuity. This abstract interface was based on the standard's foundations and concepts in order to give support to mechanisms for enabling global data interchange, particularly across application areas. A bridge between major LIMM concepts and ISO/IEC 11179 can be made, e.g. the standard's "Entity", "Property" and "Representation" concepts correspond to LIMM's "Entity_Concept", "Property" and "Representation" constructs, respectively. The language independent meta-model proposed is described as an UML class diagram in Figure 4.15. It was designed with an UML tool, since UML class diagrams are a popular form to visualise the meta-model, enabling lossless integration with MOF XMI model which is compatible with most of the model management technologies for defining and executing transformations (e.g. ATL).

Many of the information modelling languages, e.g. EXPRESS (ISO TC184/SC4, 2004), UML class infrastructure (OMG, 2011b), OWL and XSD specification (W3C, 2009; W3C, 2001) have been analysed in detail and they were the focus of the attention to create this comprehensive meta-model and, as far the mappings defined for those languages demonstrate, LIMM is able to support them with little loss of expressiveness. In resemblance to what happens in the OWL language, LIMM is capable of representing both models and data levels of MDA (Level 1 and Level 0, respectively), enabling the combined transformation of both levels at the same time, or each independently if required. With this, not only the meta-model is prepared to deal with harmonization of modelling languages, but is also capable of representing instances of models, meaning that it can be used as an intermediate platform for data harmonization (represented by the "LIMM_Instances" package depicted in Figure 4.15 on the bottom).

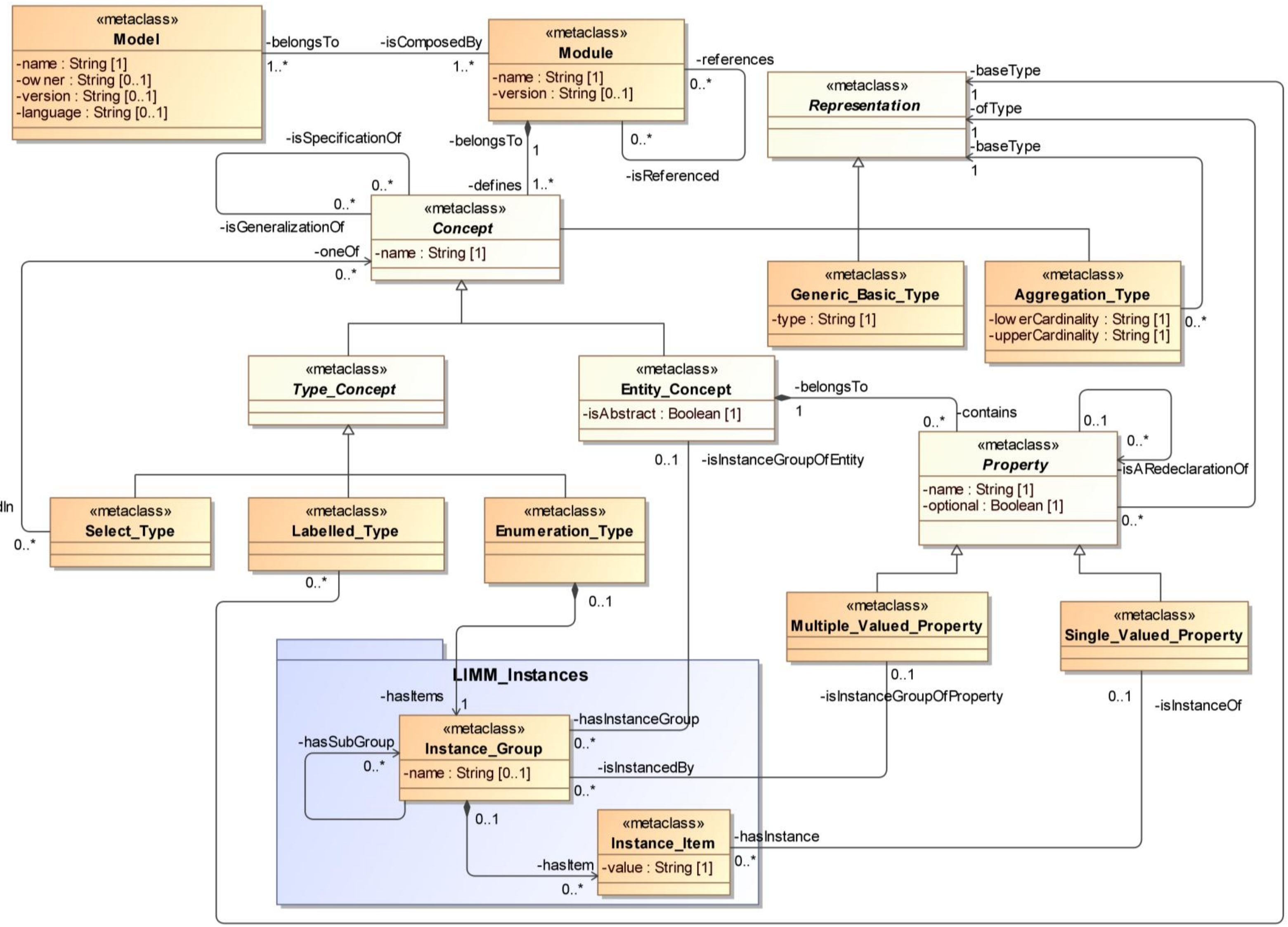


Figure 4.15: Language Independent Meta-model (LIMM)

Concerning modelling concepts, the meta-model considers the representation of entities, types, properties, basic types, aggregations, etc. Nevertheless, some languages (e.g. EXPRESS) enable explicit behavioural expressions (instantiation rules) and functions, which are not supported. However, they are considered non-fundamental for the envisaged mapping process which is mainly focused on the information model mapping at the Level 1 of the framework. A more detailed explanation of the composition of LIMM is presented, evidencing the use of each structure defined:

- *Model***:** identifies the "header" of the original model, in terms of owner, version and original modelling language. A *Model* can be composed by a multitude of *Modules*;
- *Module***:** each *Module* represents a fraction or the whole model, since original models can be distributed by a series of resources. The *Module* class identifies by name and version each part of the original model and is composed by *Concepts*;
- *Concept***:** is an abstract class to represent any kind of modelling structure defined at the root of the module (root elements of the original model representation). It either can be instantiated as complex entities (*Entity_Concept*) or as type declarations (*Type_Concept*);
- *Entity_Concept***:** this class represents an important structural part of the information model defining classes of objects. An *Entity* contains *Properties* and, when not abstract, can be used to represent palpable real life information, thus it can also be instantiated with level 0 data (through *Instance_Group*). Being a class that enables the specification of other classes, detailing a hierarchy of *Entities*, it allows to mark them as abstract if they are not meant to be directly instantiated;
- *Property***:** acts as complementary information about a given *Entity_Concept*, since a *Property* cannot exist without a belonging *Entity*. *Properties* have a given underlying associated type which can be any class inherited from the abstract *Representation* class. Moreover, depending of this type, i.e. depending whether it can assume a single value (e.g. an integer) or an aggregation of values (e.g. an array on integers), a *Property* is concretised as *Single_Valued_Property* or *Multiple_Valued_Property*. Redeclared *Properties* should be used only in case of need to redefine some other *Property* already specified within a supertype *Entity_Concept*. With this, a particular *Property* can be renamed and / or type redefined / refined. Finally, a *Property* can be linked with level 0 instances (through *Single_Valued_Property* or *Multiple_Valued_Property*) to represent palpable real life information.
- *Single_Valued_Property*: is a specific *Property* that can have at maximum a single instance value. It is connected to *Instance_Item*.
- *Multiple_Valued_Property*: is a specific *Property* that can capture multiple instance values at the same time, as is the case of arrays or matrices. It is connected to *Instance_Group*.
- *Instance_Item***:** is the valued instance of a real life concept;
- *Instance_Group***:** acting as an aggregator of *Instance_Items*, the *Instance_Group* class represents disjointedly either an instance of an *Entity_Concept*, an aggregation of the

possible values of an *Enumeration_Type* or even the values of a property whose underlying type is an *Aggregation_Type*;

- *Representation*: is the top abstract class which can go from *Generic_Basic_Type*, advanced *Types*, passing through *Entities* and *Aggregations*. This class represents the top level of abstraction of a single piece of information that can be modelled by the meta-model or that exists natively in modelling languages (e.g. "Strings");
- *Generic_Basic_Type*: this class represents a predefined language type. It can represent any basic type which the model demands (e.g. "Integer", "String", "Boolean", etc);
- *Aggregation_Type*: like the name explains, is a class to represent *Aggregations* (i.e. arrays, bags, vectors, etc), which can be limited by the "upperCardinality" and "lowerCardinality". This class has no direct information about possible contents besides the type it is associated with. This means that there is no information about possible order and duplicity of elements;
- *Type_Concept*: is an abstract class and represents the high level abstraction of selectors, renamed concepts and enumerations, such as *Select_Type*, *Labelled_Type* and *Enumeration_Type*. Together with the *Entity_Concept*, *Type_Concept* acts as a declaration of model structures, and helps the understanding of the difference which is inherent between them since it cannot have any inner elements such as *Properties*;
- *Select_Type*: is one of the advanced type structures and allows a given property to assume a multitude of different types, not limiting the instantiation to one particular type. This notion of selection only exists on the EXPRESS modelling language;
- *Labelled_Type*: allows to rename any previously defined concept or a native *Representation*;
- *Enumeration_Type*: the last advanced type, defines the use of *Enumerations*, which by definition are a set of well defined named values. These values are inherently constant and, in the LIMM, each *Enumeration* value is considered to be an *Instance_Item* attached to a specific *Instance_Group* representing the scope of all values this type can represent.

To better understand the inwards of how a normal information model looks like as a LIM, Figure 4.16 illustrates a simple example of a model to represent geometry, more specifically bi-dimensional triangles which are to be specified by a set of 3 points. Shortly and not exhaustively: the example *Model* is composed by a single *Module* which defines 4 *Entity_Concepts* and 1 *Enumeration_Type*; the entity to represent the triangle object is in fact a specification of the abstract concept named "GeometricShape" (which in a larger example can be extended by more *Entity_Concepts*, e.g. line, square, etc.), thus besides the *Multiple_Valued_Property* defined to capture the 3 vertices values, it also contains an indication of the type ("ofType") of the geometric shape given by the *Enumeration_Type* defined; and finally, the entity to represent a bi-dimensional point object defines a *Single_Valued_Property* for both the "x" and "y" coordinates, as well as another one to assign a label to a future instance of the point.

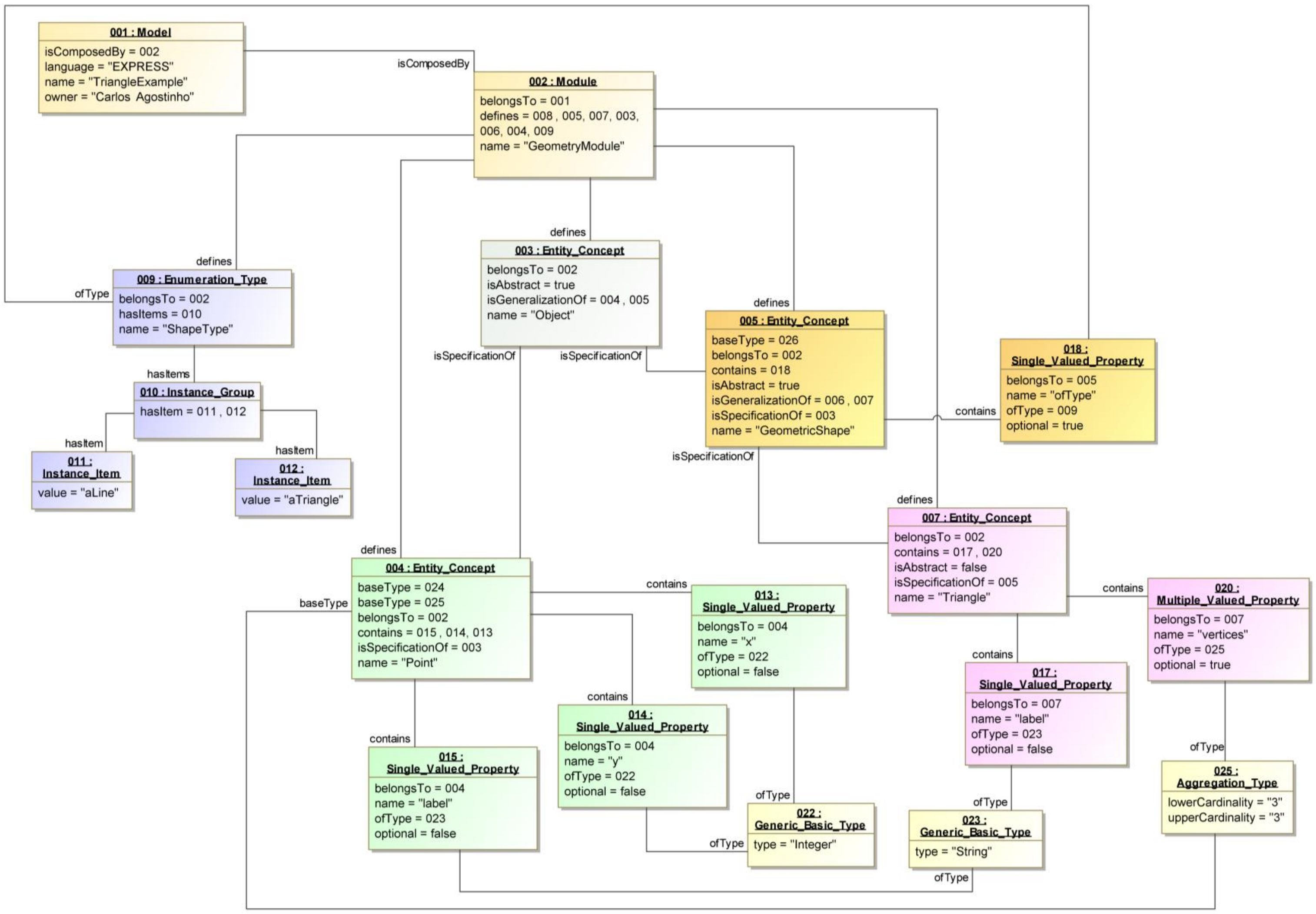


Figure 4.16: Example of a Simple Language Independent Model

4.3.4. Inter-Enterprise Harmonization Layer

Once all modelling languages from the different enterprises involved in the mapping definition are harmonised with the LIMM, and the models made available as LIMs, experts from each company should begin cooperating to define the actual P2P, P2Standard or P2Ontology mapping definition. As specified in the architecture of Figure 4.17, the inter-enterprise harmonization layer is responsible for this activity, following the same MDA horizontal transformation paradigm as before, and enabling automating transformations at the level N-1.

Besides the traditional connectivity, the semantic mismatches, found along the various model elements being mapped, are a very important topic regarding the experts' collaboration. As analysed in section 2.6.2 "Semantic Mismatches" many of the mapping morphisms will be imperfect due to a number of factors that can go from a simple encoding difference in equivalent properties to a granularity divergence. These can never be solved, but for change management and sustainability this is an important issue and the proposed architecture takes this in consideration, registering the complementarity between the model element correspondence and the semantic mismatch.

The mappings realised at this point do not suffer from the extra complexity of dealing with multi-modelling languages, focusing just on the business related constructs and easing the process of harmonising the semantic and structure level of models and ontologies. As a result of the entire process, generation of transformation morphisms for data from different enterprise nodes, or even to a reference format, is achieved, thus establishing interoperability as is the goal of the current development package.

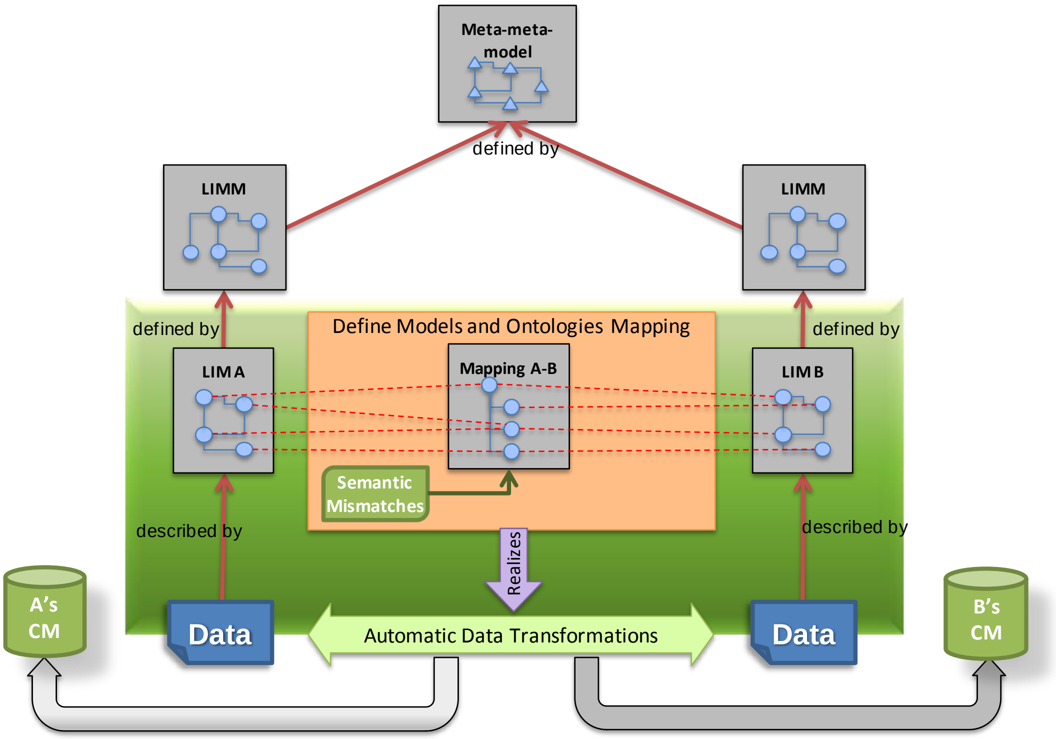


Figure 4.17: Detail of the Architecture Layer for Inter-Enterprise Harmonization

By the illustrated in the architecture of Figure 4.17, each pair of morphims (mappings and transformations) is stored on dedicated knowledge-bases designated as Communication Mediators (CM) (specified on section 4.4 below). The objective is that each organisation keeps its own CM to track relationships of their inner-elements with its business partner ones, thus maintaining a traceable record of relationships to support monitoring and intelligence activities of the package for sustaining systems interoperability, as well as “on-the-fly” composition of transformations.

MoMos defined at the modelling language harmonization layer could also be stored on each CM. However, those transformations are only used to enable the inter-enterprise mapping process, and do not have the same need for dynamicity nor monitoring. The union of the two transformations (for each direction of communication) unleashes the capability of, both automatic and transparently, communicate and collaborate with other organisations, with different modelling languages, models, semantics and ontologies. The complete automatic data exchange and translation can be accomplished between different model instances at the MDA level 0, thus completing the base for sustainable systems interoperability. Also, since all mappings of level 1 can be stored on a local knowledge base, it enables to gradually add more mappings with other partnering organisations and even to edit or delete past mappings. This provides the required adaptability of the framework to small collaboration networks, and being able to escalate to larger scenarios.

Although the MDA-based package for interoperability establishment proposes a complete solution to enable the model and language independency in multi-sized business networks, it is more focused in enabling the harmonization of the heterogeneous information models from the multiple organisations involved in the collaboration network. Semantics analysis through terminology mapping is also possible but, the further refinement of semantic interoperability is not in the scope of this dissertation.

Application scenarios illustrating the complete picture are included in section 5.1 “Application Scenarios and Test-Cases for a Self Sustainable Interoperability”.

4.4. Communication Mediator Knowledge Model

The knowledge base proposed to support the entire self-sustainable interoperability conceptual solution, designated as communication mediator (CM), is defined by means of an OWL ontology. It addresses traceability as the ability to interrelate the uniquely identifiable object versions in a way that can be processed by a human or a system. Also, morphisms are chronologically modelled through the package for interoperability establishment and updated by the package for sustaining systems interoperability, specifying relationships between information models and model elements defined in a system A (*relating*) and objects defined in a system B (*related*). The execution code for the morphims transformation is envisaged as well to be stored, thus enabling communities to build systems with reasoning capabilities able to understand each other’s representation format, without having to change their data and schema import or export.

The CM structure is presented in Figure 4.18 and described as follows. It has two main classes: *Object* and *Morphism:*

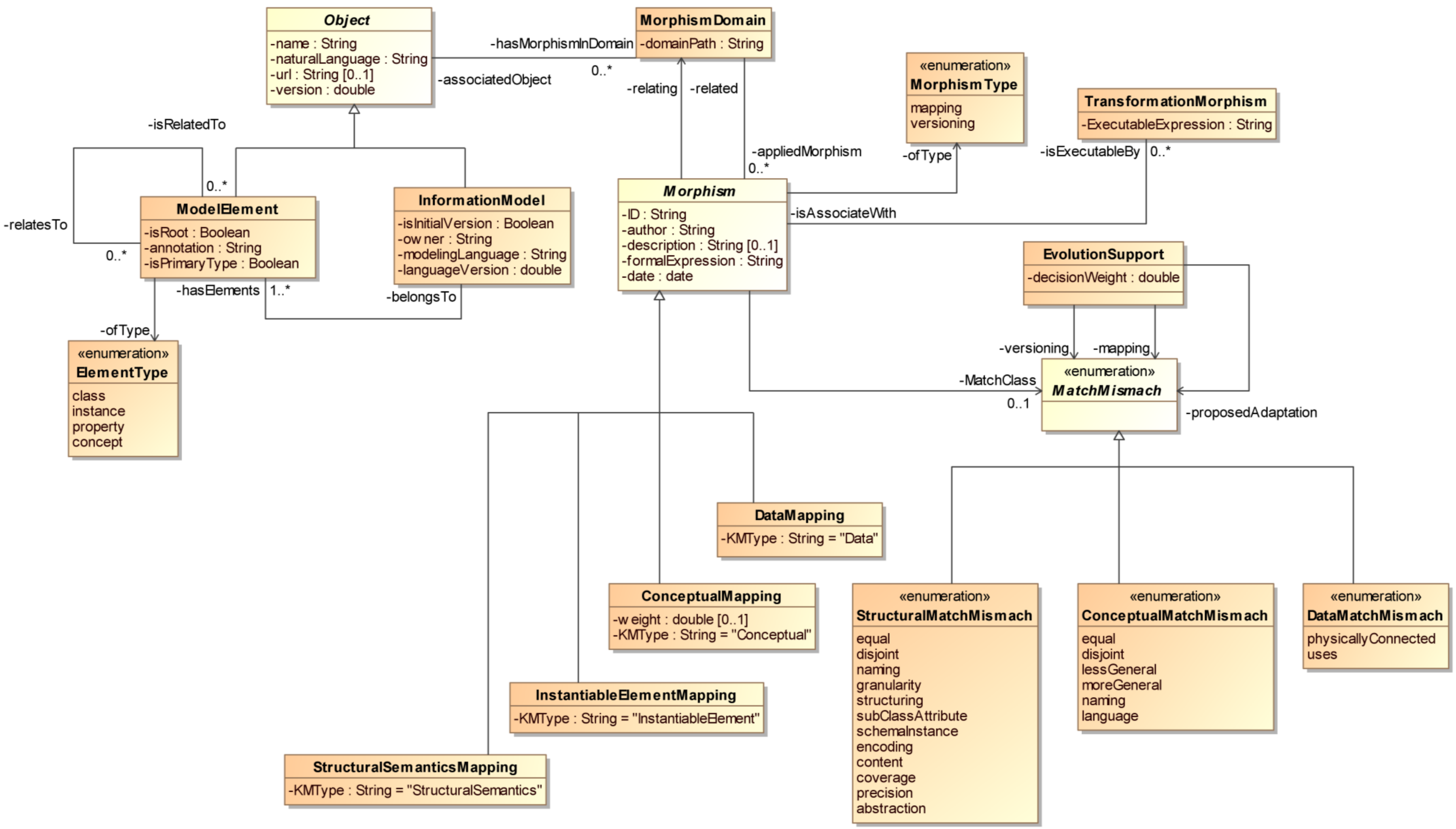


Figure 4.18: Structure of Communication Mediator

- *Object*: is an abstract class to represent any information modelling object, including the *InformationModel* or the *ModelElements* that compose it. The *Object* is identified by its name and version;
- *InformationModel*: specifies an *Object*, and as the name suggests, represents a system's information model or ontology, which has a number of *ModelElements* and is originally written in a specific modelling language;
- *ModelElement*: symbolizes any concept, class, property, or instance *Objects* used in a system's *InformationModel*. It is not meant to replicate the model structure with detail but it enables to indicate the *Object's* internal relationships;
- *Morphism*: represents the relationship (including the semantic mismatch. *MatchMismatch*) between two different *Objects*, the related and the relating. It is uniquely identified and chronologically tagged, being the author also specified. The specific structure of the *Morphism* follows the knowledge enriched tuple, explained in the next subsection 4.4.1. Nonetheless, *Morphisms* represented in this class can be of type "mapping" or "versioning" and can be executed by *TransformationMorphims*;
- *TransformationMorphim*: as studied in the MoMo literature, each mapping morphism enables a corresponding transformation, specified in this case by an executable expression that can be used by organisations' to automatically transform and exchange data with their business partners;
- *StructuralSemanticsMapping*, *InstantiableElementMapping*, *DataMapping*, and *ConceptualMapping* are specialisations of *Morphism* following the indications of the knowledge enriched tuple, explained in the next subsection 4.4.1;
- *MorphimDomain*: is the class used to link the *Morphisms* to the *Objects* using a domain path to identify the exact location of an object with an *InformationModel*.

A special case is the *EvolutionSupport* class, which is part of the mediator not to represent the knowledge of a single mapping as the previous ones, but to assist in the response to the harmonization breaking as explained in the MAS package for sustaining systems interoperability.

4.4.1. Knowledge Enriched Tuple for Mappings Representation (MapT)

Observing all previously explained MoMo formalisation technologies and methodologies for managing morphisms (see section 2.4.1 "Suitable MoMo Formalisation Techniques"), it is considered that there is not a single perfect solution to capture all the desired morphisms knowledge at once. Some methods are ideal for structural issues (e.g. model management), others for semantics providing good human traceability (e.g. semantic mismatches), while others are more formal and mathematical based. Therefore, to define mappings in a traceable and sustainable manner it is proposed the usage of an unidirectional 5-tuple mapping expression (equation 16), reusing some of the concepts introduced, that formalises the morphism between two model elements (*a* and *b*) and is enriched with semantic information that enables fast human readability, where $\forall A,B \in M, \exists a \in A \text{ and } \exists b \in B\text{: } if\ M\ is\ an\ LDMGraph\ then\ a \in V(A) and\ b \in V(B)$.

$$Mapping\ Tuple\ (MapT): < ID, MElems, KMType, MatchClass, Exp > \quad (17)$$

- "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 (entities, properties, instances, etc.). If the ID specifies a mapping at the n-th depth level of the graph, a should be at the same level;
- "KMType" stands for Knowledge Mapping Type (Figure 4.19), and can be classified as: "Conceptual" if mapping concepts and terms (very used for ontologies building, e.g. MENTOR (Sarraipa et al., 2010)), "StructuralSemantics" if mapping complex model elements, "InstantiableElement" if the mapping is specifying properties whose base type is simple (e.g. String), and "Data" for linking instances and allow other stakeholders to become aware they are tightly connected.

  $KMType = \{Conceptual, StructuralSemantics, InstantiableElement, Data\}$;

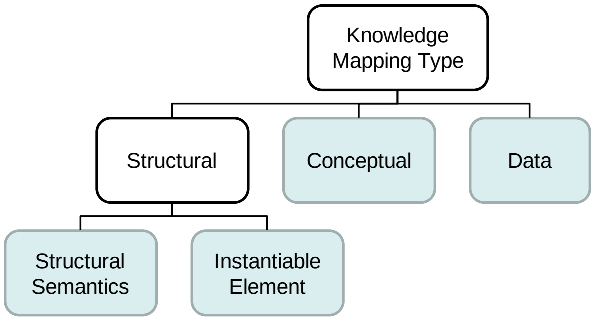


Figure 4.19: Knowledge Mapping Type Hierarchy

- "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 $MatchClass = Equal$; if $KMType = Conceptual$, the mapping is relating terms/concepts, and $MatchClass \in \{Equal, Naming, Coverage, MoreGeneral, LessGeneral, Disjoint\}$ depending on the coverage of the relationship; Otherwise, the mapping is structural or non-existent and recalling Table 2.3 of section 2.6.2 "Semantic Mismatches", $MatchClass \in \text{Table 2.3} \cup \{Equal, Disjoint\}$.

  Data mappings are a very specific case of mapping and do not fit the same mismatch categories. They can never be classified as "equal" as every object in the universe is unique, thus $MatchClass \in \{physicallyConnected, uses\}$, only identifying a high level relationship classes;

- "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 information systems, the tuple's information stored within the CM enables automatic data transformations and exchange between two organisations working with/on different information models, thus achieving an interoperable state among them and supporting the recovery from any harmonization breaking situation.

4.5. MAS Package for Sustaining Systems Interoperability

Today, most organisations are worried about how to become interoperable with their business partners. However, as important as that, is to remain interoperable along the adaptive organisation's lifecycle. Therefore, a system to monitor and look for unexpected changes in enterprise information systems is needed and envisaged in the scope of the framework for self-sustaining interoperability introduced previously on Figure 4.9 . It considers that the occurrence of harmonization breaking, means that network interoperability status has been compromised and needs to recover as soon as possible, consuming the least possible resources (both human, temporal, material or financial). In fact, every organisation part of a collaborative network should have the monitoring, intelligence and decision support skills envisaged in the framework. However, since resources are already scarce, these complex functionalities should become nearly autonomous to actually be accepted and implemented. For this reason, and also due to a wide applicability of complexity (see section 3.5.1 "Agent Based Modelling (ABM)"), the use of multi-agent systems (MAS) for the specification of the package for sustaining systems interoperability seems appropriate and fit for the optimisation of man-made complex systems (see section 3.6). Moreover, Wooldridge (2009), claims that decentralised multi-agents are considered to be an added value in the monitoring services implementation, assuring organisations' independence and interoperability among different MAS.

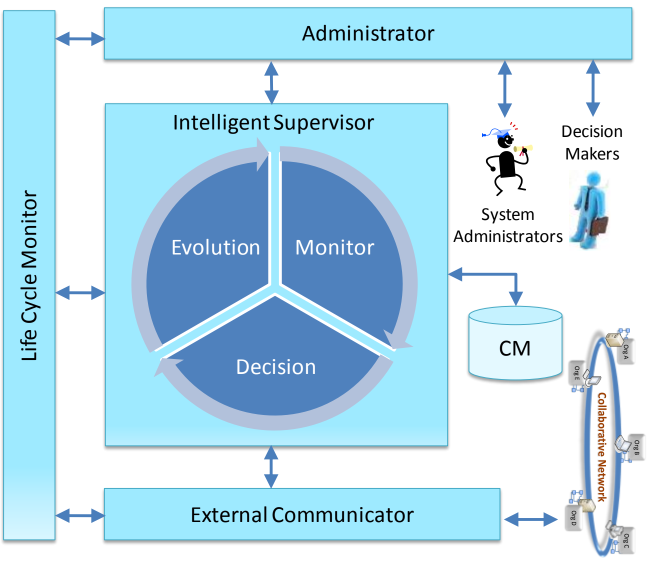


Figure 4.20: MIRAI Architecture

In order to materialise the high level abstraction structure of the framework of Figure 4.9, an architecture for monitoring morphisms in support of sustainable interoperability of enterprise systems (MIRAI) is presented on Figure 4.20 and Figure 4.21, below. It is based on four building blocks that use the previously defined communication mediator knowledge base:

- The Intelligent Supervisor is the most important of MIRAI, handling the detection of harmonization breaking as soon as CM changes. This block is responsible for managing a number of agents, searching for problems, finding model differences, calculating new entity mappings (and property sub-mappings) using the 5-tupple mapping expression (MapT), learning, evolving and updating the CM. Indeed, this block supports all phases of the adaptive loop and as such has two possible entry points (internal or external change), i.e. triggers;
- The Administrator is the interface between the MIRAI and the human user. When a new morphism is calculated, it must be proposed and approved by the human decision-maker. Weighting for this decision can be the transient simulation and analysis as addressed in the package for minimizing positive feedback loops (see section 4.6);

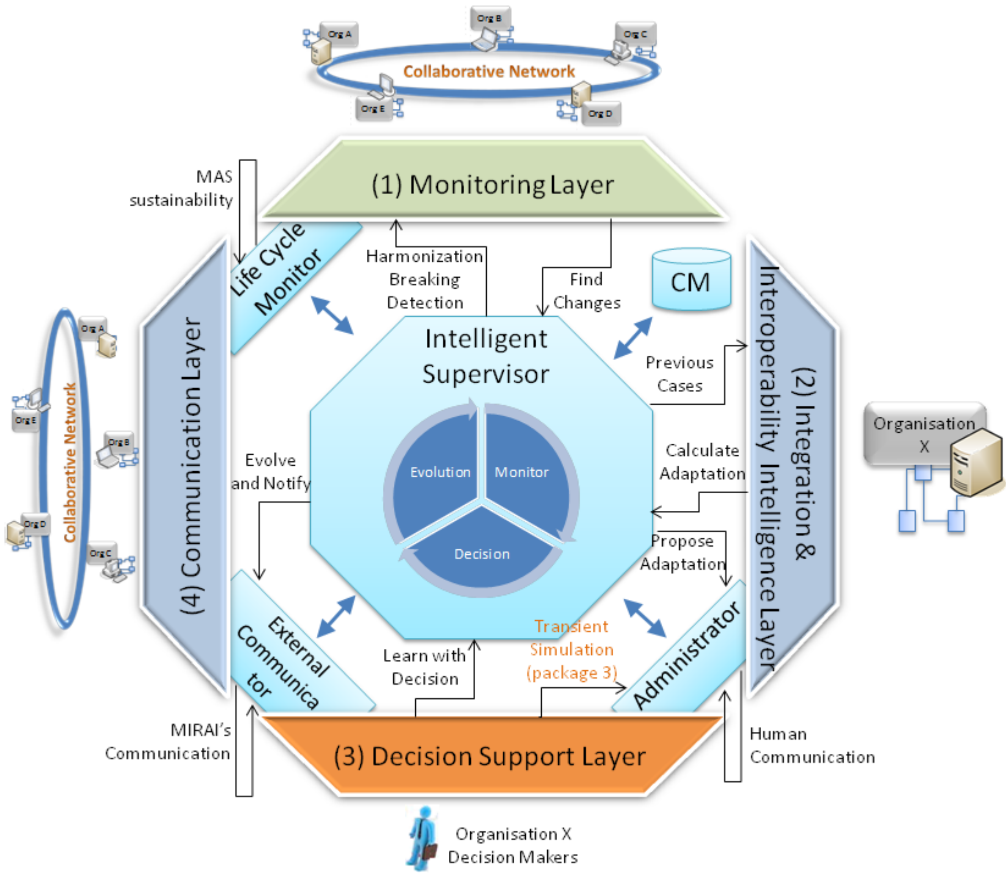


Figure 4.21: MIRAI within CAS-SIF framework

- The External Communicator, which is responsible for the communication between different MIRAI's of a collaborative network. In fact, when an evolution is stored in the organisation's CM MIRAI triggers a warning throughout the network, informing the related business partners that a change has occurred which could impact their relationships;

- The Lifecycle Monitor, which is a maintenance block to check if any agents died, and is responsible to resurrect them if required.

Following these descriptions, decentralised multi-agents are attached to the different blocks, composing MIRAI with a six agents MAS:

- *Agent Monitor Mediator* and *Agent MoMo*, part of the "Intelligent Supervisor";
- *Agent User*, part of the "Administrator";
- *Agent Persistor* and *Agent Persistor Police*, part of the "Lifecycle Monitor";
- *Agent Communicator*, part of the "External Communicator".

4.5.1. Intelligent Supervisor Block

This block is core to the entire MIRAI activity, being responsible for the detection of harmonization breaks, externally being informed by partner MIRAI's, or internally, through regular scans of the organisation's CM and searching for changes in the form morphism versioning's. After studying and determining what was the cause for the alert, the "Intelligent Supervisor" analyses whether it disrupts interoperability and, if it does, can choose among the various types of mappings to be proposed to re-establish regular communications, namely *StructuralSemantics*, *InstantiableElement*, or Conceptual (as in MapT). Data versioning's should be detected as any other, but the role of the intelligent supervisor in these situations is just informative since there is not enough metadata in the CM to facilitate a reaction to this type of changes.

As illustrated in the MIRAI architecture, besides monitoring and proposing proper adaptations, the intelligent supervisor block, also learns from the choices and decisions of the human decision makers, updating the CM (*EvolutionSupport* class) with their preferences. Therefore, the next time a similar situation occurs, MIRAI can provide solutions that along time have been preferred in similar problem situations. For reaching such objectives, MIRAI is directly associated to each CM. Moreover, within the collaborative network of enterprises there will be a kind of sub-network, i.e. the MIRAI network that enables to keep all CMs syncronyzed (only non confidencial data), maximizing the learning process as the whole distributed framework contributes with knowledge concerning user's selections.

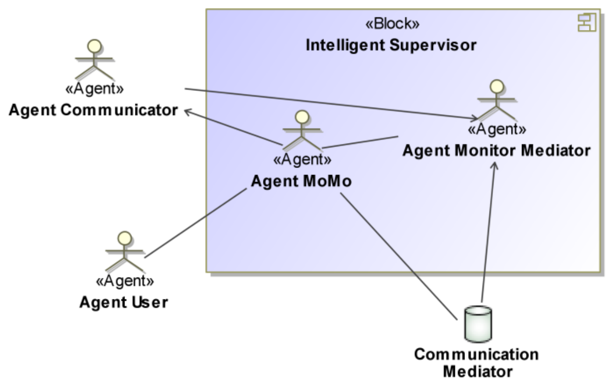


Figure 4.22: Intelligent Supervisor Agents Composition and Interaction

Figure 4.22 demonstrates how the agents composing the intelligent supervisor interact between and with the other MIRAI elements following the detailed activity specified in Figure 4.23, below. It is clear that in the envisaged architecture, regardless of their autonomy, the agents still need to cooperate between them to achieve their goals. The block's activities begin with the actions under the responsibility of the *Agent Monitor Mediator* which is also in permanent communication with the external communicator block to receive any external harmonization breaking alerts. The activities are then followed by the ones conducted by *Agent MoMo* that also has external interconnectivity, namely with the administrator block to inform and collect users' decisions. Both agents' use cases are defined in the following sub-sections.

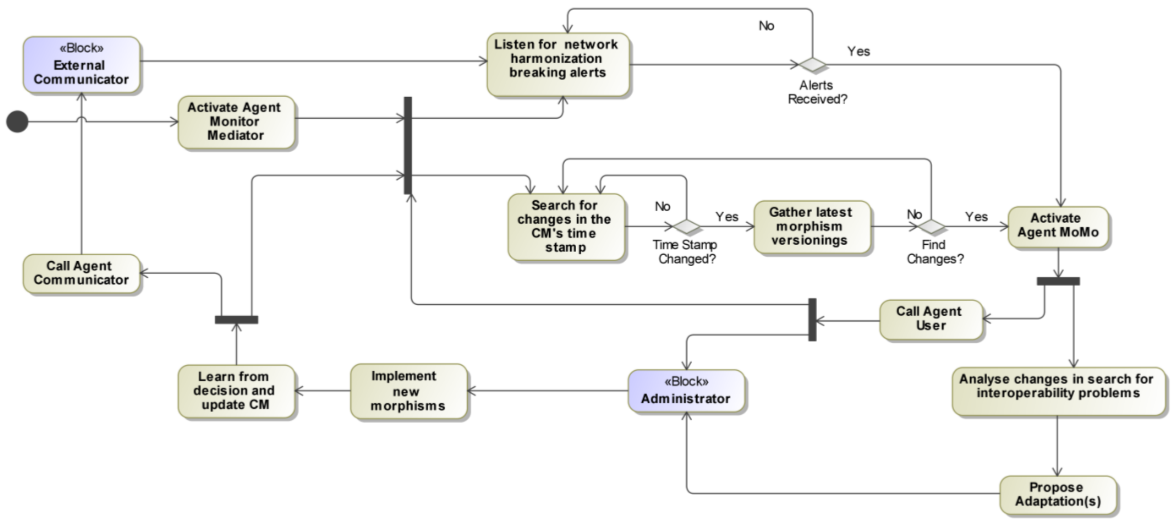


Figure 4.23: Intelligent Supervisor Activity Diagram

4.5.1.1. Agent Monitor Mediator

As the name indicates, this agent is responsible for the monitoring activities that put the whole MAS in action (see Figure 4.23 and Figure 4.24). It monitors the CM registry log, detecting changes in the knowledge base timestamps, and gathers the list of morphims registered after the previous one. Basically, it scans for versioning's which indicate an information model has been updated. Every time a new versioning is found, it activates *Agent MoMo* to respond to the evolution. Besides, this agent can receive external warnings (through the *Agent Communicator*) concerning morphism updates that will most likely require a *MoMo* response as well.

4.5.1.2. Agent MoMo

This agent acts when requested by the *Agent Monitor Mediator* and is the "maestro" responsible for the MIRAI intelligence (see Figure 4.25). As soon it starts to operate, the user is informed that something has happened and his/her attention will probably be required. However the agent's main objective is to check if the changes previously detected have an impact in the system's interoperable status, reacting with the appropriate activities in the positive case. Again, if a system's adaptation is required, the user must be contacted to decide upon the proposed response(s) (according to MapT), accepting it or suggesting others to be implemented.

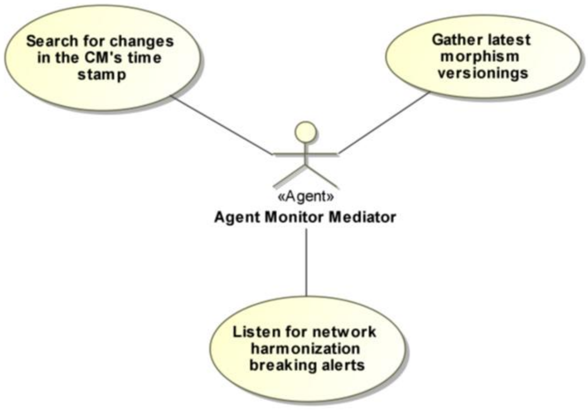


Figure 4.24: Agent Monitor Mediator Use Cases

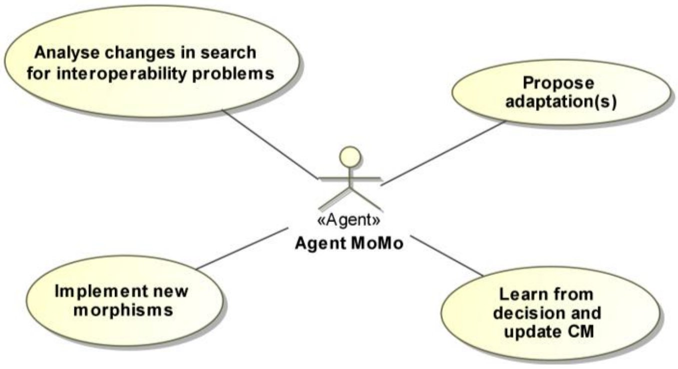


Figure 4.25: Agent MoMo Use Cases

As mentioned before, this agent updates the CM with the new morphisms and the user's preferences so that MIRAI can learn and propose better adaptations in the future.

4.5.2. Administrator Block

The administrator block is MIRAI front-end to the human user (see Figure 4.26). Through this block it becomes possible to keep track of all MIRAI activities, including the adaptation proposals. Nonetheless and most importantly it supports the user in all decisions in relation to the new mappings providing an informative bridge to the results of the transient evaluation package minimizing positive feedback loops.

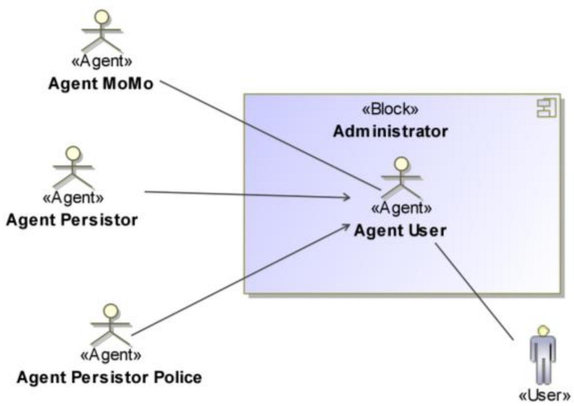


Figure 4.26: Administrator Agents Composition and Interaction

Through Figure 4.26 is becomes visible that the administrator block has a single agent, the *Agent User*, which maintains regular interaction with three other MIRAI agents following the detailed activity specified in Figure 4.27, below. Almost all of the messages exchanged are informative. For instance, it keeps listening to the agents from the lifecycle monitor block just to update the MIRAI log that can be displayed to the user at any time requested. Only in the case of mediation with the *Agent MoMo*, activities require more complex interaction, including potential interaction with the package for minimizing positive feedback loops, and support to decisions taken by the user.

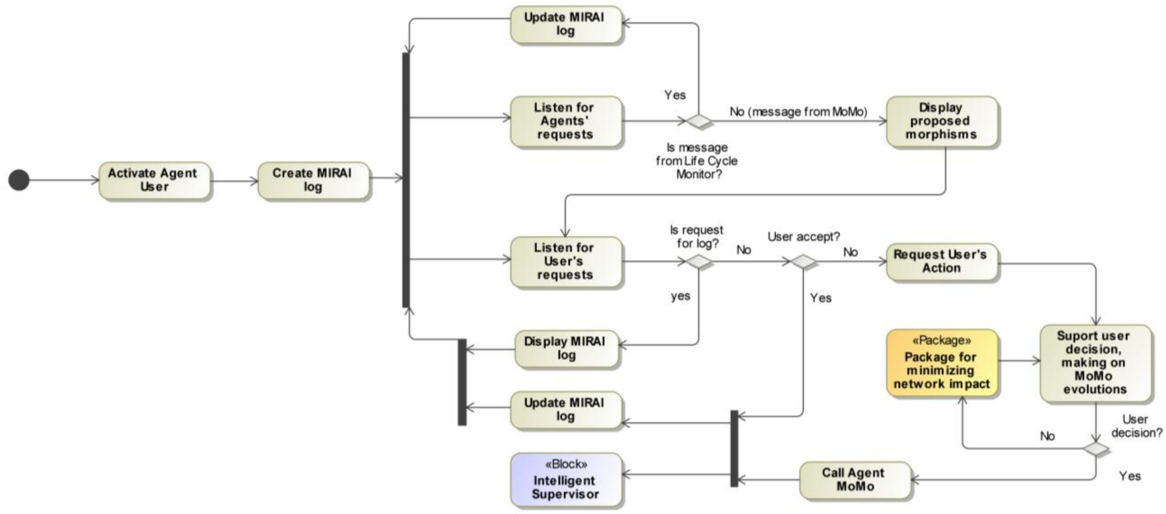


Figure 4.27: Administrator Activity Diagram

4.5.2.1. Agent User

Through this agent, the human is informed about the solutions proposed by the *Agent MoMo*, and is supported to decide whether if he/she accepts them, or proposes new ones using the MapT representation form. All use cases illustrating the actions it can perform are included in Figure 4.28.

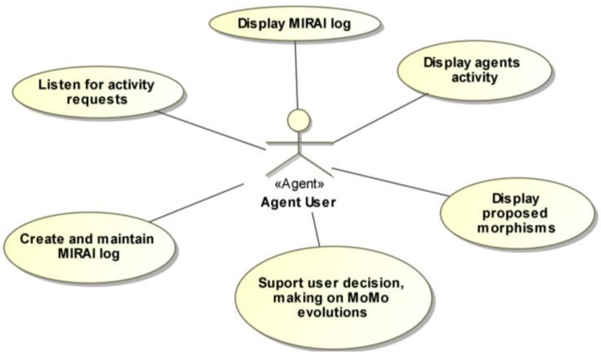


Figure 4.28: Agent User Use Cases

4.5.3. External Communicator Block

The external communicator block has a similar role to the administrator, but concerning the MIRAI's collaboration network, i.e., it is responsible for the communication between the systems, namely the agents *MoMo* and *Monitor Mediator* of the intelligent supervisor block, and the remaining MIRAI's (see Figure 4.29). It is composed of a single agent whose only purpose is to mediate messages.

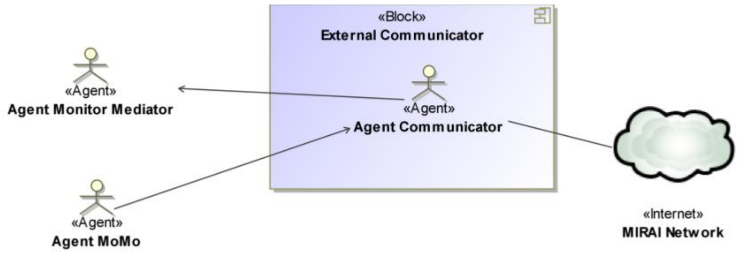


Figure 4.29: External Communicator Agents Composition and Interaction

This block listens to the network, through the *Agent Communicator* in search for news about changes occurred in another MIRAI that could cause harmonization breaking, and passes the information to the intelligent supervisor to check whether they have impact in the system. On the reverse scenario, where changes have occurred in the local system, it warns the interested neighbours about that situation, enabling a quicker reaction from them.

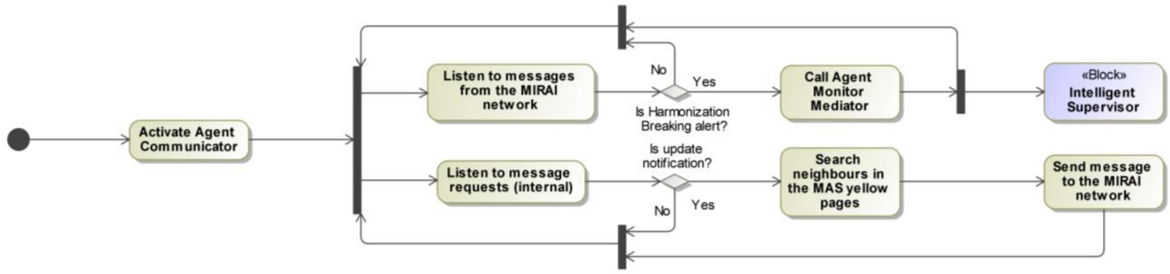


Figure 4.30: External Communicator Activity Diagram

4.5.3.1. Agent Communicator

Being the single agent of this block, agent communicator is responsible for the listening and messaging activities as described below. Also, it has access to the MAS “yellow pages”[45] to get the addresses of specific neighbours in order not to broadcast and overload the MIRAI network with messages that are not relevant (see Figure 4.31Figure 4.28).

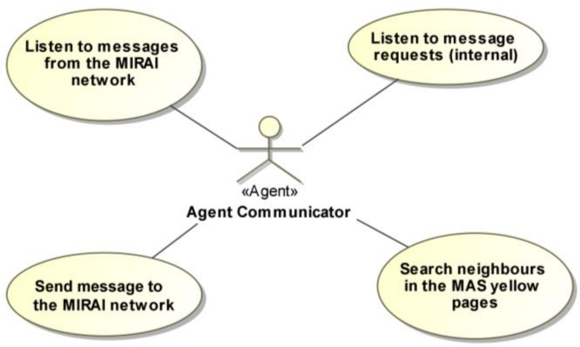


Figure 4.31: Agent Communicator Use Cases

[45] Functionality similar to the Web Services UDDI (Universal Description, Discovery, and Integration), i.e. a registry by which businesses can list themselves on the Internet and locate web service applications

4.5.4. Lifecycle Monitor Block

The final MIRAI block is responsible for the persistence of the agents, monitoring the physical network and recovering dead or missing agents due to infrastructure failures. This keeps the entire MIRAI network interoperable since without this block, if one of the agents died, communications would be lost preventing the system from acting as expected. For example, if *Agent MoMo* dies it would be impossible to have a diagnosis and a sustainable interoperability.

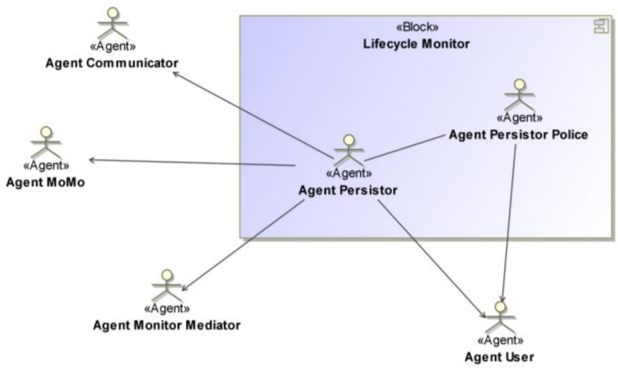


Figure 4.32: Lifecycle Monitor Agents Composition and Interaction

As illustrated in Figure 4.32, the lifecycle monitor block is composed of two agents, adding redundancy to this persisting system. One monitors the whole MIRAI (*Agent Persistor*) while the other (*Agent Persistor Police*) monitors the first one, ensuring proper reaction in case it dies. In fact, to launch MIRAI, the user only needs to launch *Agent Persistor Police* which will create *Agent Persistor* and, this one, the remaining agents. After that, and as depicted in Figure 4.33, the lifecycle monitor block enters a monitoring loop, "resurrecting" the required agents and ensuring a sustainable MIRAI. At the end of each loop, communication with the administrator block is established to register this activity in the MIRAI log.

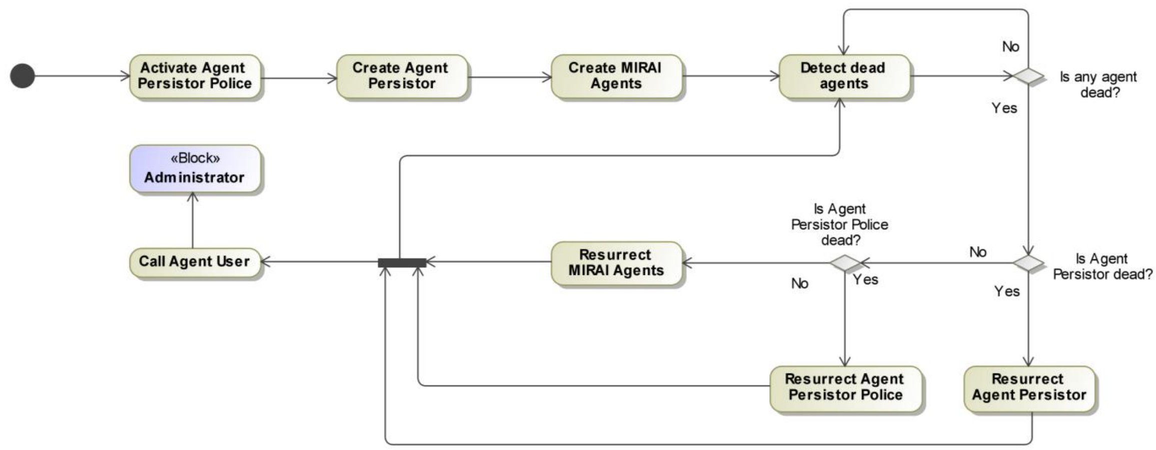


Figure 4.33: Lifecycle Monitor Activity Diagram

4.5.4.1. Agent Persistor

This agent controls the creation of any MIRAI agents, "resurrecting" them if required and maintaining the MAS in operation. It is important to note that the *Agent Persistor* also detects if the

Agent Persistor Police is dead taking appropriate reactions to ensure that MIRAI has the envisaged redundancy (see Figure 4.34).

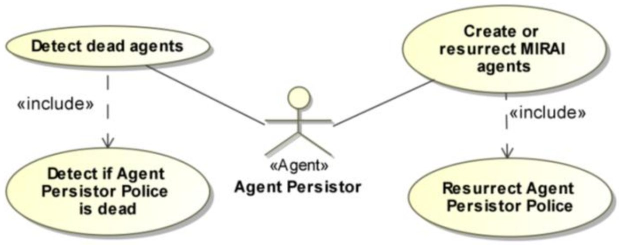


Figure 4.34: Agent Persistor Use Cases

4.5.4.2. Agent Persistor Police

This agent has a similar role to the one described above but only controls the *Agent Persistor* (see Figure 4.35).

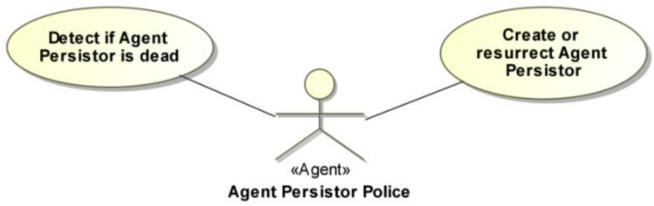


Figure 4.35: Agent Persistor Police Use Cases

4.6. Transient Evaluation Package for Minimizing Positive Feedback Loops

As explained before, in complex networks and systems, any type of disturbances may lead to costly and unexpected behaviour. MIRAI details a self-monitoring system that can detect these disturbances, analyse their implications at the organisation level morphisms, and even propose some adaptations to solve the immediate problem, sustaining network interoperability. However, these adaptations can also be seen at the network level as disturbances. As previously identified, intervention strategies on the network node can affect the network interoperability sustainability and, in the long run backfire, causing more damages to the organisation's businesses than the ones originally caused by the first problem.

Moreover, adaptations may sometimes demand more complex adjustments than just at the information system morphisms. For example, new business opportunities may arise requiring changes at the organisational level of the enterprise, or at the opposite end, some business relationships can be put in question if the negative impacts of the changes overtake the business profit. Due to these reasons CAS-SIF envisages a decision support layer that, among other activities detailed in the previous package, simulates the potential transitory caused by harmonization breaking responses, in business collaboration networks.

Transient is a term frequently used in other disciplines to represent the time interval between different stages of a system's activity. Nonetheless, here, transient does not concern solely with

time. The network re-adaptation needs to be studied from different points of view where time is an important one but not the unique, e.g. cost and benefit fluctuations, and the factors that influence them, are of major importance as well, since they are regulating the enterprise market sustainability and management strategies. Thus, being displayed by a nonlinear waveform function, sustainable interoperability transient analysis is time discretized and evaluated in term of cost-benefit ratio, defining the enterprise fitness landscape. Figure 4.36 proposes a high-level IDEF-0 (USA Secretary of Commerce, 1993) "black box" view on the transient simulation activity specifying inputs, outputs, control, and mechanisms:

- *Inputs*: The existing morphims that regulate enterprise electronic communications with the business partners and the proposed adaptations suggested by MIRAI compose the system's inputs. Along with the control variables, they feed the transient simulation system with the information required to calculate the outputs via the application of the indicated mechanisms.
- *Controls*: Are the variables that when adjusted can fine tune the outputs. For the specific case of transient simulation, it is considered that quality and business requirements, as well as the interoperability degree, influence directly the type of relationships an organisation desires to maintain with its business partners. Also, time constraints and the budget available for internal readjustments can influence the decisions one needs to make when considering the proposed adaptations. Most of the times, if it is possible to relax a bit these variables (e.g. losing a partnership or increasing the time for the next message exchange), transient periods and cost-benefit ratio can be optimised.

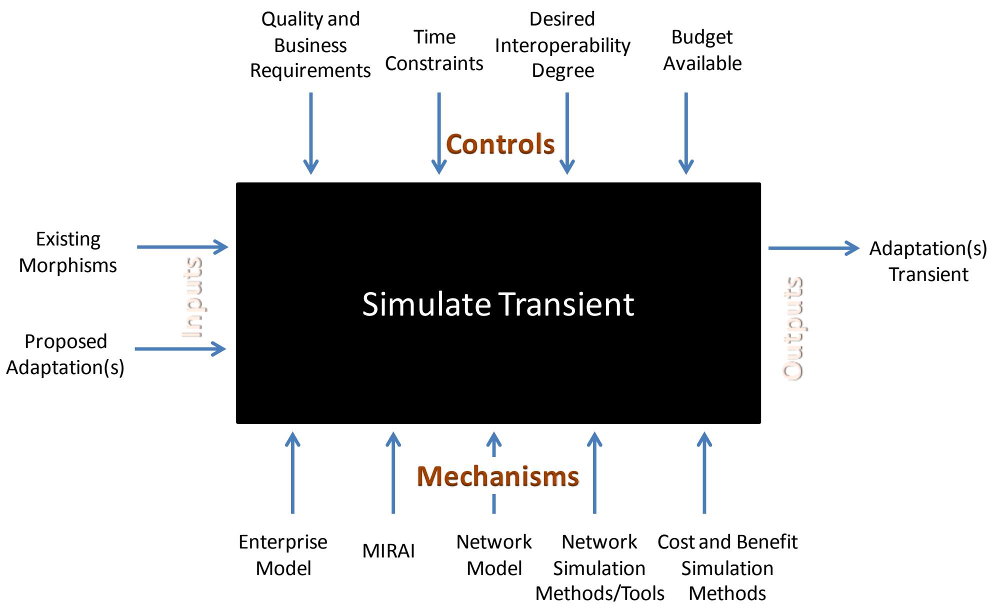


Figure 4.36: Adaptations' Transient Simulation Activity

- *Mechanisms*: Typically a mechanism is a person, facility or machine that executes the process. Since this is a simulation process, it requires models and simulation methods that can produce the required outputs based on the information provided by the inputs and the controls. To simulate the enterprise, one needs to have a well defined enterprise model

describing not only the internal information systems, but also organisational structures, business processes, etc. With this information the complexity of implementing system readjustments can be calculated. Similar situation can occur for the network, where agent technologies or neural networks (among others) can help predicting the impact of the adaptations on the partnering organisations, in such a way that cost-benefit calculations could be extrapolated for specific time intervals.

- *Outputs*: The system output is the transient estimation figure along time.

Running the transient simulation function a number of times for different adaptation sets will provide a knowledge base for decision support, which can guide the human user in the process of feeding optimal suggestions back into MIRAI and/or other enterprise systems and methods involved in the post harmonization breaking adaptation process.

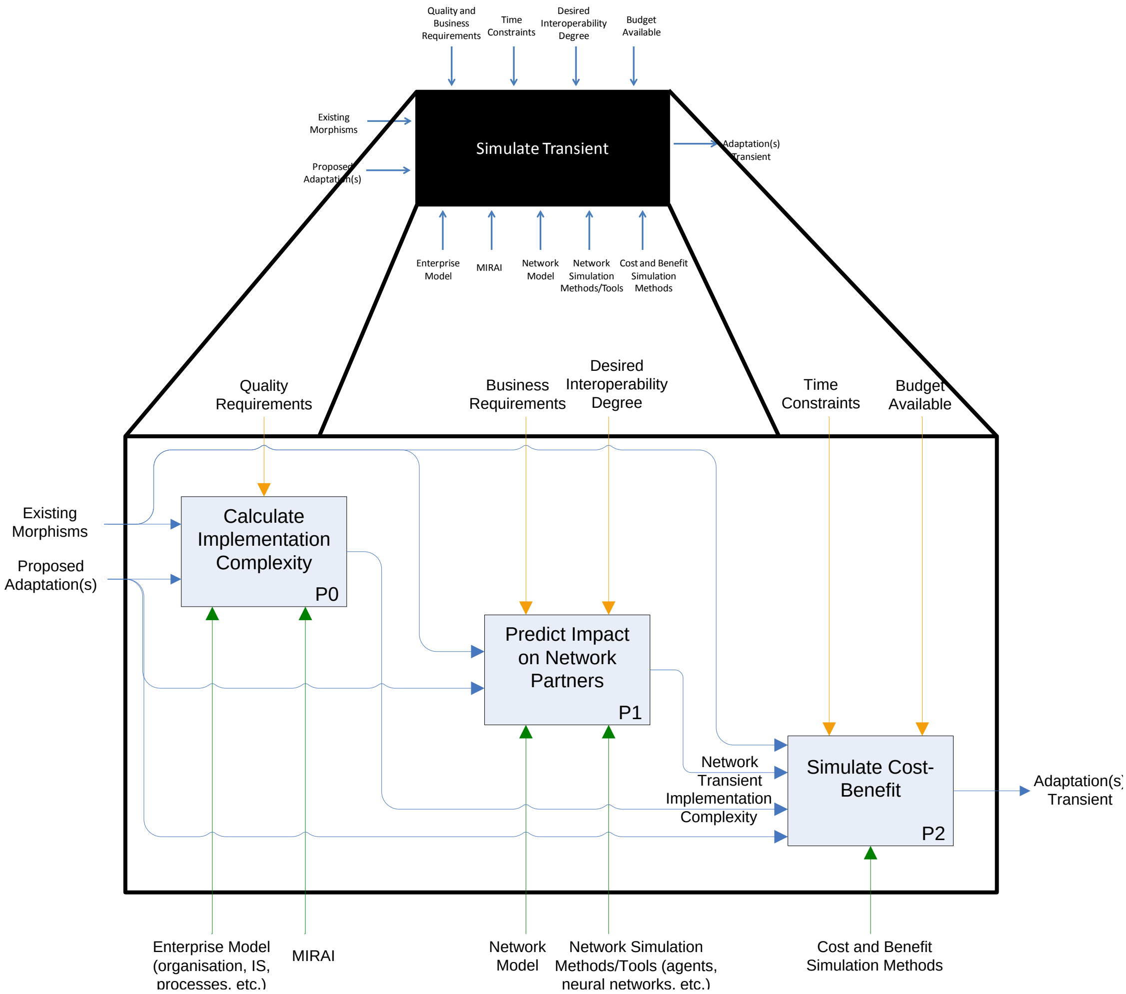


Figure 4.37: IDEF0 Detailed Specification of the Transient Simulation Activity

Figure 4.37, illustrates the activity's decomposition in three sub-processes considered fundamental towards the transient.

4.6.1. Implementation Complexity

As envisaged, sustainable interoperability transient analysis is not solely dependent on time, thus the first sub-activity (P0) considered is the calculation of the implementation complexity for each adaptation (used when estimating costs and resources). For that purpose, one could define a measure of complexity by looking at the enterprise as a white box and analyse the micro-level relationships among structures (MIRAI morphisms decomposition), concepts, people or processes that will need to be rearranged. Also quality requirements (e.g. the amount of testing required), can impact directly this complexity – more quality demands more testing and verification, implying more people involved.

4.6.2. Network Partners Impact

A complete complexity analysis cannot be constricted to the enterprise as a single system. In fact, one needs to consider also the transient periods that the proposed changes will cause in the collaboration networks the enterprise under analysis belongs to. Therefore, the second sub-activity (P1) targets the prediction of the adaptations impact on these network partners.

Based on the organisation's own business requirements and on the desired interoperability degree, this impact measure can be more or less relevant, since if direct business partners are affected, they could also need to readapt, increasing the network transient period before regular business interactions are re-established.

4.6.3. Cost-Benefit Simulation

Finally, using the results of the first and second sub-processes, and considering time and budget constraints, a cost-benefit simulation (P2) may take place to calculate the actual transient over time.

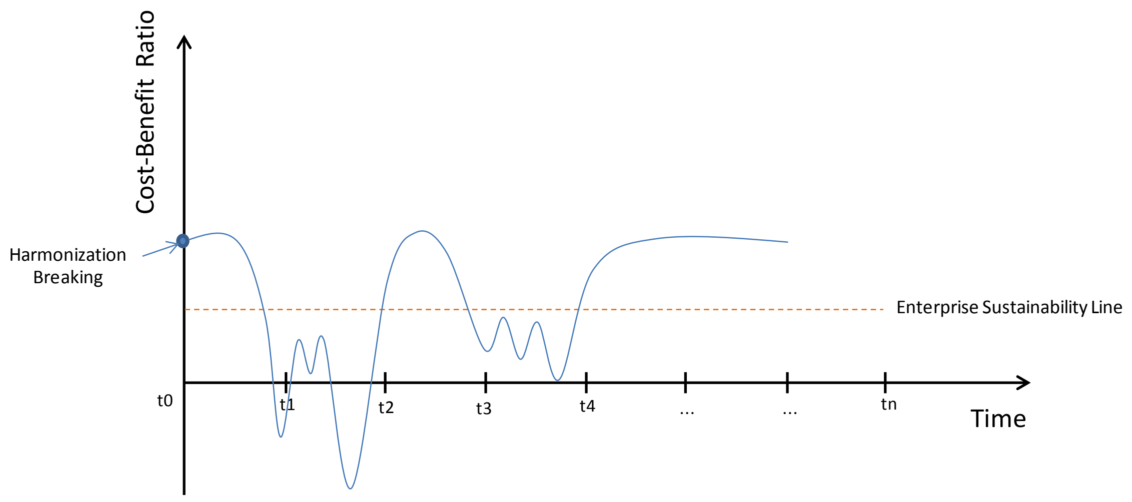


Figure 4.38: Example of Adaptations Transient in a Fitness Landscape form

Figure 4.38 exemplifies a possible result where it is clear that after harmonization breaking, the organisation experiences a sudden drop of the cost-benefit ratio, even going into negative values. Between t1 and t2, it recovers internally creating an apparent stability that is disrupted again by the

partner's reorganisation (between t3 and t4), thus lowering the cost–benefit ratio below the enterprise sustainability line. Only after t4, the system demonstrates stability. Apparently, this is an acceptable result since harmonization is achieved above the sustainability line, however, the human decision makers need to analyse the results carefully checking whether adaptations are adequate or if the organisation is spending too much time below the sustainability threshold and cannot accommodate the losses.

5. Proof-of-Concept Implementation

A proof-of-concept (POC) by definition is a realization of a certain method or idea(s) evidencing that a product or technology is viable and capable of solving a particular problem (Farlex, 2011)*. In engineering, a rough prototype of a new idea is often constructed as a POC to verify that it has the potential of being used. In these situations, it is usually small and may not be comprehensive of the whole idea. This way, this chapter begins by defining the applications scenarios and test-cases to which this thesis is applicable.*

As in the specification, the POC has been split in packages according to the framework for sustaining systems interoperability. They reply to the test-case specifications and serve as an input to the testing and practical validation of the hypothesis. The technology used and the implementation details are here described, as well as the steps required to replicate the POC.

5.1. Application Scenarios and Test-Cases for a Self Sustainable Interoperability

The objective of a scenario is to describe, step by step, how a user (or users) intend to exploit a system, essentially capturing the system behaviour from the user's point of view (CRESCENDO Partners, 2009b). Considering the motivations for this work and the fact that a self sustainable interoperability is needed not only for a single industrial sector or scenario, it is important to prove that the concept presented in chapter 4 "Self-Sustainable Interoperability" is valid throughout a number of business situations. For this reason it was decided to scope the POC scenarios within two well known business collaboration networks: Supply Chain (SC) and Collaborative Product Development (CPD).

The scenarios depicted in this chapter provide a high level vision of some of the possible use-cases for this thesis work on sustaining systems interoperability. As it would be impossible to

include all of them, a selection of the most pertinent scenarios is presented, as well as a number of test-cases (TC) highlighting some functionalities and results of the CAS-SIF technical packages.

5.1.1. Supply Chain in Manufacturing Networks

Supply chains are critical infrastructures for the production, distribution and consumption of goods and services (Nagurney, 2006). According to Ganeshan & Harrison (1995), based on the customer's order, they enable raw materials, supplies and components to be modified into finished products and then distributed to the consumers. This involves two major types of communication, supported by physical and information flows (W. J. White et al., 2004). The first usually involves moving goods "forward" from suppliers to consumers in each link of the chain, while the information flow involves exchanging product and financial data, namely electronic catalogues, orders and payments for materials, services, supplies and final products.

Figure 5.1 illustrates a very simple SC involving six stakeholders from the supplier to the consumer. Each of them can be instantiated by different organisations, thus the two flows offer multiple paths. It is considered that the consumer can either contact a physical retailer directly (Retailer ↔ Consumer) or an online e-marketplace to search for products (e-Marketplace ↔ Consumer) which, in turn, can be delivered at home using a distributor company (Distributor ⇨ Consumer) or at the retailer shop (Retailer ⇨ Consumer). In this SC, the distributor stakeholder takes a central role in the physical flow, however the direct physical path among manufacturer and retailer (Manufacturer ⇨ Retailer) suggests they can be the same company. This is an evidence that, as recognised by Wycisk et al. (2008), the understanding of supply chains as linear systems has evolved over time to complex systems.

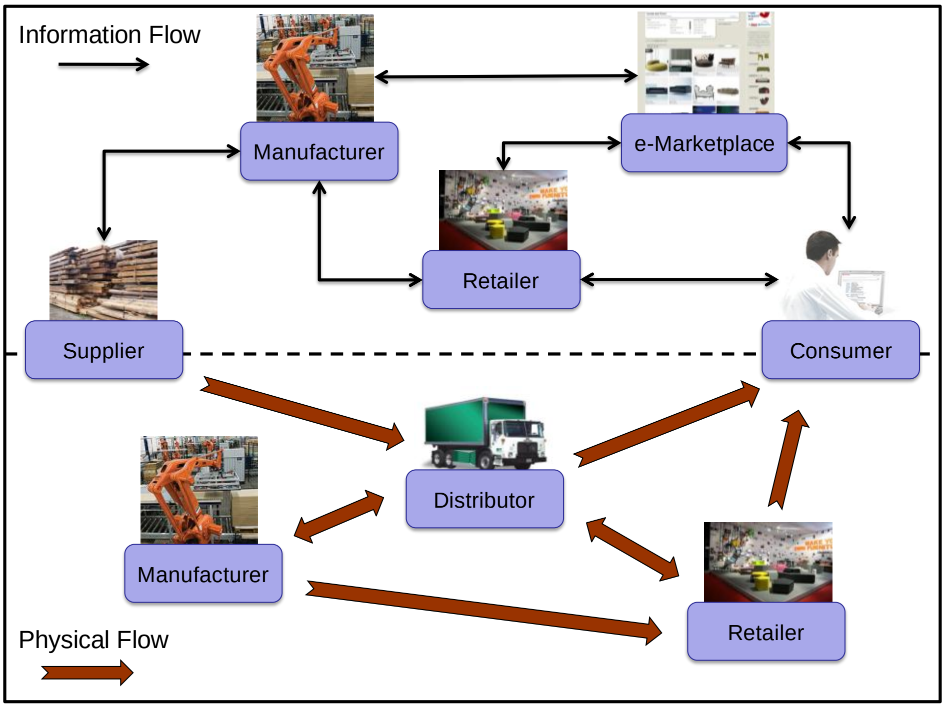


Figure 5.1: Supply Chain Flows (based on Jardim-Goncalves et al. (2007))

In today's globalized networked economy, SCs may span across continents and involve numerous players underpinned by multimodal transportation and telecommunication services. With the information being seamlessly exchanged between parties the consumer has better choices and a degree of power in customising its own particular product demands. Nonetheless, since these networks are characterized by non-centralized decision making, changes in the availability of supplies, prices, as well as disruptions to transportation or communication may cause effects that propagate throughout the entire network and decrease SC efficiency. Indeed, critical issues, such as stability, elasticity, adaptability and responsiveness to events in a global environment of increasing risk and uncertainty are of uttermost importance for effective supply chain management (Nagurney, 2006).

Therefore, the topic of sustainable interoperability within SC networks is applicable and interdisciplinary by nature, since it involves manufacturing, transportation and logistics, as well as retail and marketing. A number of application scenarios could be drawn, however the focus of this thesis concerns information systems interoperability, thus the POC illustrates how interoperability between retailers and e-marketplaces is facilitated and how they keep interoperable down the chain with manufacturers, and these with their suppliers. Two application scenarios integrate the POC and are described in the following sub-sections:

- The first scenario demonstrates how the framework for self-sustaining interoperability (presented in section 4.2 "Conceptual Solution to Enable Hypothesis"), supports the addition of a new node to a SC network;
- The second scenario focuses on a situation of harmonization breaking due to a change on a network node's information structure;

5.1.1.1. Scenario 1: Manufacturer "A" wants to enlarge network and publish product catalogues in another e-marketplace

The more interfacing nodes a manufacturing organisation has between itself and the final customer, the better chances it has of being successful and selling products. Naturally, in real life it is not as simple as that since a number of other factors can determine the success of an organisation, but for sake of objectivity, scenario 1 does not consider any of those.

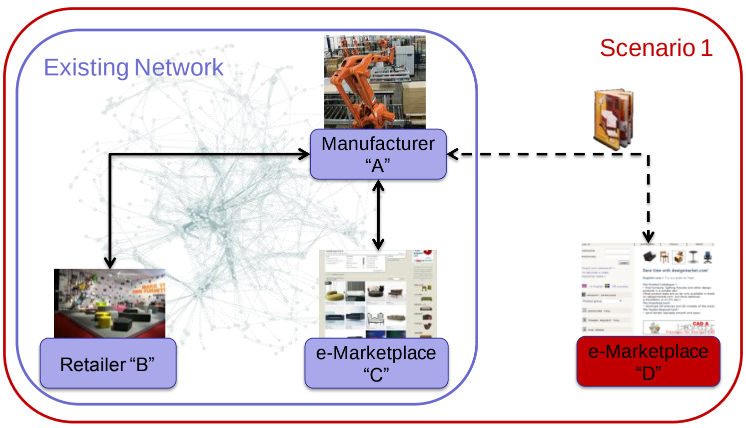


Figure 5.2: POC Application Scenario 1: Addition of a New Node to a SC Network

In this scenario, manufacturer “A” is part of a supply chain as the one of Figure 5.1 and has its product catalogue published in a retailer shop “B” and in an online e-marketplace “C”. Among them and according to the classification of section 4.1 “Interoperability Efficiency Pyramid Model (IPyM)” they have a classical case of unregulated interoperability complemented with the sustainability framework proposed. As illustrated on Figure 5.2, the goal of the scenario is to analyse weather the framework is capable of supporting an additional node in the manufacturer’s network, specifically a new e-marketplace “D” so that “A” can publish its product catalogues and reach different audience from the one targeted by “C”.

A generalisation of the envisaged interoperable network and the steps towards achieving it are depicted in Figure 5.3, with four enterprises (“A”, “B”, “C” and “D”). These organisations neither share the same modelling languages nor information models in their internal IS, thus the complete workflow (as envisaged in the packaged for interoperability establishment) from each enterprise joining the collaborative network, towards getting to understand each other and finally to communicate transparently and automatically is explained in the next three steps:

1. The first step to reach an interoperable state involves model translation to a common language of understanding, thus achieving the envisaged language independency. In the solution, this step consists in doing a transformation of the enterprise “D” (e-marketplace) model to an instance of the language independent meta-model (LIMM) of section 4.3.3. Thus, mappings are defined at the meta-modelling level 2 between the meta-model of “D” (its modelling language) and LIMM. They must be implemented via two morphisms: the *L2Limm* that has the direction of transforming D’s model to its representation in the LIMM and, on the other hand, the *Limm2L* which has the opposite direction. By applying the first transformation *L2Limm(D)* the translation of the e-marketplace’s modelling language occurs, hence, the model’s structure is now a language independent model (LIM). A similar step would be required for enterprise “A” (manufacturer) but it already has a representation of its model in LIMM format since that was done once it became part of the supply chain network.

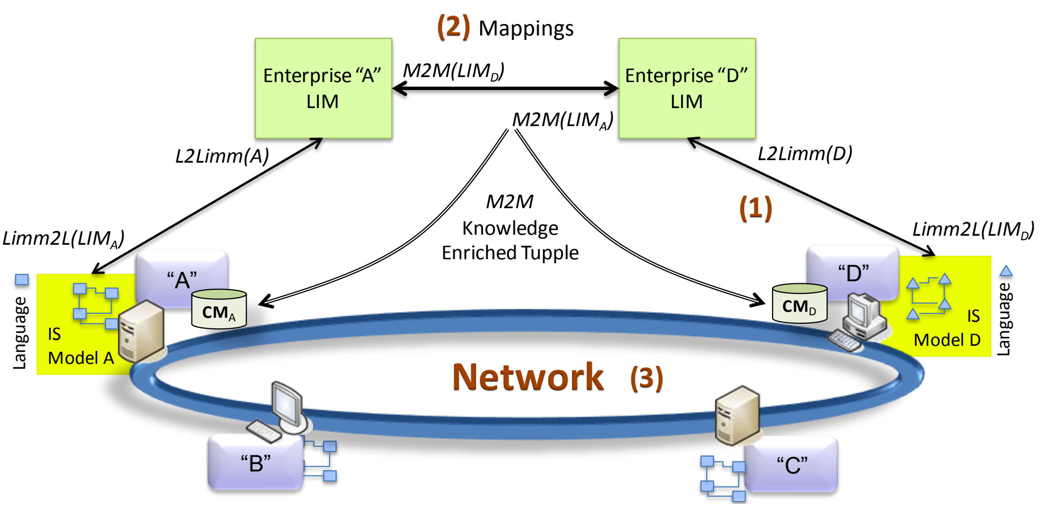


Figure 5.3: Interoperability Establishment Independently of Models and Languages

2. The second step starts once both enterprises have their models as LIMs (i.e. LIM_A and LIM_D), and the actual model mapping can begin. The relations between the both information models must be manually established by experts, linking all mappable nodes using the knowledge enriched tuple (MapT) of section 4.4.1, and registering them in a local knowledge base (i.e. the communication mediator) that contains all the mappings with the organisation's business partners. This enables gradual mapping on a need-to-serve basis since the CM can store the work progress. Through this step the second level of morphisms (*M2M*), at the meta-modelling level 1 (model), is obtained and is the basis for an interoperable automatic communication between "A" and "D";
3. Once the above steps are complete, all the executable transformation code stored in the local CM will be used to transform data in the bidirectional communication between each pair of enterprises and the tuple used for monitoring purposes. Having this, if more enterprises are required to enter/leave the network, it is possible to add, remove and edit all the mappings and the transformation code.

Regarding the POC, four test-cases have been selected for implementation within this scenario, to demonstrate some functionalities of the MDA-based package for interoperability establishment and the MAS package for sustaining systems interoperability.

Test-Case 1.1 – OWL language mapping

The first TC replicates the initial actions of the above explained step 1, i.e. defining the *L2Limm* mapping morphism (θ). Being a marketplace technologically oriented towards the semantic web, enterprise "D" has its information system model specified in OWL (W3C, 2009) while "A" uses a simple E-R database with XML interfacing. The first that needs to be done is getting them to abstract from language specificities, thus facilitating the understanding of connection points between the two IS models. Note that in the scenario "A" has already done this abstraction so the TC is focused on "D".

$$TC1.1\text{: } language\ mapping\ \theta(OWL, LIMM) = L2Limm \tag{18}$$

Test-Case 1.2 – language transformation

It is expected that each modelling language interfaced in the modelling language harmonising layer (depicted in section 4.3.2) of the MDA-based package for interoperability establishment is capable of being injected to a LIMM representation by means of automated model morphisms. This TC replicates the final actions of the above explained step 1, i.e. the execution of the transformation morphims that uses the mapping previously defined.

$$TC1.2\text{: } language\ transformation\ \tau(OWL, LIMM) = L2Limm(OWL) = LIM_D \tag{19}$$

Test-Case 1.3 – model mapping (P2P)

The third TC demonstrates that model morphisms are applied not only to language transformations but also to generate model to model transformations (above explained step 2). This functionality is the foundation of the inter-enterprise harmonisation layer (depicted in section 4.3.4) of the MDA-

based package for interoperability establishment, which is responsible to harmonise LIMM representations of heterogeneous information models. This TC demonstrates how MapT is defined and how information is stored in the enterprise “A” CM so that future product catalogues could be sent automatically to “D”.

$$TC1.3: model\ mapping\ \theta(LIM_{A,}\ LIM_{D}) = M2M_{AD}\ tuple \quad (20)$$

<u>Test-Case 1.4 – data transformation</u>

As Test-Case 1.2 is for TC1.1, this test-case is for TC1.3. This way, the final test case of this application scenario has the objective to simulate the actual data exchange between the two enterprises “A” and “D”. Following Figure 5.3 and inter-enterprise harmonisation layer description, the mapping between “A” and “D” is defined based on a LIM model, thus to execute the $M2M_{ad}$ transformation among LIM models it is required a previous transformation from the original input format, which is defined at the time of the language mapping. A transformation from LIM_D to D itself would be necessary to complete the transformation cycle but, for the sake of simplicity of this TC, a similar transformation will be presented later on a different TC.

$$TC1.4: data\ transformation\ \tau(A,LIM_{D}) = M2M_{ALimm}\ (A) + M2M_{AD}\ (LIM_{A}) = LIM_{D} \quad (21)$$

5.1.1.2. Scenario 2: Retailer “B” decides to adopt an international product data standard has its official data exchange format

The adoption of an international product data standard, such as the STEP application protocols, might provide a significant boost for manufacturing businesses. Indeed studies show that the use of STEP technology, whether in-house or between organisations, could generate significant savings per year in automotive, aerospace, and other sectors (PDES Inc., 2006). Through these numbers, STEP has proven its value to industry, thus it is only natural that if an AP exists for the sector, companies would like to try and adopt it.

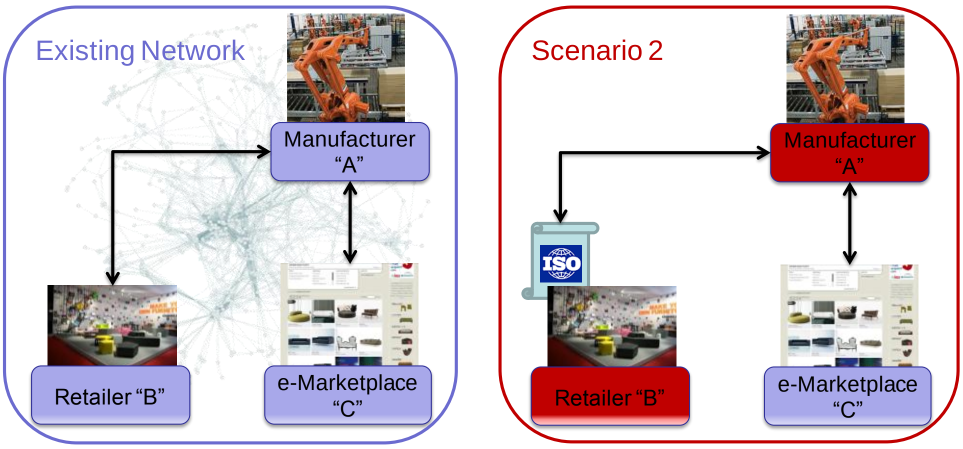


Figure 5.4: POC Application Scenario 2: Harmonization Breaking due to a Change of a Node’s Information Structure

In this scenario, retailer “B” is the same one presented in Figure 5.2 and is part of a supply chain as the one of Figure 5.1. Among the several networks it belongs to, it conducts business with many manufacturers, including “A”, thus if “B” decides to adopt a STEP AP to mediate all its data

exchange, the existing morphism between them needs to be updated (Figure 5.4). According to the IPyM classification of section 4.1, “B” is moving from a case of unregulated interoperability to standard-based, accelerated by the sustainability framework proposed. Despite not being addressed here, this is an opportunity to implement standard-based communications in the entire SC network.

Regarding the POC, another five test-cases have been defined for implementation within this scenario, to demonstrate some functionalities of the MDA-based package for interoperability establishment, the MAS package for sustaining systems interoperability, and also transient evaluation package. Some of the test cases are similar to the ones of the first scenario. Still some have been repeated (i.e. TC 2.1 and TC2.2) due to their significance and for future reference in the industrial validation of section 6.4 “Industrial Acceptance”.

<u>Test-Case 2.1 – EXPRESS language mapping</u>

The language mapping of EXPRESS language, i.e. STEP’ and AP236’ modelling language, to LIMM has been defined as a new TC due to its significance in terms of future adoption of the framework for self-sustaining interoperability. It is only normal that organisations already adopting an international standard for data exchange become unwilling to adopt or even test the proposed solution if that implies extra work for them. For this reason, the maximum number of language mappings need to be defined and included in the POC.

$$TC2.1\text{: } language\ mapping\ \theta(EXPRESS, LIMM) = L2Limm \tag{22}$$

<u>Test-Case 2.2 – versioning mapping</u>

The second TC of this scenario demonstrates how MapT is used for versioning’s and how that information is added to the one stored in the enterprise “B” CM so that it can evolve from the previous information structure to the STEP AP. This TC is to be implemented using furniture data, thus the elected AP is AP236 (ISO TC184/SC4, 2006).

$$TC2.2\text{: } versioning\ mapping\ \theta(LIM_B, LIM_{AP236}) = M2M_{B236}\ tuple \tag{23}$$

Despite being a different morphism it is still *M2M*, thus this TC is very similar to the one illustrated by TC1.3, leading to a transformation according to the previously defined model relationships. For this reason the transformation itself or the integration with LIM are not included in specific TCs.

<u>Test-Case 2.3 – reaction of the network to an harmonization breaking</u>

The change of the information structure of enterprise “B” has a direct impact in the network behaviour. In this particular case, the morphism between “B” and its direct business partners will be affected and stop being interoperable if not updated accordingly. This TC evaluates if the network detects this harmonization breaking and how it responds to “B” evolution in terms of M2M versioning. Note that the reaction of B is included in the analysis because the node also needs to readjust is own mappings.

$$\begin{gathered} TC2.3\text{: } harmonization\ breaking(B, network)\ \Rightarrow reaction(A)\ and\ reaction(B) \\ while\ reaction(x) =\ nothing, new\ M2M\ tuple, M2M\ tuple\ adjustment\ or\ M2M\ tuple\ deletion \end{gathered} \tag{24}$$

Test-Case 2.4 – decision and learning

As explained in the MAS package for sustaining systems interoperability (section 4.5), by proposing a readjustment of the network morphisms and a transient estimation to a decision maker on each enterprise affected by the harmonization breaking (e.g. "A" and "B" itself), knowledge previously acquired is used. Yet, the acceptance of the readjustment proposed will also provide knowledge that needs to be taken in consideration for future reference. This test-case enables to demonstrate the type of information given to "B" human decision maker and how MIRAI learns considering the decisions made.

$$TC2.4: decision\ (B, new\ M2M_{BA}, transient) = accept\ or\ reject\ or\ new\ M2M\ tuple \Rightarrow CM_B\ update \quad (25)$$

Test-Case 2.5 – data transformation

The final test case of this application scenario is very similar to TC1.4 but with the particularity to simulate the actual data exchange between two enterprises ("A" and "B") following the morphism evolution of TC2.4. A transformation from A to LIM_A itself would be necessary to complete the transformation cycle but, since a similar one was illustrated on TC1.4, it is not repeated here.

$$TC2.5: data\ transformation\ \tau(LIM_A, B) = new\ M2M_{AB}\ (LIM_A) + M2M_{LimmB}\ (LIM_B) = B \quad (26)$$

5.1.2. Collaborative Product Development in System Engineering Networks

As companies grow larger and products become more complex, hierarchical organisations and networks are established to master the increasingly large organisation sites, the technical complexity and the specialisation that is involved to master this complexity (Kamrani, 2008). Also, since companies are constantly trying to tap innovation and creativity across the globe, this growth also results in geographical dispersion of people and departments. Thus, whether it is for the creation of a new component or redesigning an existing product, companies are looking outside their boundaries for product engineering services. In this sense, collaborative product development (CPD) has become imperative to strengthen niche values and remain competitive in the global market (Chu et al., 2006).

However, collaboration for product engineering poses many challenges. According to PDMA[46], *"collaborative development differs from simple outsourcing in its levels of depth of partnership in that the collaborative firms are linked in the process of delivering the final solution"* (Jalali, 2009). To develop a product, an entire ecosystem must come together including the several engineering capabilities from design, simulation, production, etc., and involving multiple stakeholders from software houses, intellectual property vendors, suppliers, or manufacturers (external part of Figure 5.5). CPD aims at closely integrating cross-functional or cross-organisational teams in order to competitively develop products in shorter time periods, providing cost reductions, and improved quality. Requirements emerge from the need to have federated information coming from marketing, project execution, as well as from the PLC.

---

[46] Product Development and Management Association: http://www.pdma.org/

As studied in Athena IP (2005), independently of the sector it is applied to, CPD processes can involve: (i) coordination of the programme logic to deal with product development projects and tasks; (ii) the work-share logic to handle contractual issues, co-operating teams and multi-skills actors; and (iii) the product logic which is concerned with operation, integration and technology driven details (center part of Figure 5.5). Therefore, with so many variables and the fact that partners are likely to be geographically dispersed, it is important to have a single version of truth for all the participating stakeholders (Jalali, 2009).

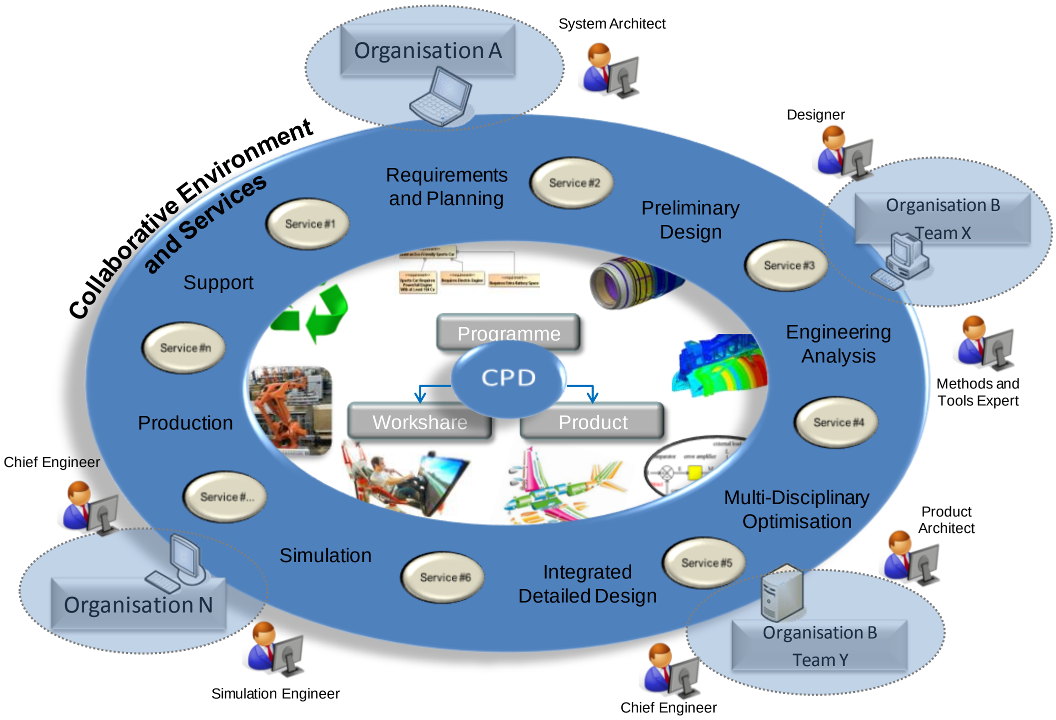


Figure 5.5: Environment and Services within a Network for Collaborative Product Development and Engineering

The topic of sustainability of systems interoperability within this type of networks is fully applicable, even if focusing only on the product logic view, which is the closest to the systems interoperability topic addressed in this dissertation. As stressed by (Kamrani, 2008), software integration is a complex process that requires a courtly solution and, despite significant advancements in the data-sharing techniques, the critical issue remains maintaining the relationships and dependencies among the different types of data originated by the different stakeholders involved in the CPD network. This is even more noticed in networks that do not use any central collaboration system. For instance, in a typical development process the design occurs largely before the final full-scale manufacturing and, in most cases this design is later altered or refined for the manufacturing difficulties or, problems detected at simulation time. In case different components are designed by different vendors, a single undetected change in one of them can affect the full product integration and the work of other teams in the network.

As in the case described for the supply-chain network, a number of different application scenarios could be drawn to support the work on sustaining systems interoperability within CPD networks. Nonetheless, since the SC scenarios and test cases are focused on software systems

interoperability and information models, here the concern is to complement the POC with issues also related to systems engineering, namely product development. Hence, the two scenarios that integrate the POC from the CPD network include software and data interoperability test cases:

- The first scenario of CPD (and 3rd of the POC) Demonstrates how the proposed self-sustainable interoperability framework can support the integration of an organisation in a collaborative design network;
- The second scenario, illustrates how the CPD network supports a design change in one of the components and reacts to harmonization breaking not in the systems but in the integrated product level;

5.1.2.1. Scenario 3: Supplier "D" joins a CPD network, bringing in a component whose design needs to be integrated into an assembled product in the detailed design stage

As already explained CPD practices are recognised as critical to the development of competitive products in today's dynamic market. Suppliers that coordinate the development and manufacturing of specific components are involved from the beginning in a network of global companies, where market-specific products are manufactured in the host country and assembled somewhere over the globe. This early involvement results in a complete understanding of all the requirements and a consensus approach to product design, manufacturing and support processes (Kamrani, 2008).

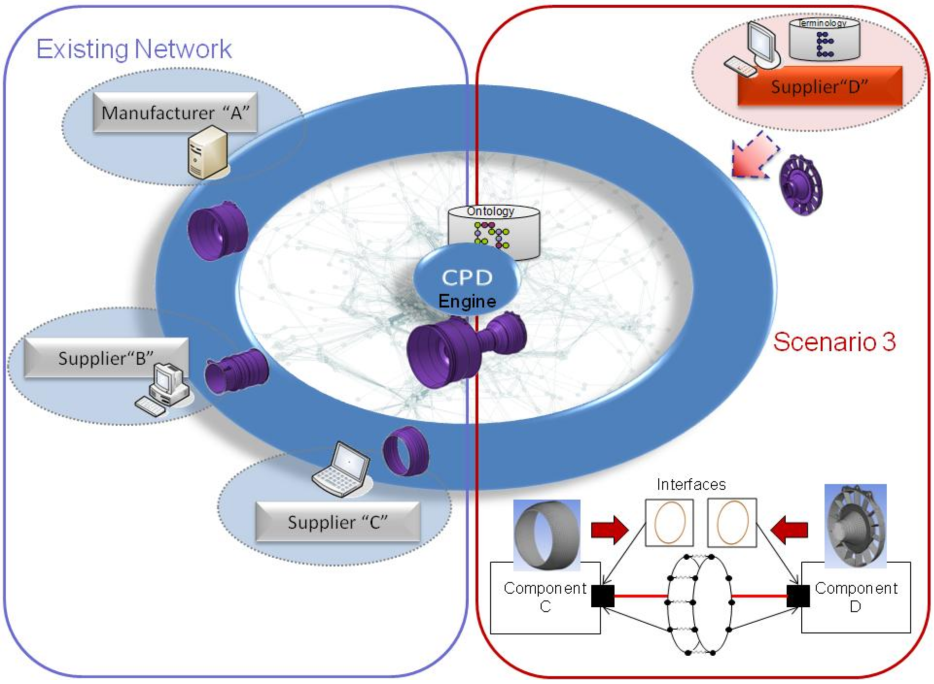


Figure 5.6: POC Application Scenario 3: Addition of a Note to the Network and Integration of Engineering Data

This scenario envisages the involvement of supplier "D" in a collaborative network (it could be using MBSE paradigm having all types of models integrated) for the development of an aircraft engine. The network already has three partners cooperating at the early design stages and engineering analysis but, to enable a simulation of the final product, an integrated design is required including a component provided by "D". The scenario illustrates not only the process of

interoperability establishment in terms of information systems structure ("D" joining the network and adopting a common collaboration model), but also in terms of terminology (conceptual morphisms – C2C) and product design as depicted in Figure 5.6, where the mapping among the different instances of components need to be defined (third level of morphisms - D2D, defined at the meta-modelling level 0, i.e. data).

The integration of different CAD[47]/CAM[48]/CAE[49] tools shortens the development process and optimises the design. Thus, scenario 3 considers the usage of an internal collaboration model (COLLAB) that facilitates the definition of mechanical interfacing points/surfaces among different components (i.e. mappings) independently of the tool used for the product design. This collaboration model is not the focus of the scenario, neither is part of the contributions of this dissertation, however it is important to notice that the principles behind it are similar to the ones of LIMM, allowing companies to focus on the mapping definition instead of being concerned with technical difficulties caused by different tool details (language details in the case of LIMM). According to the classification of section 4.1 "Interoperability Efficiency Pyramid Model (IPyM)", companies within this CPD network are on the semantic interoperability level due to the usage of a common ontological interoperability model, and this scenario enables to evaluate the sustainability framework in terms of support to model and data interoperability. Four additional test cases have been defined for implementation to demonstrate some functionalities of the MDA-based package for interoperability establishment and the MAS package for sustaining systems interoperability. Only the TCs relevant to illustrate the different tests concerning the first scenarios are included.

<u>Test-Case 3.1 – model mapping</u>

The first TC described for scenario 3 demonstrates the mapping definition of "D" with the collaboration model used by the CPD network. It is a TC similar to TC1.3 since it envisages the harmonisation of LIMM representations of both input and output models, not only at the structural level, but also at the semantic level. Doing this integration using inter-enterprise harmonisation layer of the MDA-based package for interoperability establishment provides the CDP network with the robustness required to support any changes at the enterprise systems level, e.g. a change of design software or version.

$$TC3.1: model\ mapping\ \theta(LIM_{D,}\ LIM_{Collab}) = M2M_{D\_Collab}\ tuple \qquad (27)$$

<u>Test-Case 3.2 – conceptual mapping</u>

With so many stakeholders and actors mastering the different engineering capabilities around the network, sometimes issues concerning the terminology can become a barrier to interoperability. Even if all work in the same domain, people might have different interpretations for the same term. To illustrate how that can be avoided using the CPD common ontology, this TC uses MapT for conceptual mappings, storing them in the enterprise "D" CM and enabling organisations collaborating in this CPD network to become semantically interoperable.

[47] Computer-Aided Design
[48] Computer-Aided Manufacturing
[49] Computer-Aided Engineering

$$TC3.2: conceptual\ mapping\ \theta(LIM_{D\_ONTOLOGY}, LIM_{Collab\_ONTOLOGY}) = C2C_{DCollab}\ tuple \quad (28)$$

<u>Test-Case 3.3 – data mapping (P2P)</u>

The third TC demonstrates how the knowledge enriched tuple for mappings representation can also be used for a different type of mapping, i.e. data mapping, which in this case enables to relate different component instances. For simplification, mappings are considered at LIM level.

$$TC3.3: data\ mapping\ \theta(LIM_D, LIM_C) = D2D_{DC}\ tuple \quad (29)$$

Since this mapping is defined at the lowest meta-modelling level (data) it will not enable any automatic data transformations. Nevertheless, the objective of relating two instances is only to establish a link between them and allow other stakeholders to become aware they are tightly connected. If meant for that purpose, this relationship can be specified at the CPD collaboration model, but if not stored in the company's CM the sustainability framework would not be able to detect an harmonization breaking situation (to be analysed in TC 4.2) neither benefit from the other packages functionalities.

<u>Test-Case 3.4 – data transformation</u>

As in TC1.4, the goal of this test is to simulate the actual data exchange. In this case what will be simulated is the exchange of data from enterprise "D", which uses OWL, with the collaboration system, so that another company ("C" according to Figure 5.6) can use its own system to perform additional tasks on the integrated product. For simplicity reasons, the transformation stops with data from "D" at the LIM representation of COLLAB.

$$TC3.4: data\ transformation\ \tau(D, LIM_{CollabD}) = M2M_{DLimm}(D) + M2M_{D\_Collab}(LIM_D) + C2C_{Dcollab}(COLLAB_D) = LIM_{CollabD} \quad (30)$$

Because this network is operating at a semantically interoperable level, a terminology transformation (C2C) can be required as well.

5.1.2.2. Scenario 4: Supplier "C" analyses and proceeds with a change in the design of its component to improve simulation results

A hierarchical distribution of work is essential for large organisations and complex products (Kamrani, 2008). This way it becomes evident that different stakeholders have different roles with the CPD network, share responsibilities and conduct their activities in parallel, reducing costs considering a time consuming synchronous relationship. However to make this working methodology a success, they must remain interoperable at the cost of wasting weeks of work if the data becomes outdated.

This scenario (see Figure 5.7) considers that a simulation of the assembled product performed by manufacturer "A" (a large OEM) demonstrated that the product was not performing as expected. Receiving this information, supplier "C" decides to try a reformulation of its component, updating its design without considering the previously defined interfaces with the components from "B" and from "D". Scenario 4 contributes with two more test cases for the POC, focusing the MAS package

for sustaining systems interoperability. They are different from the ones previously analysed as the harmonization breaking occurs at the data instead of model level.

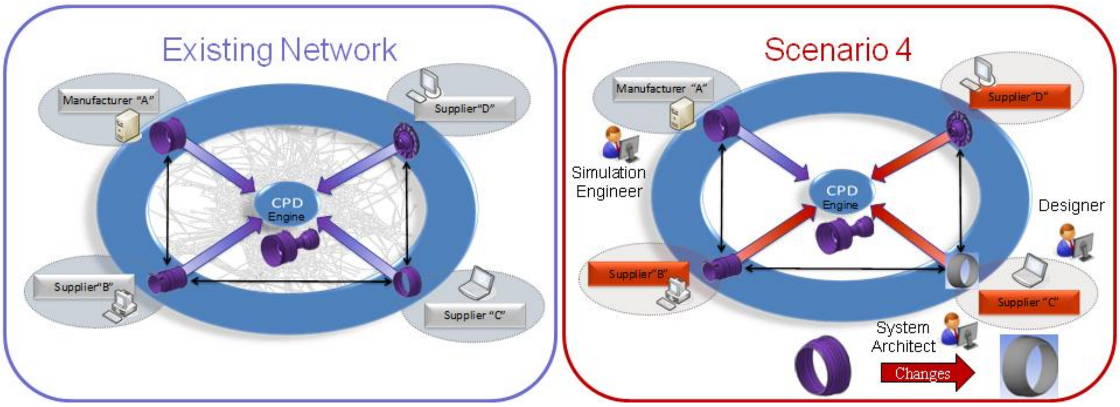


Figure 5.7: POC Application Scenario 4: Harmonization Breaking due to a Change of a Node's Component Design

<u>Test-Case 4.1 – versioning mapping</u>

The first TC of this scenario demonstrates how MapT is used for data versionings and, how that information is added to the one stored in the enterprise "C" CM, so that it can evolve from the previous design structure ensuring traceability in case the new design does not work as expected.

$$TC4.1: data\ versioning\ mapping\ \theta(LIM_C, LIM_{C'}) = D2D_{CC'}\ tuple \quad (31)$$

This is a morphism similar to the one of TC2.2, but its inclusion is important to evaluate how the sustainability framework reacts to a data harmonization breaking (next TC).

<u>Test-Case 4.2 – reaction of the network to an harmonization breaking</u>

The change of the design data from component "C" has a direct impact in the network results, namely on the integrated product. In this particular case, the D2D morphisms between "C" and its direct business partners will be affected and stop being interoperable if not updated accordingly. This TC evaluates if the network detects this harmonization breaking and how it responds to "C" evolution in terms of a D2D versioning.

$$\begin{gathered} TC4.2: harmonization\ breaking(C, network)\ \Rightarrow reaction(C)\ and\ reaction(B)\ and\ reaction(D) \\ while\ reaction(x) =\ nothing, new\ D2D\ tuple, D2D\ tuple\ adjustment\ or\ D2D\ tuple\ deletion \end{gathered} \quad (32)$$

5.1.3. Summary of Scenarios and Test Cases

Summarizing, 15 test cases have been identified along 4 scenarios to demonstrate a variety of situations where the framework for self-sustaining interoperability is applicable, either in the establishment of interoperable relationships, sustaining those relationships, or even real data exchange. As pointed in Table 5.1, three of the test cases (TC3.2, TC3.3, and TC4.1) simulate mappings that, despite being possible through the package for interoperability establishment, envisage topics not core to this dissertation (e.g. conceptual mapping for ontology building), and are included only with the purpose of highlighting the multidisciplinary use of the CM and the knowledge enriched tuple for mappings representation (structure, concepts, data). Finally, from all TCs, there are 2 situations where harmonization breaking should be detected (TC2.3 and TC4.2).

The transient evaluation package for minimizing network impact (presented in section 4.6) is the least represented package with only one TC associated. This is due to the fact that research is still at methodological level, and despite integrated in a CAS-SIF layer, no dedicated POC implementation to support massive testing is envisaged at this point. Therefore, it is illustrated only in a simplified TC situation where the decision maker can, depending on the transients associated, opt among different harmonization breaking responses. Additionally, and already considering subjects for future work one could define TCs to analyse the network response to a new node addition (e.g. in scenario 1), as well as TCs to simulate the reaction of a network to a node deletion (e.g. due to enterprise insolvency). Up to now the only possible response is the removal of all morphims connected to that specific node. A more helpful and adequate response would require metadata to describe the node's business activities and involve searching of possible replacements in neighbouring networks.

5.2. Implementation of the MDA-based Package for Interoperability Establishment

The main objective of this proof-of-concept consists on the implementation of the MDA-based package for interoperability establishment (previously specified in section 4.3) which, according to the scenarios described is to be used across 11 TCs. This POC enables the specification of mapping and versioning morphisms, their storage in the communications mediator knowledge base, and the corresponding execution of data transformations.

By the analysis of the TC requirements, one concludes there is the need to define LIMM mappings for two different languages (level 2 of the MDA), i.e. OWL and EXPRESS, and six mapping situations including two level 1 model mappings and the corresponding data transformations, and two versionings. The implementation of the inter-enterprise harmonization layer should support as well the definition and storage of conceptual and data mappings, thus enabling self-sustainable interoperability on top of semantic interoperability as envisaged in the IPyM model of section 4.1.

5.2.1. Technical Instantiation of the Proposed MDA Architecture

Given the context of MDA, QVT is the standard transformation language proposed by OMG. However, considering the languages analysed in the literature review (section 2.4.2 "Executable Transformation Languages"), ATL has currently the largest user-base and the most extensive information available such as reference guides, tutorials, programmers' forums, etc. As evidenced by Jouault & Kurtev (2007), it is a largely used language to implement MDA based tools, having a specific development toolkit plug-in available in open source[50]. By all these reasons it was decided to use ATL to implement model and language transformations in the scope of the MDA-based package for interoperability establishment.

[50] Eclipse Modelling Project - http://www.eclipse.org/modeling/

Table 5.1: Matching Scenarios and Test Cases with the Technical Packages of the Framework for Self-Sustaining Interoperability

Business Situation	Scenarios	Test Cases	MDA-based Package for Interoperability Establishment	MAS Package for Sustaining Systems Interoperability	Transient Evaluation Package for Minimizing Positive Feedback Loops
Supply Chain in Manufacturing Networks	**Scenario1:** Manufacturer "A" wants to enlarge network and publish product catalogues in another e-marketplace	$TC1.1: language\ mapping\ \theta(OWL, LIMM)$	X		
		$TC1.2: language\ transformation\ \tau(OWL, LIMM)$	X		
		$TC1.3: model\ mapping\ \theta(LIM_A, LIM_D)$	X		
		$TC1.4: data\ transformation\ \tau(A, LIM_D)$	Data exchange		
	Scenario 2: Retailer "B" decides to adopt an international product data standard has its official data exchange format	$TC2.1: language\ mapping\ \theta(EXPRESS, LIMM)$	X		
		$TC2.2: versioning\ mapping\ \theta(LIM_B, LIM_{AP236})$	X		
		$TC2.3: harmonization\ breaking(B, network)$		X	
		$TC2.4: decision\ (B, new\ M2M_{BA}, transient)$		X	X
		$TC2.5: data\ transformation\ \tau(LIM_A, B)$	Data exchange		
Collaborative Product Development in System Engineering Networks	**Scenario 3:** Supplier "D" joins a CPD network, bringing in a component whose design needs to be integrated into an assembled product in the detailed design stage	$TC3.1: model\ mapping\ \theta(LIM_D, LIM_{Collab})$	X		
		$TC3.2: conceptual\ mapping\ \theta(LIM_{D\_ONTOLOGY}, LIM_{Collab\_ONTOLOGY})$	Optimal semantic packages[51] are not part of this thesis (goal here is to show MapT capability to represent other mappings)		
		$TC3.3: data\ mapping\ \theta(LIM_D, LIM_C)$			
		$TC3.4: data\ transformation\ \tau(D, LIM_{CollabD})$	Data exchange		
	Scenario 4: Supplier "C" analyses and proceeds with a change in the design of its component to improve simulation results	$TC4.1: data\ versioning\ mapping\ \theta(LIM_C, LIM_{C'})$	Optimal semantic packages are not part of this thesis (goal here is to show MapT capability to represent other mappings)		
		$TC4.2: harmonization\ breaking(C, network)$		X	

[51] See Sarraipa et al. (2010) and (CRESCENDO Partners, 2010)

Although ATL transformation input models can be represented in plain text, it is preferable to use previously validated XMI serialised meta-models conforming to the MOF meta-meta-specification (level 3 of the MDA). Regardless of not all languages have their specifications available as a MOF model, both EXPRESS (OMG, 2010b) and OWL 2.0 (W3C, 2008) do. However, this last one is not yet a recognised OMG version and some improvements, supporting the development of this POC, have been conducted towards matching exactly the language specification (W3C, 2009). Corrections include some misnamed elements, defining some missing relationships, etc.

Finally, the MDA-based package for interoperability establishment, envisaged the storage of all morphisms at a local knowledge base (CM) which, due to the large reasoning potential that can be required in future work, has been implemented using OWL technology for ontologies. Protégé[52] was the open source software used for the CM creation. With a community of thousands of users, Protégé's internal mechanism for ontologies has been used successfully in many application areas.

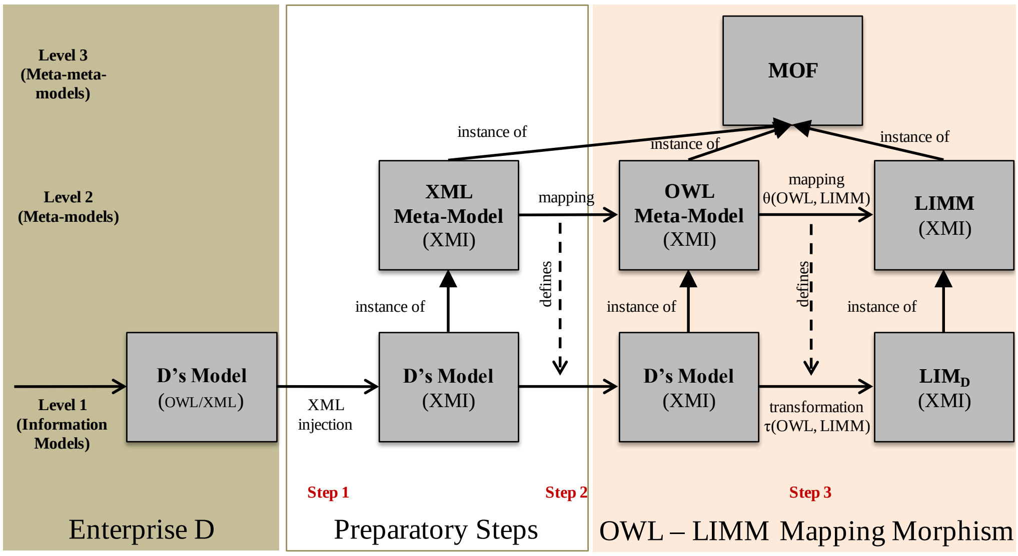


Figure 5.8: OWL Instantiation of the Modelling Language Harmonization Layer

5.2.1.1. Modelling Language Harmonization Layer

This layer's implementations are responsible for answering requirements of the following TCs:

- TC1.1 "$language\ mapping\ \theta(OWL, LIMM)$" – To enable a mapping among the OWL meta-model and the LIMM, one needs firstly to put the OWL data in an XMI serialisation following the OWL meta-model specifications. Nevertheless the procedure to do so is not as straightforward as desirable since, in spite of the inputting OWL model is already XML serialised, it cannot be directly processed by the ATL toolkit which needs XMI as an input. The complete process for accomplishing the language mapping test case is illustrated in Figure 5.8, 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).

---

[52] http://protege.stanford.edu/plugins/owl/community.html

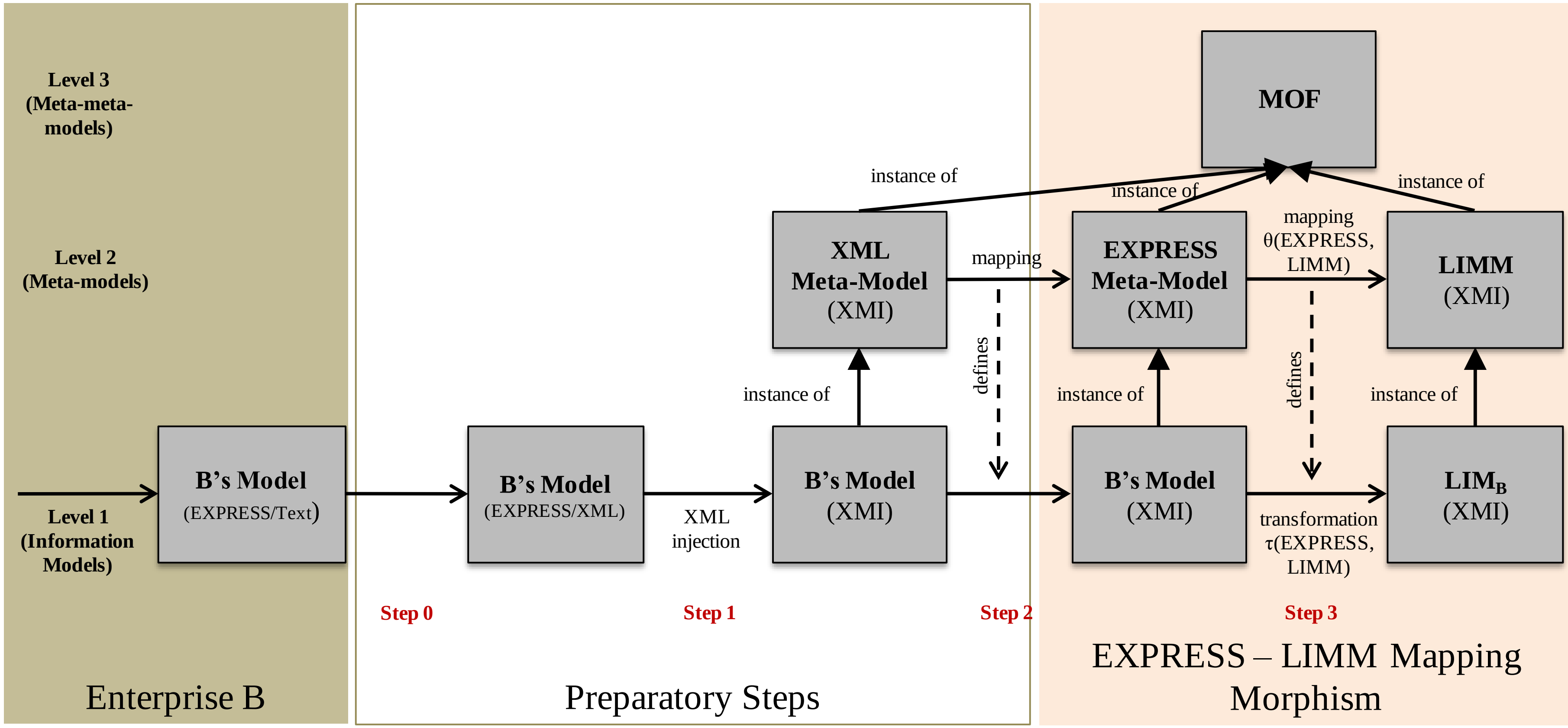


Figure 5.9: EXPRESS Instantiation of the Modelling Language Harmonization Layer

- TC1.2 "$language\ transformation\ \tau(OWL, LIMM)$" – According to the architecture specified for the package for interoperability establishment, this test case is a direct consequence of the previous. In fact, by using ATL as the MDA language, one is at the same time specifying the mapping and defining the transformation rules (illustrated under step 3 of Figure 5.8).
- TC2.1"$language\ mapping\ \theta(EXPRESS, LIMM)$" – This test case is in theory very similar to the TC1.1 already described, but in practice it needs an extra preparatory step (marked as step 0 of Figure 5.9) since the EXPRESS modelling language, where STEP product data standards are described, is given in a text format. The use of a XML serialisation simplifies the process of representing the input models as instances of the EXPRESS meta-model. It can be natively injected by the ATL modelling tools, creating a valid XMI serialized instance of itself, and the process can follow the exact same steps as in TC1.1.

It is important to note that the reverse process implies reverse mapping definitions, transformations and XML extraction from XMI. Hence, after defining the modelling language harmonization layer instantiated it becomes clear that XMI is another technology which plays a fundamental role through the whole workflow of transformations. All MOF based instances in the context of these ATL modelling tools are serialised in XMI, no matter if it regards to the meta-models, models or even data instances (MDA level 0). Still, to be usable by the ATL rules, the available EXPRESS XMI meta-model had to be imported to an UML tool (UML MagicDraw[53] was used) and converted to an ECORE representation of XMI (Eclipse EMF approach of MOF, used to represent meta-models in the EMF framework).

5.2.1.2. Inter-Enterprise Harmonization Layer

This layer's implementations are responsible for answering requirements of the following test cases:

- TC1.3 "$model\ mapping\ \theta(LIM_{A}, LIM_{D})$" – LIMM has the unusual capacity of storing both model and data instances within the same physical file, in resemblance to what happens with OWL. It potentiates the actual data transformation at a language independent form as well, thus avoiding the definition of mapping morphisms at this abstract level, which would have to be reengineered back to the original model languages. This integration of model and data maintains a forward flow of activities from company "X" to the abstract interface, and from there to the company "Y". However, as illustrated in Figure 5.10, before the definition of the model mapping (step 5), similar steps as the ones conducted for the modelling language harmonization layer need to be followed to append data into the LIM model (step 4). For this TC, since enterprise "A" was already part of the network, that preparatory step 4 is not required.

---

53 https://www.magicdraw.com/

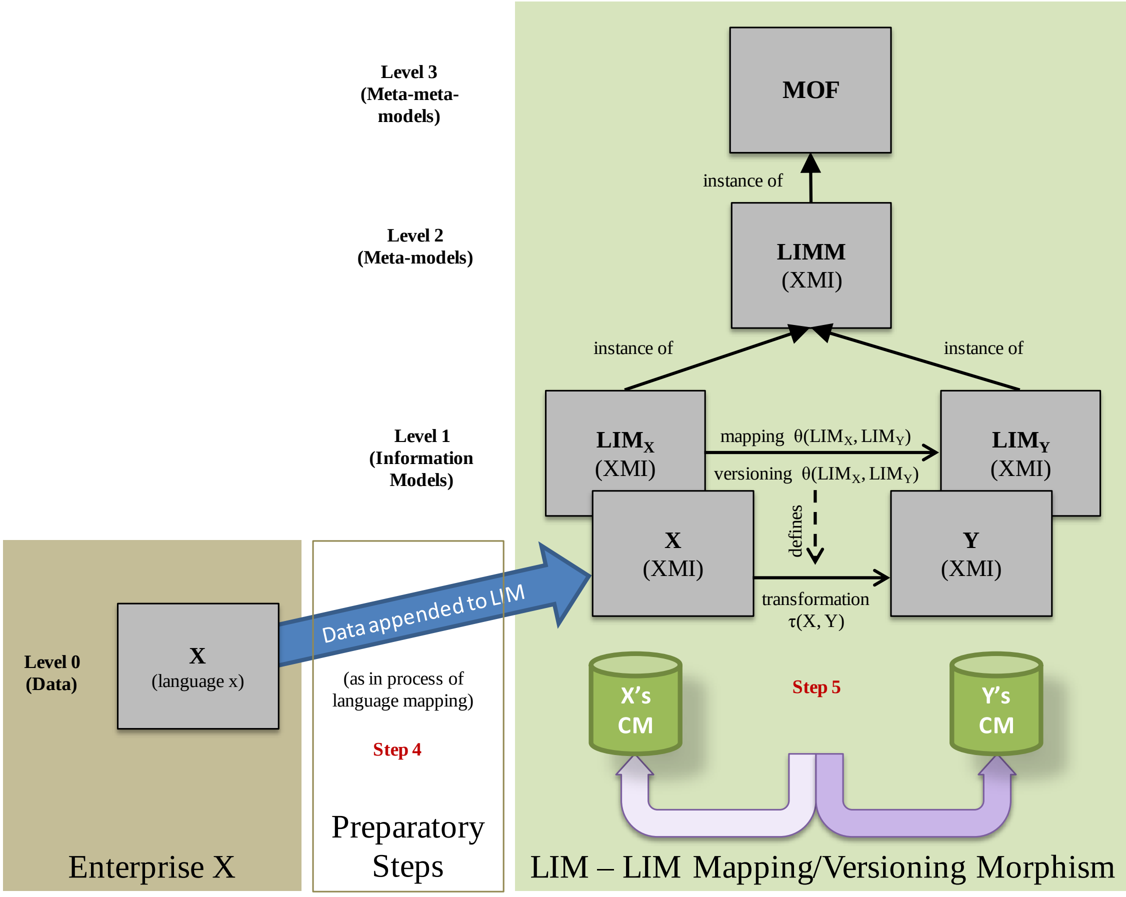


Figure 5.10: Instantiation of the Inter-Enterprise Harmonization Layer

- It is very important to preserve the user's technology abstraction, envisaged by LIMM, thus the mapping process is supported by a collaborative tool capable of visualizing and interacting with models and concepts in a way that model element' relationships and dependencies are easily understood by domain actors with no knowledge of technical rules. For this purpose, graph-like visualization tools have been analysed, not being associated with other types of technical diagrams (e.g. UML, etc.). Among the many analysed, the open source JGraph[54] has been elected and modified to read LIM model files and store morphisms at the CM without any difficulty to the end user. This tool enables the configuration of many styles of shapes and edges, validation of connections, and a built-in undo and redo manager. With it LIMM's *Entity_types*, *Type_Concepts*, and *Instance_Groups* can assume different node shapes, leaving the edges for *Property* declarations.
- TC1.4 "$data\ transformation\ \tau(A, LIM_D)$" - According to the architecture specified for the package for interoperability establishment, this test case is a direct consequence of the previous, i.e. the mapping specified also defines the transformation rules (step 5 of Figure 5.10).

[54] http://www.jgraph.com

- TC2.2 "*versioning mapping* $\theta(LIM_B,\ LIM_{AP236})$" – A model versioning morphism is a special case of mapping that needs to relate the previous version of a model with the new one, so that data can be automatically migrated from one version to the other. Hence TC2.2 implementation also follows the instantiation depicted in Figure 5.10 with the exception that by being a versioning, means that "B" was already fully integrated (model and data) with LIMM, and step 4 is not required as well.
- TC3.1 "*model mapping* $\theta(LIM_D,\ LIM_{Collab})$" – This test case elicited the totality of requirements expressed in Figure 5.10. As enterprise "D" is joining the network and establishing interoperability with LIMM for the first time, requires the implementation of both step 4 and 5.
- TC3.2 "*conceptual mapping* $\theta(LIM_{D\_ONTOLOGY},\ LIM_{Collab\_ONTOLOGY})$" – As evidenced in Table 5.1, conceptual mapping for reference ontology building is not a core concept to this PhD. However it was included due to the interest in demonstrating the large spectrum of applicability of the tuple defined for the formal representation of morphisms (MapT). Also, by illustrating this type of mappings complementary to the ones of TC3.1, it becomes possible to simulate the framework for sustaining systems interoperability on top of semantic interoperability. Nonetheless, in resemblance to what happens in models, conceptual mapping may enable (but not in all cases) automatic transformation of data terminology (step 6 of Figure 5.11) in a semantically concerned network.

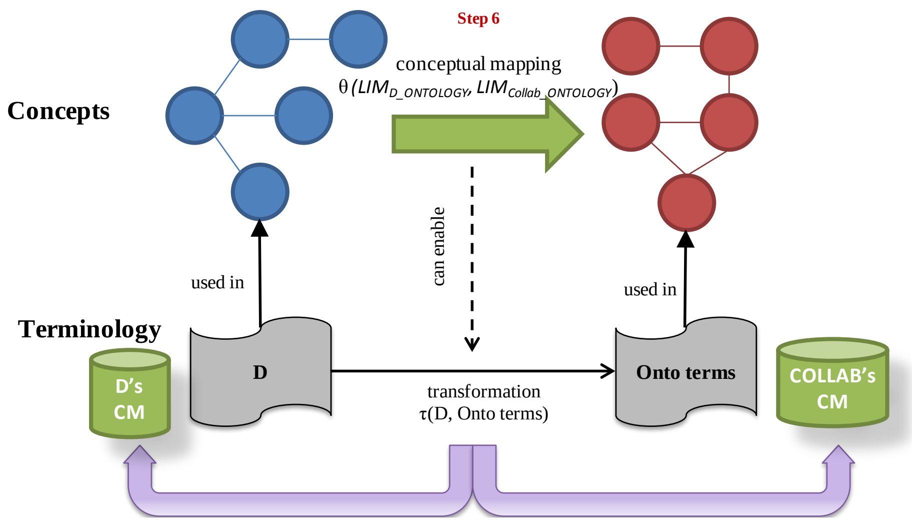


Figure 5.11: Instantiation of the Inter-Enterprise Harmonization Layer for Conceptual Mappings

- TC3.3 "*data mapping* $\theta(LIM_D,\ LIM_C)$" – This test case is also not exploring a core concept to this PhD. In fact, data mapping (step 7 – not illustrated in any figure) does not enable any kind of subsequent transformations' automation because it can be as simple as to establish a connection among real life objects. They just become related to perform some kind of activity, but according to the MDA levels, zero is already the lowest abstraction. However, in spite of this, data mappings are included as a test case since they are relevant for industrials to become aware of future changes (harmonization breaking) in

the relationship of physical objects that can cause similar damage as an interoperability-type harmonization breaking.

- TC3.4 "$data\ transformation\ \tau(D,LIM_{CollabD})$" – As in TC1.4, this test case is a direct consequence of the mapping defined in TC3.1, i.e. according to the architecture specified for the package for interoperability establishment, the model mapping also defines the transformation rules (step 5 of Figure 5.10). However this transformation morphism addresses a scenario of semantic interoperability and besides being enabled by steps 1 to 5, it is complemented with step 6 of Figure 5.11.
- TC4.1 "$data\ versioning\ mapping\ \theta(LIM_C,LIM_{C'})$" – Is a test case defined to complement TC3.3 and enables the subsequent testing of data harmonization breaking (TC4.2) in the package for sustaining systems interoperability. It does not have any special technical requirement for the POC since in LIMM instances (data) are handled as any other concept. This mapping procedure uses the same implementation of the interoperability establishment package, with technology independent structures and visualization/interaction requirements, but just does not benefit from MDA.

5.2.1.3. Technology Used

To summarise the most important technologies and tools (re)used throughout the proof-of-concept implementation, a short objective oriented table is presented next (see Table 5.2). Most of them have already been identified in the test case technical instantiation, but some will only be introduced in the steps implementation.

5.2.2. Step 0: XML Serialisation of EXPRESS Text Models

At the current date, a public transformation of EXPRESS ASCII text to MOF meta-model instance does not exist, so the architecture instantiation had to be developed bottom up from the preparatory steps to the transformations of step 3.

Eurostep EXPRESS Parser[55] (EEP) is a freeware command line EXPRESS tool to support the development of STEP application protocols. It enables EXPRESS models to be verified against the EXPRESS language specification (ISO TC184/SC4, 2004) and can also be used to generate a "pretty printed" form of the verified EXPRESS model in an XML serialised document. Therefore, EEP has been used to perform the execution of this preparatory step.

Every input model external to the interoperability establishment package is considered to be valid, coherent and complete. Hence, every external EXPRESS model must pass the EEP validation or will face the consequence of immediately failing the whole transformation process. Yet, the use of EEP is justified with other extremely important feature: the XML tag formatting export (serialisation). Additionally, EEP allows multiplatform support and integration, e.g. Eclipse.

Figure 5.12 depicts step 0's input and output. From now on, the same example is used to increase comprehension throughout all steps (except otherwise explicitly identified).

---

[55] http://www2.pdteurope.com/global/solutions/download-software.aspx

Table 5.2: Technology (re)used in the Interoperability Establishment Package POC

Technology			Tools/Models		
	Purpose	Step		Purpose	Step
MDA	Foundation for models' transformations	1..5	M2M Eclipse Modelling Project	ATL plug-in (ANT enabled) and starting examples	1..6
ATL	Executable transformation language for model morphisms	1..6	OWL 2.0 meta-model	Meta-model complemented and used to define the language mappings	3
ECORE	Approach of MOF implementation by Eclipse EMF	1..6	EXPRESS meta-model	Meta-model used to define the language mappings	3
XMI	Format for models and data interchangeable representation within the package	1..6	MagicDraw UML	UML designing tool for UML class diagrams definition and export to other formats (e.g. ECORE XMI)	3
OWL	Format for knowledge bases representation - used in the CM	5..7	Protégé	Ontologies tool for the CM design	5..7
JAVA	Programming language to modify and integrate the JGraph based tool with the CM	5..7	JGraph	Graph-like LIM visualization and graphical mapping establishment	5..7
			JAXB	JAVA XML binding framework to parse LIM with Java (integration with JGraph)	5..7
			JENA	OWL API to persist data into the CM	5..7
			EEP	Validator and parser of text EXPRESS models to XML	0

5.2.3. Step 1: XML Injection

The ATL transformations that implement the mappings defined on step 3 must be applied to instances of well defined MOF meta-models (XMI serialised). However, at this stage the input OWL and EXPRESS models are available only in XML and, therefore need to be instantiated into MDA level 2 XMI conforming meta-models. This procedure is called injection and is natively supported by the ATL Eclipse engine used. In the POC implementation this injection is performed at the "LoadModel" ANT[56] Task execution, i.e. when the model is actually being loaded from the file and before applying any transformation. This XML injection takes all elements, attributes and text from the input XML model and transforms it into an instance of the XML meta-model (Figure 5.13).

[56] http://ant.apache.org/

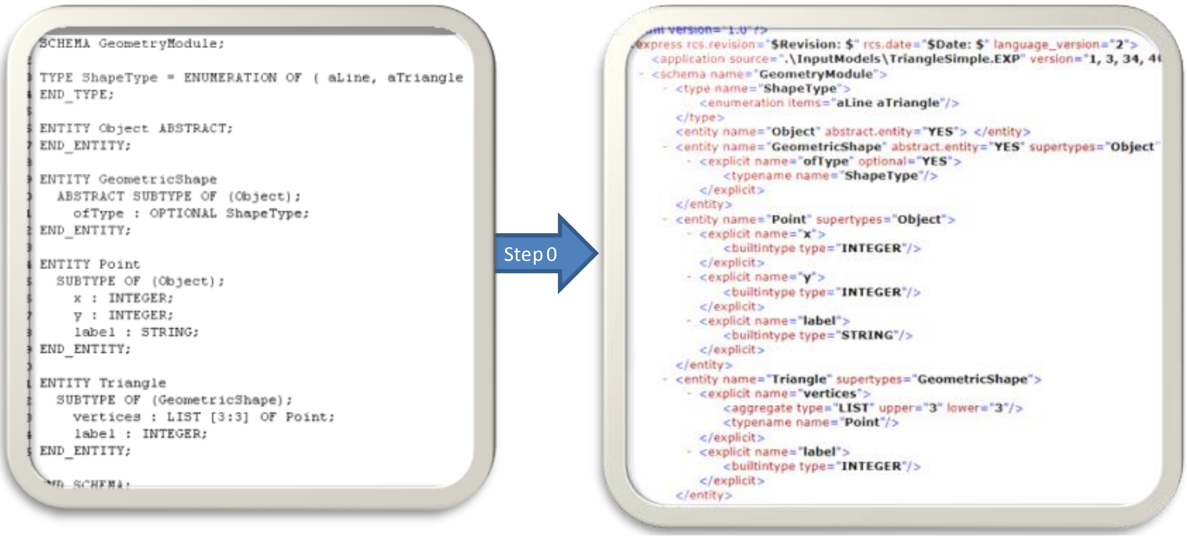


Figure 5.12: Example of Step 0 Input's and Output's

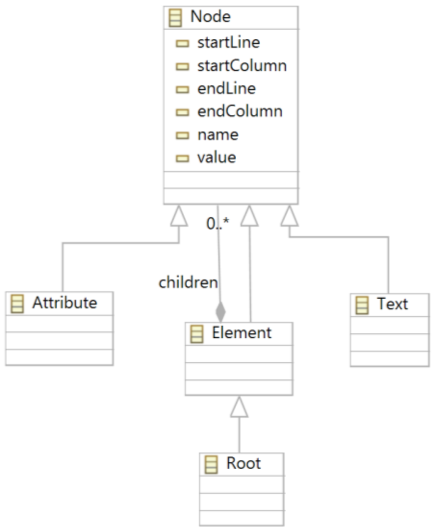


Figure 5.13: XML Meta-Model (from Rosendal (2005))

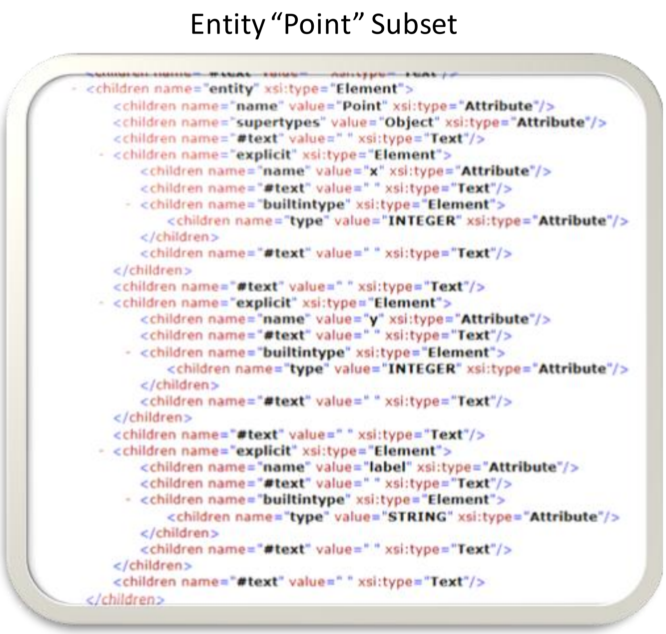


Figure 5.14: Example of Step 1 Output's

The XML meta-model is composed by only five classes and, as in step 0, no information from the original file is lost, since only the structure of the input is changed to follow the "Element-children" relationships of the meta-model illustrated. The result of applying step 1 to the EXPRESS-based geometry model example used before can be seen on Figure 5.14.

5.2.4. Step 2: XML – Input Language Meta-Model Mapping and Transformation

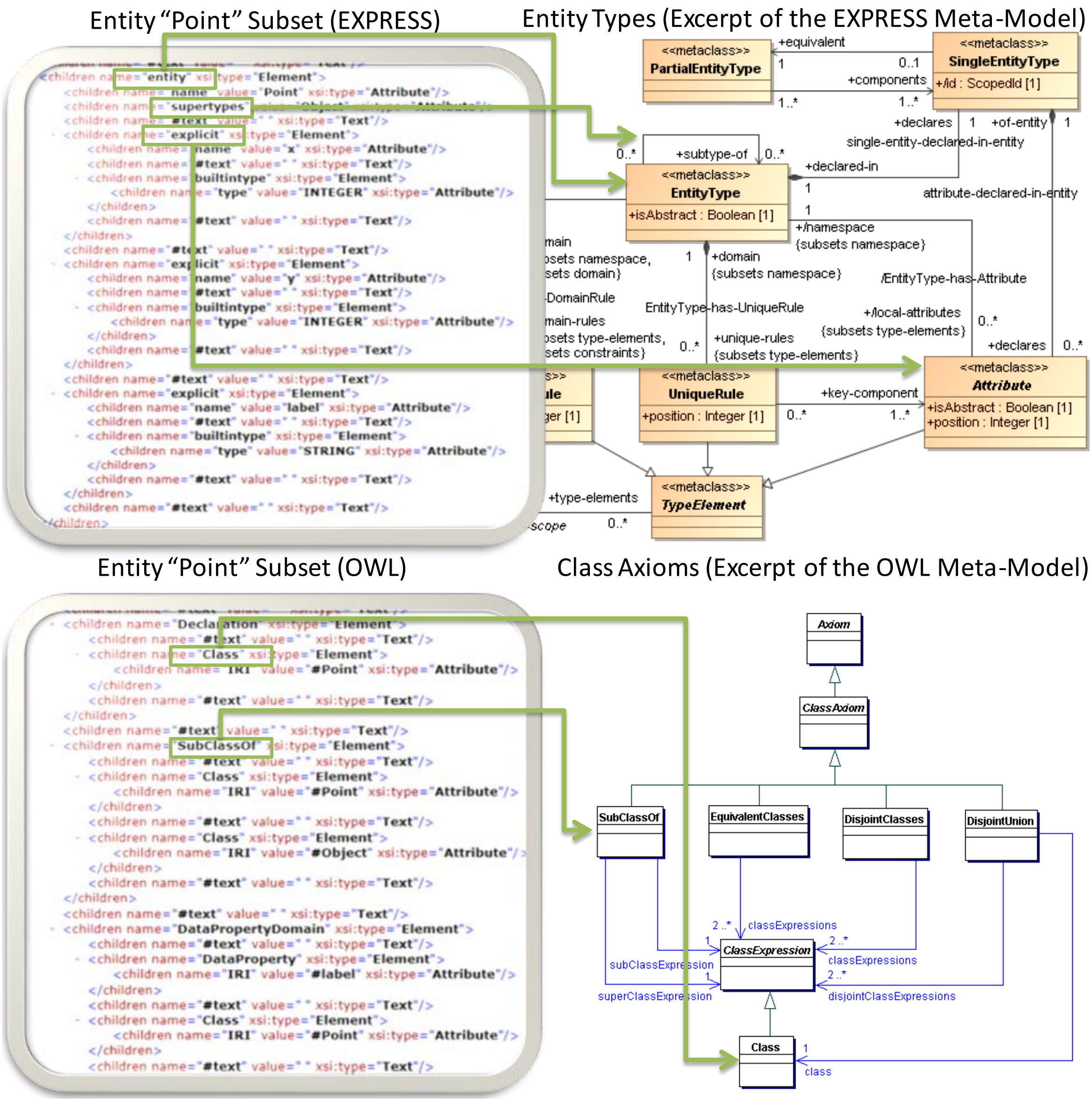


Figure 5.15: Example of Step 2 Mapping's

Now that, since the input is available in a serialized MOF compliant XMI, ATL can parse and execute all the mappings and transformations required. However, the input is now organised according to the structure exemplified in Figure 5.14 and not yet according to the input language meta-models (i.e. OWL or EXPRESS, as defined in the test cases). Thus, the final step before the actual language mapping definition is to map and transform that structure into the corresponding meta-model. This step is executed through ATL that navigates in the document structure in search for the mapeable elements (e.g. children.name="entity" maps to the EXPRESS meta-model *EntityType*, or children.name="class" maps to the OWL meta-model *Class*). The process is more or less straightforward as the names used at the input are similar to the ones in the meta-model concepts, and the elements follow a comparable structure (see example on Figure 5.15).

5.2.5. Step 3 and 4: Language Mappings Definition and LIMM Instantiation

In section 4.3.3 the language independent meta-model is defined as the basis for achieving the modelling language harmonisation between heterogeneous models (goal of step 3). This is achieved by mapping the desired modelling language, i.e. meta-model, with the LIMM. By creating these “translators” for each modelling language, the inter-enterprise harmonisation layer enables, in later steps, to integrate different organisation information structures using high level concepts of each modelling language analysed, with as little information loss as possible. Therefore, the quality of the mappings between the language meta-models and LIMM are the foundations for the steps towards the state of interoperability desired. The relations established allow to simplify proprietary complex meta-models, which scatter information across a multitude of classes and structures, narrowing it to simple high-level concepts easier for human analysis and transformations’ automation.

Two bi-directional mappings between LIMM and other modelling languages were defined for the POC, namely with EXPRESS and OWL languages, as specified by the test case requirements. These mappings, in a first stage, are represented in a table where a correspondence is setup between the modelling language meta-model concepts and the LIMM ones. MapT tuple, of section 4.4.1 could be used, but it is not required as it was thought for situations were the information loss (mismatches) is high and requires traceability.

5.2.5.1. EXPRESS – LIMM Mapping Morphism

Table 5.3 is an extract of the complete mapping for the EXPRESS concepts. In the first column, third row (shadowed in grey), EXPRESS concepts are selected from the meta-model and specialised through the various types they support (e.g. *SchemaElement* as *EntityType*, where *EntityType* is sub-type of *SchemaElement*). The same happens for LIMM concepts on the second column.

In the following rows, the various attributes belonging to the selected concept (in this case *EntityType*) are enumerated and mapped to the corresponding LIMM element(s). The notation used allows the identification of the origin and destination of a class relationship, e.g. “(Attribute)(EntityType.local-attributes)”, means that the “local-attributes” which belong to the *EntityType* class, link with *Attribute* class. Between brackets is defined the type of the concerning class, represented by the property path, much like the explicit casts in programming languages (e.g. ANSI C).

Although the mapping morphism definition tried to cover the maximum information as possible, some particular concepts were impossible to represent, either because they weren’t relevant to the expressiveness of the models (e.g. the EXPRESS concept *InvertibleAttribute*) or due to functional behaviour constraints which would lose their value when translated to other modelling language (e.g. the EXPRESS package Expressions). The representation of these “ignored” concepts or properties are also explicitly included in the mapping tables, as it can be observed in the few first lines of Table 5.3, where *InvertibleAttribute*, *RangeRole*, *DomainRole* and *UniqueRule* marked as “NOT MAPPED”.

Table 5.3: EXPRESS' "EntityType" Mapping to the LIMM (mapping extract)

EXPRESS Meta-model (OMG, 2010b)	**LIMM**
For each:	
EntityType ***(SchemaElement as EntityType)***	**Entity_Concept** ***(Concept as Entity_Concept)***
(InvertibleAttribute) (EntityType.attributes)	NOT MAPPED
(RangeRole) (EntityType.play-range-role)	NOT MAPPED
(DomainRole) (EntityType.plays-domain-role)	NOT MAPPED
(UniqueRule) (EntityType.unique-rules)	NOT MAPPED
(*(ScopedId)* (EntityType.id)).localname	Entity_Concept.name
EntityType.isAbstract	Entity_Concept.isAbstract
(EntityType) (EntityType.subtype-of)	*(Entity_Concept)* (Entity_Concept.isSpecificationOf)
(Attribute) (EntityType.local-attributes)	*(Property)* (Entity_Concept.contains)
(Redeclaration) (EntityType.redeclarations)	*(Property)* ((*(Property)* (Entity_Concept.contains)).isARedeclarationOf)

The full EXPRESS meta-model is divided in a series of packages: "Algorithms", "Core", "Enumerations", "Express2", "Expressions", "Instances", "Rules" and "Statements". From all these, and as illustrated in Figure 5.16, many have not been mapped due to the same reasons supra cited, but "Core" and "Enumerations" packages' concepts have been completely mapped.

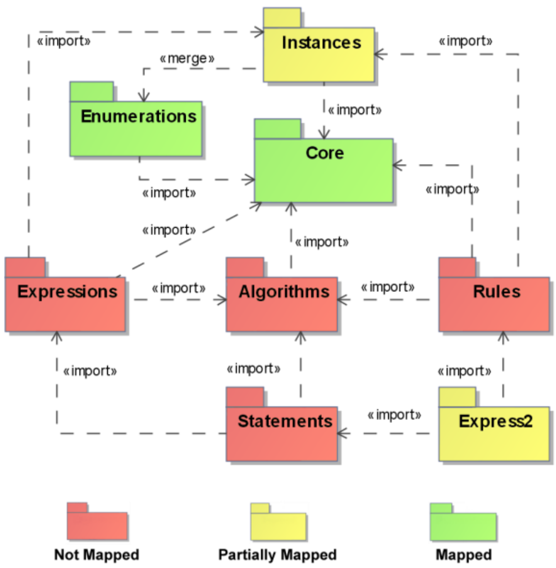


Figure 5.16: Mapping Status of $\theta(EXPRESS, LIMM)$

Finally, the "Instances" package provides EXPRESS, a capacity similar to LIMM and OWL and complies with POC step 4, representing instances of every class. This means that the EXPRESS meta-model comprehends a way to represent not only the Level 1 of MDA (model level) but also the Level 0 (data level), and even supports the possibility of the existence of the two levels at the same time, i.e. an instance of the meta-model is not exclusively composed by instances of the Level 1 – it can be either a mix of the two levels, or just one.

5.2.5.2. *OWL – LIMM Mapping Morphism*

A similar exercise has been performed for OWL. Using the same nomenclature, Table 5.3 presents an extract of the complete mapping for the OWL concepts. However, unlike EXPRESS, OWL *Class* is structured quite differently from LIMM, with many incoming relationships and few embedded properties or outgoing links. This fact explains the reason why, to be able to map all the sub-elements of LIMM's *Entity_Concept*, one has to look in three different OWL elements, namely *Class*, *SubClassOf*, and *ObjectPropertyDomain*. OWL is a language that provides a high degree of freedom on the declaration of the modelling concepts and one could have the element specified in different ways, thus adding a lot of redundancy in the mapping definition.

Table 5.4: OWL' "Class" Mapping to the LIMM (mapping extract)

OWL2 Meta-model (W3C, 2008)	**LIMM**
For each:	
Class ***(ClassExpression as Class)***	**Entity_Concept** ***(Concept as Entity_Concept)***
(*(IRI)* (Class.entityIRI)).value *With: IRI as FullIRI*	Entity_Concept.name
"false"	Entity_Concept.isAbstract
SubClassOf ***(Axiom as ClassAxiom as SubClassOf)***	
(ClassExpression)(SubClassOf.subClassExpression)	*(Entity_Concept)* (Entity_Concept.isSpecificationOf)
ObjectPropertyDomain ***(Axiom as ObjectPropertyAxiom*** ***as ObjectPropertyDomain)***	
(ObjectPropertyExpression)(ObjectPropertyDomain. objectPropertyExpression) *With: ObjectPropertyExpression as ObjectProperty*	*(Property)* (Entity_Concept.contains)

Due to the language specification, which is meant to enable semantic reasoning over models, the full OWL meta-model also contains a number of concepts that are not mappable to LIMM. For example, *Annotations*, restrictions such as *ObjectSomeValuesFrom*, *ObjectHasSelf*, and axioms like *DisjointClasses* or *EquivalentClasses*, are just some examples. Of course by not mapping them, the transformation is withdrawing expressiveness of the model, but only on behavioural components which does not endanger the goals of the package for interoperability establishment.

To conclude this mapping analysis, it is worth noticing that the OWL meta-class *Individual* has a key role to address step 4 of the POC. It is directly related to classes and properties to represent their instances and can be mapped to the instances package of language independent meta-model. When associated to a class or to an aggregation of values, *Individual* is mapped to the *Instance_Group* element, and in the case of association to a property whose type is a basic (e.g. string, integer), the mapping is directly performed with the *Instance_Item* to represent the real value.

5.2.5.3. Transformation: LIMM Instantiation

After a successful XML injection (step 1), an instance of the language meta-model, XMI serialised, can be obtained through a first simple mapping (step 2). Then using the mappings previously defined it is possible to implement the transformation to LIMM with ATL rules enabling it as a service that can be executed on demand anytime needed (step 3). A graphical example of the result of applying the mappings to the geometry module used in the explanation of the POC implementation steps can be found on Figure 4.16 “Example of a Simple Language Independent Model”. Independently of which language was used at the beginning, the same conceptual models will generate the same results at LIMM level (i.e. LIM outputs).

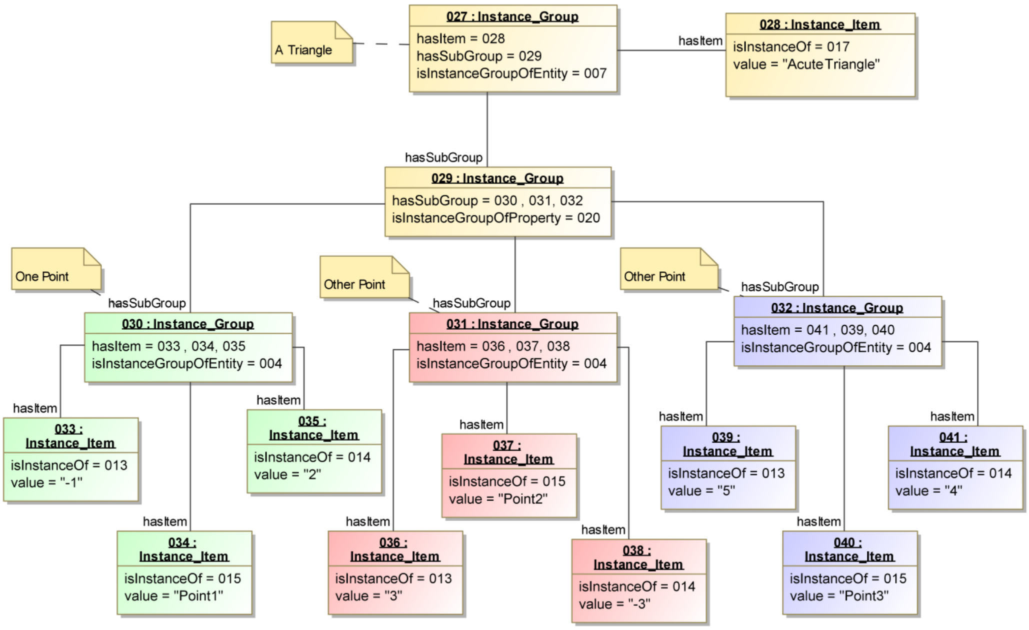


Figure 5.17: Example of Instances Represented on a LIM

The LIM corresponding to the original input model (output of the language transformation $\tau(OWL, LIMM)$ or $\tau(EXPRESS, LIMM)$) is the output intended to the inter-enterprise harmonisation layer (architecture instantiated in Figure 5.10), representing the original model with some loss of behavioural characteristics but translated to the common interface using high abstraction concepts easier for the non-technical experts to follow. Nevertheless, up to step 3, all ATL rules have regarded transformations of the information models only. However, by the illustrated in Figure 5.10, step 4 envisages the integration of the instances, enabling the inter-enterprise harmonization layer to execute mappings among two different organisations smoothly. Since LIMM can represent either models and/or model instances, and the first part does not imply the second (instance package is detachable and optional), it is possible to append those instances onto the XMI serialisation. Figure 5.17, complements the example with the instances of a concrete “Acute Triangle”, represented as an *Instance_Group* which is linked with the triangle *Entity_Concept* (id “007”) previously mapped.

5.2.6. Steps 5, 6 and 7: LIM – LIM Mapping and Transformation

LIM to LIM transformations are enabled and envisaged in these implementation steps. According to MapT's knowledge mapping type (KMType), they can be used to harmonise information model structures, concepts, and even data instances through an implementation of the inter-enterprise harmonisation layer. As explained in the analysis of the requirements of TC1.3, this implementation redounds in a collaborative tool that preserves the technology abstraction but still provides a graphical means to define the different kinds of mapping, while storing them in the communication mediator knowledge bases of the enterprises involved. As mapping definition is a complex and time consuming task, the idea is that it is capable of halting the process at any time without loosing progress.

5.2.6.1. Mapping Tool

JGraph has been elected and modified to read LIM model files and store morphisms at the CM. 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. Naturally, some adjustments had to be made to enable the interaction (mapping definition) among two different graphs, and to become integrated with LIM and CM. A JAVA binder (JAXB[57]) was included to allow the unmarshalling (interpretation) of LIM files, and JENA[58] - a Java API for OWL providing services for model representation, parsing, database persistence, querying - was used for the integration with the communication mediator.

Concerning the visual customisation, Table 5.5 highlights the POC's graphical choices.

Structural Mapping

Step 5 envisages the definition of a structural mapping or versioning among two different LIM data models. According to the package for interoperability establishment specification in section 4.3, mappings should be formalised with a knowledge mapping tuple (MapT) that enables to capture not only the simple relationship, but also the semantic mismatch in case of imperfect mappings and a formal expression (normally mathematical). The mapping tool previously presented enables to do that by providing a mapping editor where the user can view and edit manually the mappings defined graphically. In order to maintain simplicity, the same example will continue to be used to illustrate these mappings. Nonetheless, complex models have been used to validate the POC with data related to the test case descriptions (see industrial validation of next chapter 6 "Implementations Testing and Hypothesis Validation").

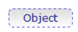
According to Figure 5.18, two organisations (A and B) have very similar, but different representations of triangle geometry. It is possible to see the multiple structural mappings defined, from the higher level ones among the *Models*, or *Modules*, down to more specific ones

---

[57] Java Architecture for XML Binding: http://www.oracle.com/technetwork/articles/javase/index-140168.html
[58] Available at: http://protege.stanford.edu/plugins/owl/jena-integration.html

as “Point” *Entity_Types* or its attribute sub-mappings. Highlighted on the mapping editor is the link being established between the “x” coordinate of A’s point, with “z” of B’s. This mapping is actually disjoint and could be defined with any other element of A. At first it does not make much sense, but since “z” coordinate is mandatory, the mapping needs to exist, otherwise the posterior ATL data transformation among both organisations would be invalid and not interoperable. Also, B’s complex entity “Geometric_Plan” has no structural equivalence in model A. One may consider it to have some degree of conceptual relationship with “Object”, but for the moment it remains unmapped.

Table 5.5: Mapping Tool Graphical Choices

Concept	Graphical Representation	Comments
Model		It is a container
Module		It is a container
Instantiation_Group		It is a container
Entity_concept (not abstract)		
Entity_concept (abstract)		Abstract concepts are only instanced by their subtypes so they are marked differently (mapping should not be defined for these entities) and they internal elements not represented as they will only be used in the sub-entities.
Type_concept		
Generic_Basic_Type		
Instance_Item		Only exists within the Instance_Group container
Specification of Entity_concept		Full, bold line in a solid cylinder marking the parent entity
Single_Valued_Property (not optional)		Full arrow style
Single_Valued_Property (optional)		Dotted arrow style
Multiple_Valued_Property		Cardinality marked within brackets
Mapping		Incorporated mapping editor enables to control all the elements of MapT and write ATL rule

The complete structural mapping that can be defined between both models is presented on the next tables.

Table 5.6: Model MapT Example

ID		A
MElems = (a,b)	a	Triangle_Example
	b	Another_Triangle_Example
KMType		StructuralSemantics
MatchClass		Coverage
Exp		b ∩ a ≠ {}

Table 5.7: Module MapT Example

ID		A.1
MElems = (a,b)	a	Geometry_Module
	b	Geometry
KMType		StructuralSemantics
MatchClass		Coverage
Exp		b ∩ a ≠ {}

Table 5.8: Entity_Concept MaptT Example

ID		A.1.1
MElems = (a,b)	a	Triangle
	b	Triangle
KMType		StructuralSemantics
MatchClass		structuring
Exp		b ~ a

Table 5.9: Multiple_Valued_Property MapT Example

ID		A.1.1.1
MElems = (a,b)	a	Triangle.vertices
	b	Triangle.nodes
KMType		StructuralSemantics
MatchClass		Naming
Exp		b = a

Table 5.10: Entity_Concept MaptT Example

ID		A.1.2
MElems = (a,b)	a	Point
	b	Point
KMType		StructuralSemantics
MatchClass		Coverage
Exp		b ∩ a ≠ {}

Table 5.11: Single_Valued_Property MapT Example

ID		A.1.2.1
MElems = (a,b)	a	Point.x
	b	Point.x
KMType		InstantiableElement
MatchClass		Equal
Exp		b = a

Table 5.12: Single_Valued_Property MapT Example

ID		A.1.2.2
MElems = (a,b)	a	Point.y
	b	Point.y
KMType		InstantiableElement
MatchClass		Equal
Exp		b = a

Table 5.13: Single_Valued_Property MapT Example

ID		A.1.2.miss
MElems = (a,b)	a	Point.x
	b	Point.z
KMType		InstantiableElement
MatchClass		Disjoint
Exp		b = ‘0’

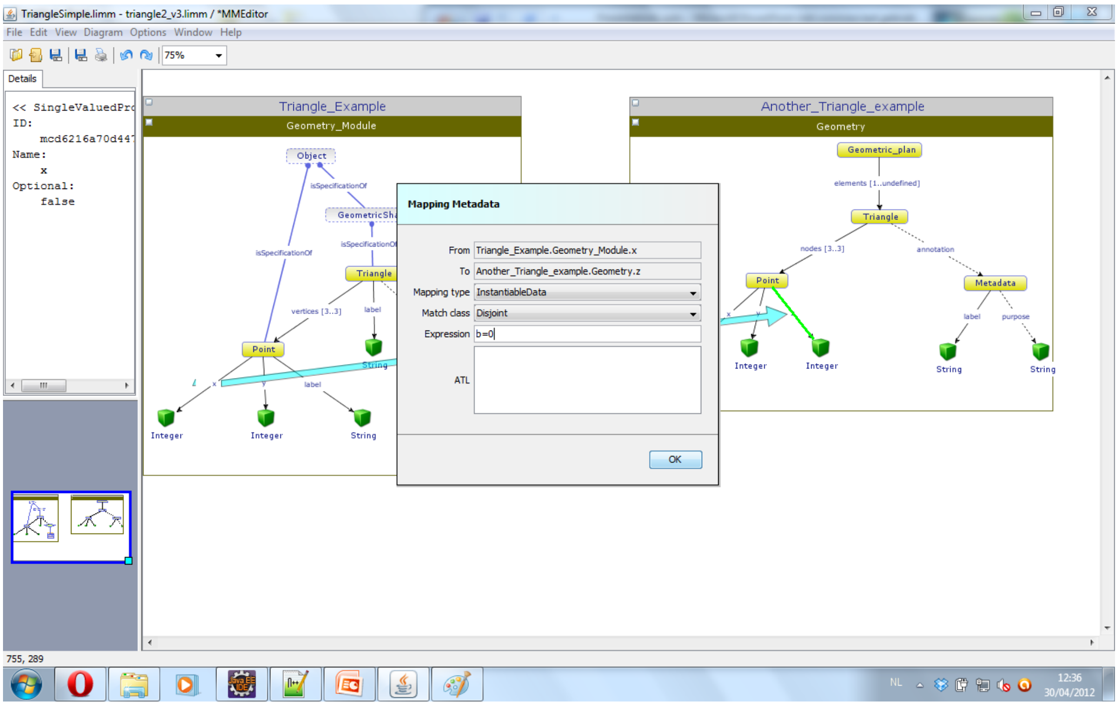


Figure 5.18: Mapping Tool Snapshot

Conceptual Mapping

The conceptual mapping establishment (step 6) is done exactly in the same way. Normally it is used to complement the structural mapping and relate specific concepts with a reference ontology, as envisaged in TC3.2, but it can also be used at a P2P level as well. As illustrated in Figure 5.11, this mapping can (but not necessarily) enable the transformation of terminology at the instances level. For example the conceptual mapping among A's "Object" and B's "Geometric_Plan" just complements the semantic understanding of the full mapping (Table 5.14). As these *ModelElements* are not classified as "ofType" "concept" in the communication mediator (see Figure 4.18), transformations are not reflected in instances, unlike the example of Table 5.15 where reference data concepts are being mapped for a language translation. Using this conceptual morphism, label instances of A can be translated into B at the time of data exchange.

Table 5.14: Conceptual MapT Example for Class

ID		Concept_1
MElems = (a,b)	a	Object
	b	Geometric_Plan
KMType		Conceptual
MatchClass		moreGeneral
Exp		b ⊆ a

Table 5.15: Conceptual MapT Example for Concept Reference Data

ID		Concept_2
MElems = (a,b)	a	acute triangle
	b	triângulo agudo
KMType		Conceptual
MatchClass		language
Exp		b = a

Data Mapping

The last type of mapping exemplified in the POC is the data mapping, which is important for industrials to define the existence of relationships between physical elements. For example in the test case 3.3, two airplane components need to be linked and marked as "physicallyConnected"

since they are to be assembled together. TC4.1 illustrates the revision of one of these components, which through the package for sustaining interoperability, will need to raise an alert at the assembly line.

It is not easy to define a meaningful data mapping for the triangle example since it describes geometric data and not real physical elements. Nevertheless, MapT tuple and the CM remain the mechanisms for representing and storing them.

5.2.6.2. Transformation: Using an Instantiated CM

After a successful mapping definition (step 5 and/or step 6), enterprise A LIM instances can then be transformed to an enterprise B LIM using the combination of the ATL rules specified for each mapping. Being a unidirectional mapping, it is stored on A's CM and can be assessed by internal services whenever this company needs to exchange data with B.

The LIM corresponding to the output of the language transformation ($\tau(OWL, LIMM)$ or $\tau(EXPRESS, LIMM)$) is the input needed to the inter-enterprise harmonisation layer which will then execute the structural ($\tau(LIM_A, LIM_B)$) and conceptual ($\tau(LIM_{ConceptA}, LIM_{ConceptB})$) (if any) transformations on the instantiated elements. To complete the cycle and enable an interoperable data exchange, the instances of LIM_B still need to be reconverted to the expected data representation language, which may be achieved by a simple XML extraction of the LIM XMI serialised file or, in alternative, might need to go through the inverse process of the language harmonization layer.

Figure 5.19 summarises the application of all the transformations to the same geometry example. It simulates the situation described on TC3.4, where data from one company (using OWL) is translated to a collaborative system (LIM). A snapshot of the Protégé tool is inserted in the upper left corner where it is possible to view not only A's data structure ("Object", "GeometricShape", "Triangle", "Point"), but also some property instantiation rules ("vertices" are exactly 3 "Point") and the corresponding instances which one wants to put in the collaborative system. $\tau(OWL, LIMM)$transformation is illustrated with a subset of steps 3 and 4 ATL rules, and the instances part of the result (top right corner) displayed in a UML graphical view (see Figure 5.17 for a larger image). Finally, $\tau(LIM_A, LIM_B)$) and $\tau(LIM_{ConceptA}, LIM_{ConceptB})$ corresponding to the execution of step 5 and 6 are just symbolically represented with a MapT table (detailed view has been presented before from Table 5.6 to Table 5.15) with background ATL rules built from the information in the CM. The end result in LIMM format, already according to the collaborative system data structure, is visible on the mapping tool view[59] (bottom left corner), where one can see the data for the three different points that constitute the triangle with one of the coordinates ("z") set to zero as specified in the mapping rule. The translation of the term "acute triangle" to "triângulo agudo" is also visible.

[59] This tool was not fine tuned to display data instances thus they look a bit awkward. However, since the end result described in this transformation is still at LIMM format, it was decided to present it that way

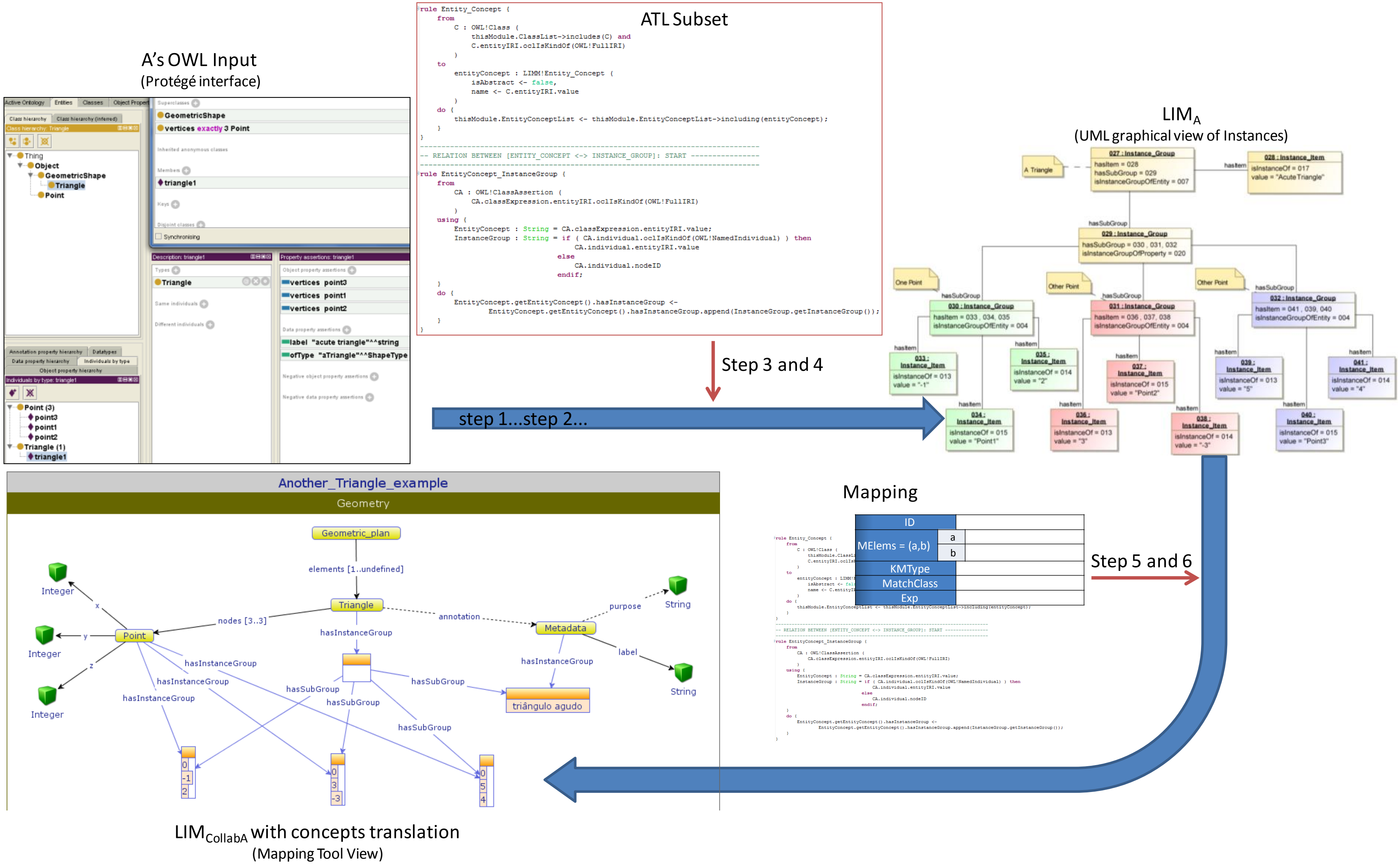


Figure 5.19: $\tau(OWL, LIMM)$, $\tau(LIMA, LIMB)$, $\tau(LIM_{ConceptA}, LIM_{ConceptB})$ Transformation Sequence

In support of the transformation example, Figure 5.20 includes a snapshot of A's instantiated CM where it is possible to see (bottom left corner) that it is instantiated with 3 *InstantiableElement* mappings, highlighting (bottom right corner) the one where "z" is assigned the value "0", 2 "Conceptual" mappings, and 5 "StructuralSemantics", among 2 different *InformationModels*.

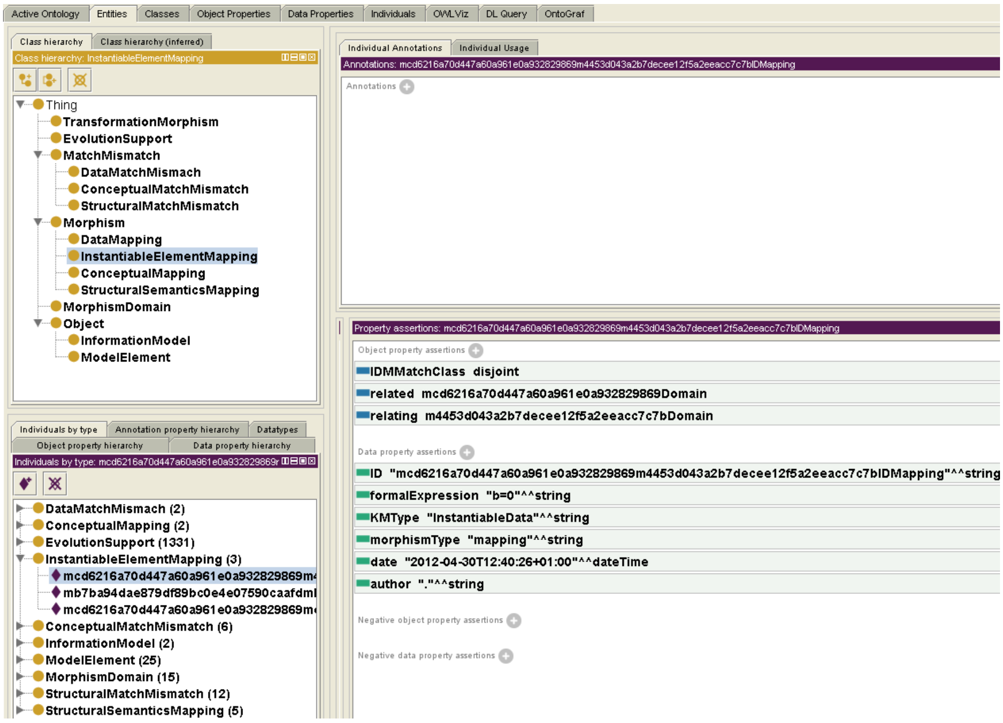


Figure 5.20: A's CM Snapshot

5.3. Implementation of the MAS Package for Sustaining Systems Interoperability

The main objective of this proof-of-concept consists in the implementation of the MAS package for sustaining systems interoperability, most specifically the MIRAI architecture (presented in Figure 4.20) for monitoring and detection of harmonization breaking, as well as the preparation of adequate system readjustment responses. According to the scenarios described, this POC is to be used across 4 TCs, re-enabling interoperability where it has been disrupted and using the morphisms knowledge available on the CM in support of the system's intelligence and self-reorganisation.

By the analysis of the TC requirements, one concludes there is the need to detect 2 harmonization breaking situations, support the user in one readjustment and resume data exchange after decision is taken as envisaged in the CAS-SIF framework.

5.3.1. Technical Instantiation of MIRAI

As explained in previous chapter 4, software agents in the form of MAS were elected to implement the CAS-SIF framework due to their specific characteristics, namely autonomy and flexibility (see analysis of section 3.5.1 "Agent Based Modelling (ABM)"). Several agent platforms are proposed

by the research community, with different specifications and claimed advantages/disadvantages, but JADE[60] was elected for the POC development, given it is widely used (Bellifemine et al., 2001), compliant with the FIPA specifications, and supported by JAVA, the development environment used in the previous package for interoperability establishment.

Since the CAS-SIF framework envisaged in the specification of this package foresees the continuous interaction with each enterprise's communication mediator knowledge base, other used technologies include OWL, JENA and Protégé used already before in the previous package, and also W3C's SPARQL[61], a query language able to retrieve and manipulate stored knowledge.

The next sections explain in more detail the functional behaviour and algorithms behind MIRAI's agents' implementation. The POC is divided in five steps used to describe how the MIRAI works from the point of view of two organisations, i.e. the one that causes the harmonization breaking, and the partner that is affected by the changes (see Figure 5.21):

- The first step concerns the setting up of the multi-agents system on top of the enterprise system;
- Step 2 describes the methodologies and implementation of the activities envisaged for the monitoring process and detection of the morphisms versioning ($\theta(X, X')$);

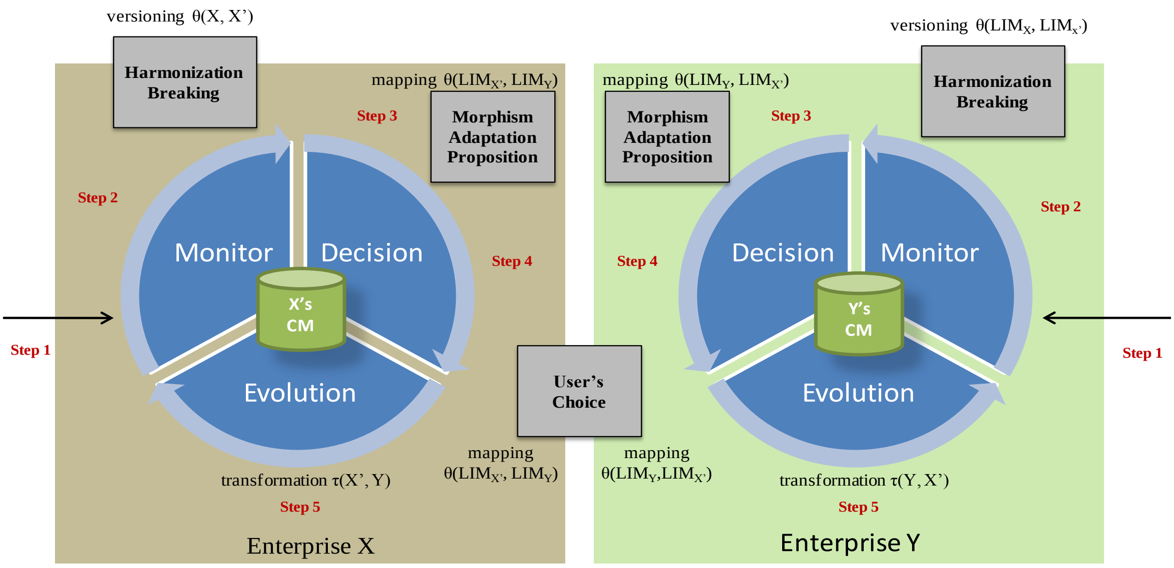


Figure 5.21: MIRAI Implementation Steps Instantiation

- Step 3 is focused on the self-organisation, namely on the implemented algorithm for the generation of new mapping morphisms, responding to harmonization breaking;
- Step 4 illustrates the user's interaction implementations, including the learning algorithm that enables MIRAI to considers the user's choice in a response to a future problem;

[60] JAVA Agent Development Framework: http://jade.tilab.com/
[61] SPARQL Query Language for RDF: http://www.w3.org/TR/rdf-sparql-query/

- And the final step details the communications implemented between the different MIRAIs It is during this step that the enterprise system can re-establish data exchanges and transformations following the newly defined morphisms ($\tau(A,B)$).

This package implementation is responsible for answering requirements of the following test cases:

- TC2.3 "$harmonization\ breaking(B, network)$" – One of the major goals of MIRAI is exactly the detection of harmonization breaking situations. In this case the MAS detects an internal change that puts in risk the communication "B" maintains with the remaining network. This company is embedding a STEP standard directly in its own system data structures, thus major versionings occur and are detected by the implementation of the MIRAI intelligent supervisor block. Special focus is given to the *Agent Monitor Mediator* and the activities of step 2.
- TC2.4 "$decision\ (B, new\ M2M_{BA}\ \ , transient)$" – The transient evaluation package for minimizing network impact would be one of the major contributors to this TC. However, as explained before no dedicated POC implementation were envisaged, thus the activities to tackle this TC are carried mostly by the *Agent MoMo* from the intelligent supervisor block, which is based on the harmonization breaking detection analyses and proposes the most adequate B to A morphisms readaptation (step 3), and also by the MIRAI administrator block to ensure a proper communication with the human decision maker (step 4).
- TC2.5 "$data\ transformation\ \tau(LIM_A,B)$" – Data transformation also is not an activity for the package for sustaining systems interoperability. However, it is dependent on the mapping existing between organisations, and within this context assumes special relevance as it was changed to respond to the harmonization breaking. Moreover, since the transformation is occurring in the direction of A to B, TC2.5 demands an efficient communication among A an B's MIRAI systems, readjusting the mappings also in that direction (step 5 and steps 3 and 4 from the right side perspective of Figure 5.21).
- TC4.2 "$harmonization\ breaking(C, network)$" – This TC is similar to TC2.3, envisaging to test the system's capabilities to detect interoperability harmonization breaking. Nevertheless, and despite the technology applied is the same (MIRAI intelligent supervisor block), it is not so much focused in technical interoperability and automation of the response, but just on the identification of a problem so that users involved become aware some real life products have changed and might not work together as expected.

5.3.1.1. Technology Used

To summarise the most important technologies and tools (re)used throughout the proof-of-concept implementation, a short table is presented on Table 5.16.

5.3.2. Step 1: MIRAI setup

In section 4.5 "MAS Package for Sustaining Systems Interoperability", the CAS-SIF framework and the enabling MIRAI multi-agents architecture have been explained in depth, identifying the major

building blocks and describing the different agents use cases and activities sequence. In total six agents are envisaged to interact as illustrated in Figure 5.22.

Table 5.16: Technology (re)used in the Sustaining Interoperability Package POC

Technology			Tools/Models		
	Purpose	Step		Purpose	Step
Agents	Technology concept that support this POC	1..5	JADE	Framework to support agents development	1-5
FIPA	Protocols to specify the communications between the agents	1..5	Protégé	Ontologies tool for the CM design verification during TCs execution	2..4
OWL	Format for knowledge bases representation - used in the CM	2..4	JENA	OWL API to persist data into the CM	4
SPARQL	query language able to retrieve and manipulate stored knowledge	2, 3			
JAVA	Programming language used in the agents implementation	1..5			

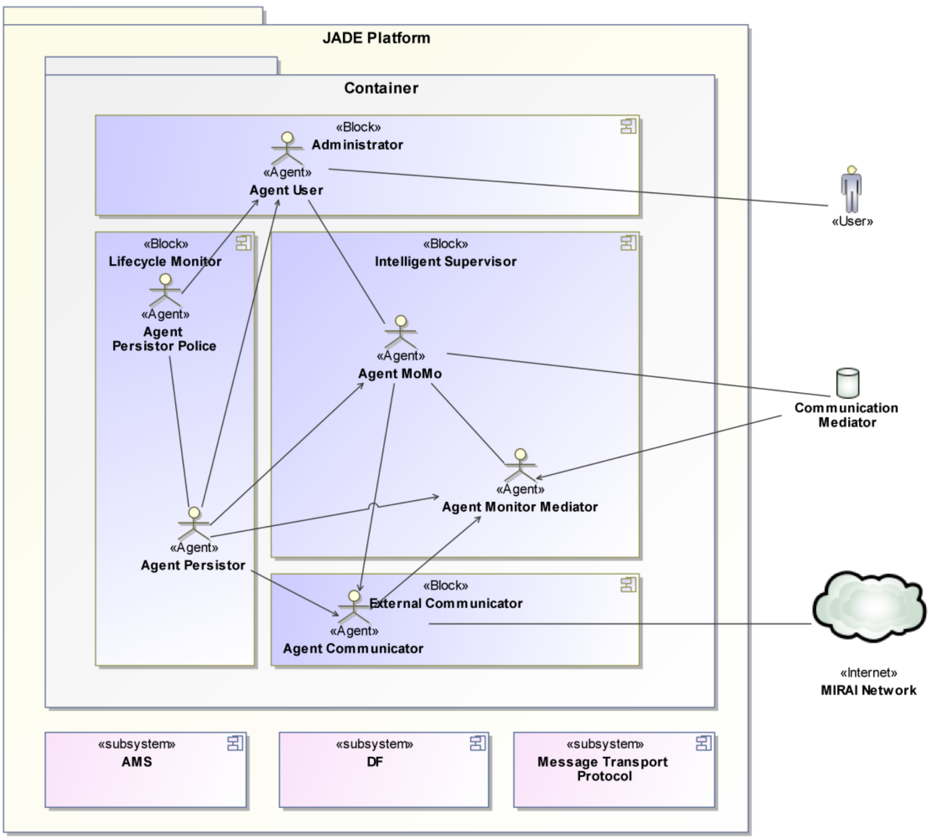


Figure 5.22 - MIRAI Architecture Set-up using JADE

MIRAI has been set-up using JADE, which envisages an environment where agents live on top of a platform that provides them with basic services such as message delivery and special agents sub-systems, namely the “DF” to provide a kind of yellow pages service where agents can publish their service provision, and the “AMS”, an agent able to perform platform management actions (e.g. starting and killing agents on demand of other agents). In JADE, a platform is composed of one or more containers of agents to achieve distribution and agents’ organisation in activity blocks as

proposed. However, as MIRAI intends to be replicated along the enterprise collaboration network, several independent JADE platforms are launched as required and inter-platform communication is envisaged via the “Message Transport Protocol”, which is able to transmit at request of other agents, ACL messages according to the FIPA specifications.

To achieve POC step 1, one needs to launch the JADE runtime, creating a container already including the *Agent Persistor Police*. According to the activity diagram specified in Figure 4.33, it asks the “AMS” to launch the *Agent Persistor* which does exactly the same for the remaining four agents, thus launching MIRAI. While performing these activities, “DF” automatically registers them to become available to the entire platform.

5.3.3. Step 2: Monitoring and Harmonization Breaking Detection Routines

The implementation of step 2 routines are carried majorly on the intelligent supervisor block, namely on *Agent Monitor Mediator* and *Agent MoMo* which are in constant interaction with the internal CM repository checking the file system metadata and using SPARQL queries to access existing morphisms. According to what has been specified in chapter 4 (see Figure 4.23), the monitoring process must occur in two different ways that correspond to the perspectives illustrated above in the MIRAI implementation steps instantiation, i.e. the point of view of the organisation that causes the harmonization breaking (1) and the point of view of the partners affected by it (2):

1. This point of view’s activities begin with the *Agent Monitor Mediator* scanning the file system searching for changes in the CM OWL file. As harmonization breaking does not occur every day, this is done only once a day to not overload the system. Only in case that file has been changed the real harmonization breaking detection algorithm begins, i.e. *Agent Monitor Mediator* queries the CM for new versioning morphisms, and them it passes the control of the actions to *Agent MoMo* that analyses the existing mappings morphims (with the business partners) and the new versionings to check if interoperability is at risk (see example of activity in Figure 5.23). Independently of the result of the analysis, the implementation of the administrator block considers an informative message to the user, reporting what has happened.

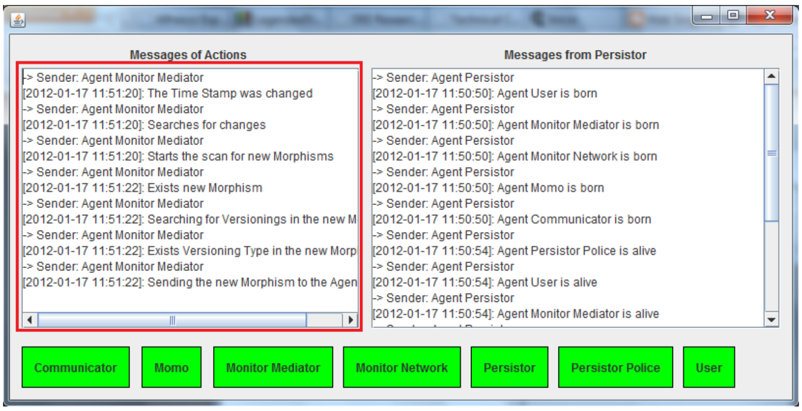


Figure 5.23: Agent Monitor Mediator Working

2. From the point of view (2), activities actually begin with the interaction between *Agent External Communicator* with *Agent Monitor Mediator*. Whenever information about an

harmonization breaking originated on a partner organisation is received, the *Monitor Mediator* triggers *Agent MoMo* to carry a similar process as before. The major difference is that the relevant versionings are passed together with the alert message.

5.3.4. Step 3: Algorithm for the Generation of new Mapping Morphisms

The earlier step 2 explains how MIRAI, namely the intelligent supervisor block, detects changes within the morphisms of a network of companies. Here the focus is to explain how the *Agent MoMo* generates new morphism(s) to respond to harmonization breaking situations caused by versioning morphism(s). Consequently, the algorithm for the generation of new mapping morphisms conducts the following course of actions to generate a new MapT:

1. Analyse detected versioning;
2. Analyse existing mapping;
3. Propose new mapping.

Some parameters are easier to calculate, such as ID (automatic), KMType (the same as the existing mapping), or the mapping elements (MElems), obtained from the CM's *ModelElements* linked to both *Morphism* objects, but others are dependent on more complex calculations based on the two existing morphisms (mapping and versioning). Figure 5.24 shows an abstraction of a situation where MIRAI 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 ($g = M_1 \rightarrow M_1'$) is detected (versioning). The figure also represents the existing mapped relation (f) between the two models (M_1 and M_2), f^{-1} for the opposite situation, and the respective model elements that are targeted by the algorithm (x from M_1, y from M_2, and z from M_1').

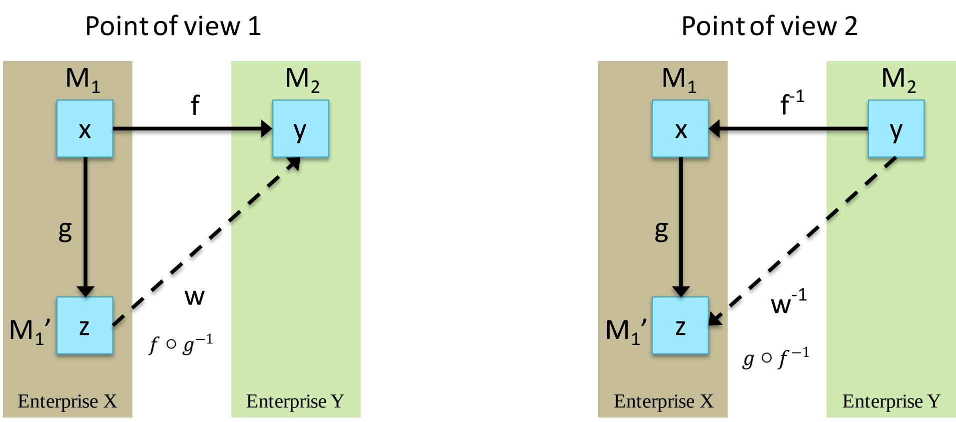


Figure 5.24: Generation of New Mapping Morphisms

Figure 5.24 formalises a transformation in a generalised way, where for every x and y that are elements of Model, f is the tuple (belonging to MapT domain) that enables the transformation morphism τ. Applying mathematics transitivity principle of binary relations, it is possible to calculate the remaining constituents of MapT, namely Exp and also the ATL which is not in the tuple but is very important for the transformation execution: for all x, y and z, if xRy and yRz exist, then xRz also applies. Equation 34 claims that having a relation from the element x to the element y (designated by the morphism f) and another from element x to element z (designated by the

morphism 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 by the morphism w), composed of g^{-1} and f.

$$\forall\, x, y \in Model\, and\, f \in MapT, if\, \exists\, f(x,y) then \exists\, \tau : M \longrightarrow M, and\, \tau(x,f) = y \qquad (33)$$

$$\forall\, x, y, z \in Model\, and\, w \in MapT, if\; \exists\, \tau(x,f) = y\, and\, \tau(x,g) = z, \\ then\, w = f \circ g^{-1}\, and\, \tau(z,w) = y \qquad (34)$$

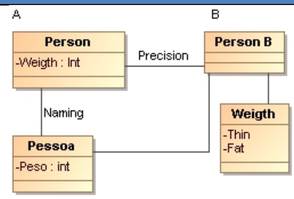
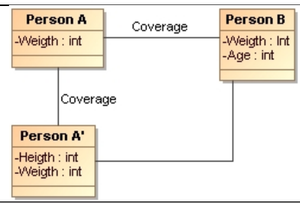
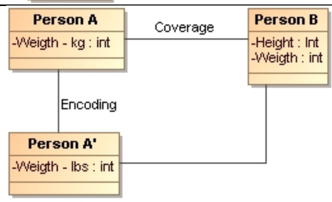
$$\forall\, x, y, z \in Model\, and\, w \in MapT, if\; \exists\, \tau(y,f^{-1}) = x\, and\, \tau(x,g) = z, \\ then\, w^{-1} = g \circ f^{-1}\, and\, \tau(y,w^{-1}) = z \qquad (35)$$

However, this automatism has some restrictions which could probably be improved if the conceptual mappings would be used in the algorithm as well, namely:

- The composed ATL and Exp used in the new morphism is valid only if $\tau(x,g^{-1})$ is a lossless transformation, otherwise data would be lost by this functional composition.
- For the cases the versioned elements are decomposable (not of KMType = "InstantiableElement"), the new ATL rules must gather the ones corresponding to the sub-elements while Exp is always defined by the user.

Finally, the MapT MachClass suggestion follows a weighted pattern matching strategy that learns with the user's choices as explained in detail on step 3. An initial pattern matching table (see Table 5.17) has been defined and loaded at system start, but its weight values change over time.

Table 5.17: Subset of the Initial Pattern Matching Table for MatchClass Recalculation

Versioning	Mapping	New Mapping	Weight	Example
Naming	Precision	Precision	1	
Coverage	Coverage	Coverage	1	
Encoding	Coverage	Coverage + Encoding	1	
Any Match Mismatch	Equal	Match Mismatch (Versioning)	1	-

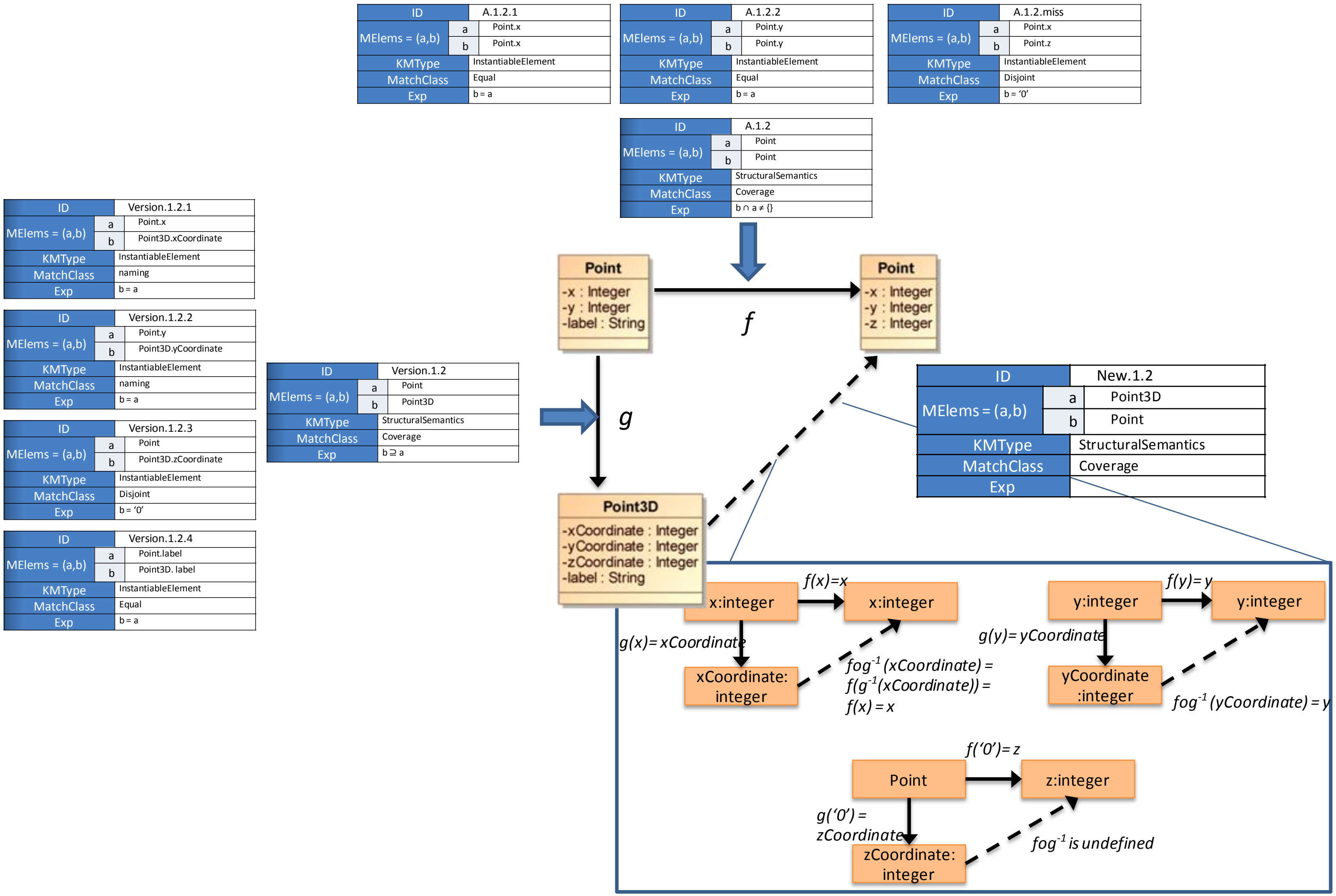


Figure 5.25: Example of New Mapping Creation

To exemplify the algorithm, and continuing with the geometry example used before, let's consider that the "Triangle_Example" model has evolved and is now adopting 3D coordinates (using new names as well) just like the "Another_Triangle Example" of its business partner. With this change, most of the model elements remain the same and equation 34 is perfectly adjusted. The specific case for the "Point" *Entity_Concept* is illustrated on Figure 5.25. On the top part are the existing mappings between enterprise X and enterprise Y. On the left are the versioning's registered, where it is possible to see that two of the attributes have a naming mismatch, and on the bottom right one can see the POC results following the algorithm specified. The new MatchClass is of "Coverage" type as indicated in the patterns table and the Exp is left empty as it is a composed element. The $f \circ g^{-1}$ results for the "InstantiableElements" are applicable for two of the 3 sub-elements.

5.3.5. Step 4: Decision Making and Learning

The implementation of step 4 routines are carried majorly on the intelligent supervisor and administrator blocks, namely on *Agent MoMo* and *Agent User* which communicate with the objective of proposing adaptation solutions to the user, collect the results and assimilate the decisions for future reference. Following the activity diagrams specified in Figure 4.23 and Figure 4.27, in the scope of decision making, *Agent User* is activated as soon as a new mapping is proposed, and communicated it to the human user through an interactive interface. On Figure 5.26 is possible to see a snapshot of a very simple application developed to present information to the user. In this case, the new mapping (choice) is already pre-instantiated by the actions of *Agent MoMo*, and the interface enables the user to accept or refuse it asking, then, the new choice. Since users may not have an immediate decisions, they can postpone their actions by simple closing the application, or mark it as concluded ("Done").

If implemented, this would be the step to integrate the package for minimizing positive feedback loops.

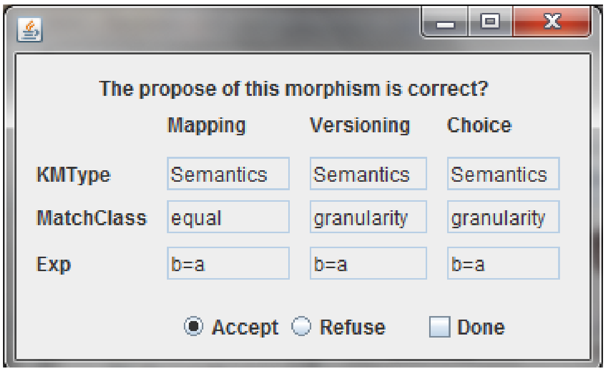


Figure 5.26: Proposal of a New Morphism Interface

After the human decision maker submits is choice of mapping, then the communication will be reverse and the *Agent User* passes that information to the *Agent MoMo* that learns with the choices using a weighting algorithm By the example used before, with the "Point" evolution, the MatchClass suggestion turns out to be appropriate, however, if label did not exist on "Point3D", then a better one would be selected by the user as the two entities would be experiencing only a "naming"

mismatch. To help that this choice is taken in consideration for future reference, in the CM was included a table (*EvolutionSupport*) where all the MatchClass patterns are introduced:

1. The best result for all the possible choices is and set to 1 (initial ideal patterns) while the rest are set to 0.
2. Then, every time the *Agent MoMo* has to propose a new mapping, it queries this knowledge base and chooses the result with the higher weight for the queried pattern.
3. When the user accepts the answer the *Agent MoMo* will increment the weight by 1.

As this was implemented to prove the concept that user's choices could influence *Agent MoMo's* proposals, the decision method is not optimised and the weighting proportion is probably not the best. Nevertheless better algorithms could be applied.

Finally, this step is responsible for storing the new morphisms in the CM so that data transformations, through the ATL rules, can be executed as illustrated previously in Figure 5.19.

5.3.6. Step 5: Communication between MIRAI's

This section approaches the communication between different enterprises, following the specification of section 4.5.3 "External Communicator Block" and explains the implementation of *Agent Communicator*, which is responsible to forward harmonization breaking notifications to the network and vice-versa. As introduced in step 1 (section 5.3.1.1), JADE already provides some built-in functionalities that enables communication among different JADE platforms, i.e. the message transport protocol, which in theory should assure the interoperability of different platforms. Nevertheless, during the POC implementation that was not successful and for time constraints the communication between MIRAI's was simulated manually providing specific input messages to the different *Agent Communicator*.

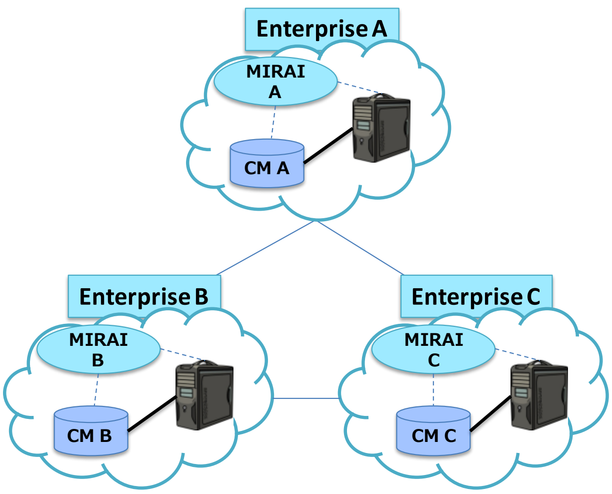


Figure 5.27: Communication between MIRAI's

.

6. Implementations Testing and Hypothesis Validation

Research is the driving force of modern society. However, a research result is not a contribution to the field if not known or nobody can use it due to lack of proper validation (Camarinha-Matos, 2009). *When properly applied throughout the lifecycle of developments, verification and validation lead to high quality results. Indeed, to a great extent the quality of research determines the future and a society that aims to play a leading role needs not only to invest on research programs but also to carefully monitor progress and assess the impacts of the various research initiatives.*

This chapter presents the validation methodology implemented along this research project lifecycle, and illustrates some of the major metrics to evaluate and discuss the initial hypothesis.

6.1. Validation Components

To evaluate the several research developments and proof-of-concept implementations detailed along the previous chapters, a component-based validation strategy has been defined considering the actual software testing executed upon the POC implementation, the validation through peers acceptance, and industrial validation.

Figure 6.1, gathers all three, and illustrates a parallel execution of all components with special emphasis on the acceptance by peers and implementations testing. The first becomes more relevant towards the final stages of thesis validation while the second has predominance in the initial stages where the POC development started to be defined and implemented. Also industrial validation and acceptance plays a preponderant role, ideally more towards the research conclusions stages so that in becomes adopted, but also in earlier stages where the ideas are

discussed and their relevance accessed. Each of these components results is described in detail in the following sections.

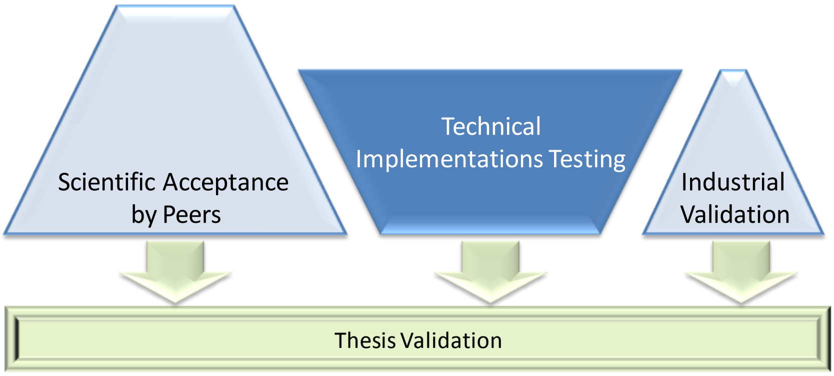


Figure 6.1: Validation Components

6.2. Technical Implementations Testing

Testing and validation is the process of trying to find errors in a system implementation by means of experimentation. This experimentation is usually carried out in a special environment, where normal and exceptional use is simulated. The aim is to gain confidence that during normal use the system will work satisfactory, since testing real systems can never be exhaustive because they can only be tested during a restricted period of time. On the other hand, testing cannot ensure complete correctness of an implementation since it can only show the presence of errors, not their absence (Tretmans, 2001).

In the next sections some methodologies will be presented to best approach the test definition applied to the particular proof-of-concept implementation of this dissertation. Although a successful testing applied to the POC does not mean that it is ready to work as a commercial software (since it was never intended to be one), it can validate that all major functions and modules are correctly (or not) working and with that validate the feasibility of a future full implementation.

6.2.1. Analysis of Existing Testing Methodologies

There are many testing methodologies available to test software engineering, many of them are abstract concepts like white / black / grey box testing, unit testing, conformance testing, etc., where functional and structural testing are distinguished from each other focusing more either on the internal code or on the overall functionality (Myers, 1979; L. White, 1987). Proved methods to apply these concepts are defined in many standards and revised throughout the years based on the expertise of using them and their practical results (e.g. ISO/IEC SQuaRE[62], ISO/IEC 9646[63], etc).

6.2.1.1. SQuaRE-based iSurf Functional and Non-Functional Evaluation Methodology

iSurf was a research project integrated in the EU Seventh Framework Programme, which main objective was the development of *"an environment to facilitate the collaborative exploitation of*

---

[62] ISO/IEC 25000 - Software product Quality Requirements and Evaluation (ISO/IEC, 2007)
[63] Open Systems Interconnection -- Conformance testing methodology and framework (ISO/IEC, 2001b)

distributed intelligence of multiple trading partners in order to better plan and fulfil the customer demand in the supply chain" (I-Surf STREP, 2010). However, besides providing a knowledge-oriented inter-enterprise collaboration environment for SMEs to share information on the supply chain, iSurf invested in the development of an evaluation and testing framework, following and complementing the standard process envisaged by the reference model ISO/IEC CD 25040 (ISO/IEC, 2007) of the SQuaRE series of standards.

SQuaRE details the activities and tasks providing their purposes, outcomes and complementary information that can be used to guide a software product quality evaluation. The iSurf project describes in detail the procedures used to generate the evaluation criteria to be applied for the functional and non-functional characteristics of intermediate and final products, namely functionality, reliability, usability, efficiency, maintainability and portability. These techniques and their evaluation criteria were modularised as recommended in ISO/IEC 25041 former ISO/IEC 14598-6 (ISO/IEC, 2001a) specifying the evaluation methods applicable for quality measurement (functional / non-functional). The modules are documented as specified on the standard ISO/IEC 14598-6:

1. Provide formal information about the evaluation module and gives an introduction to the evaluation technique described in the evaluation module;
2. Defines the scope of applicability of the evaluation module;
3. Specifies the input products required for the evaluation and defines the data to be collected and measures to be calculated;
4. Contains information about how to interpret measurement results.

Functional and non-functional evaluation modules provide a flexible and structured approach to define criteria for monitoring the quality of intermediate products (e.g. prototypes and POCs) during the development process and for evaluation of final products. These modules define a set of structured instructions and data used for an evaluation, as well as the format for reporting the measurements, with the purpose of ensuring that software evaluations can be repeatable, reproducible and objective. The aim is to make the various aspects (principles, metrics, activities, etc.) of evaluation visible and to show how they are handled. Also, the iSurf evaluation modules define the criteria for the evaluation of components considering the quality characteristics specified on SQuaRE, namely *functional*:

- Functionality: Functional Test Cases;
- Functionality: Unit Tests;

And *non-functional*:

- Reliability: Fault tolerance Analysis;
- Usability: User interface;
- Efficiency: Execution time measurement;
- Maintainability: Inspection of development documentation;

- Portability: Analysis of software installation procedures.

6.2.1.2. Tree and Tabular Combined Notation (TTCN) – Test Notation Standard

The Tree and Tabular Combined Notation is a notation standardised by the ISO/IEC 9646-1 for the specification of tests for communicating systems and has been developed within the framework of standardised conformance testing (TTCN-3, 2011). Based on the black-box testing model, the tests are specified through tables which are divided in general description, constraints, behaviour and verdict. With TTCN, the test behaviour is defined by a sequence of events which represent the test *per se*. This sequence can be approached as a tree, containing branches of actions based on the evaluation of the system output after one (or a series of) executed event that can be of one of two types:

- *Actions*, which are preceded by an exclamation point before its brief description, and represent actions performed on the System Under Test (SUT);
- *Questions* that are preceded by an interrogation point, and represent evaluations of the output of the SUT after one or more actions are completed.

Since the answer can be positive or negative, multiple questions can exist at the same indentation level, covering all possible outputs of the system. After a completion of a TTCN test table, a verdict must be deliberate, either "Success", "Failure" or "Inconclusive" based on the sequence of events and the outputs of the system. Table 6.1 presents a simplified example describing an evaluation description for a phone call. After a series of actions and evaluations (questions events) a different verdict is attained. The table can textually be read as:

1. The user picks up the headphone;
2. Tests if the dialling tone is present;
3. If the dialling tone is present, then the user must dial the a phone number. Otherwise, if the dialling tone is absent, the verdict is a "Failure" of the possibility of establishing a phone call;
4. If there is a calling tone after dialling the number, the user may test if the line is in fact connected;
5. If the line is connected, the user may hung up the headphone and the verdict is set as "Success" on establishing a phone call, otherwise the verdict is a "Failure";
6. If the dialling tone is not heard, but a busy tone instead, then the user may hung up the headphone and the verdict is set as "Inconclusive" on establishing a phone call;
7. If none of the tones corresponds to calling or busy, then the verdict is set as "Failure" on establishing a phone call.

Some more examples and tutorials are available at the TTCN-3 website (TTCN-3, 2011).

Table 6.1: Simplified example of a TTCN table test

Test Case		
Test Group:	Basic Connection	
Purpose:	Check if a phone call can be established	
Comment		
Behaviour	**Constraints**	**Verdict**
! Pick up headphone		
? Dialling tone		
! Dial number		
? Calling tone		
? Connected line		
! Hung up headphone		Success
OTHERWISE		Failure
? Busy tone		
! Hung up headphone		Inconclusive
OTHERWISE		Failure
? Dialling tone absent		Failure

6.2.2. Adopted Test Methodology

Chapter 5 of this dissertation addresses the implementation of chapter's 4 frameworks and architectures specification, as a proof-of-concept. Nevertheless, despite the importance of validation and testing already explained earlier, unlike a commercial product, the POC is not supposed to be flawless, robust and complete solution. Instead it seeks to be a working proof of feasibility for the solution and/or idea. In fact, the POC itself has been developed with the concerns of addressing a predefined set of industrial scenarios and test cases (see section 5.1 "Application Scenarios and Test-Cases for a Self Sustainable Interoperability"). Hence, applying such a complex test methodology as the one specified on the iSurf project that targets the full development lifecycle of the product from the POC stages to prototyping and commercialization does not make sense. It is too much time consuming and is not really an added research value for this PhD dissertation.

Thus, a mix of validation tests has been chosen. Based on the iSurf test methodology and the TTCN tables proposed by ISO/IEC 9646, a series of functional test cases and unit tests were designed and applied to the various predefined TCs previously identified in Table 5.1. On the other hand, non-functional tests envisaged in iSurf, such as reliability, efficiency and portability were also addressed. All the results and tests definitions are reported in the next sections 6.2.3 "POC Functional Testing" and 6.2.4 "POC Non-Functional Testing".

6.2.3. POC Functional Testing

To address the functional testing of the proof-of-concept implementation, a TTCN table was designed for each industrial TC where execution was implied, covering one or more implementation steps. The usage and effectiveness of external tools is not included in the functional testing. TC1.1 and TC2.1 are not testable for functionality as they describe modelling

language mapping definitions, which are activities executed manually. These mappings will be tested implicitly in the test cases that need to execute the transformations. Nevertheless, unlike those, the other type of mappings definition, as envisaged in TC1.3, TC2.2, TC3.1, TC3.2, TC3.3, and TC4.1, are testable since they are executed through a prototype tool and envisage interaction with the CM knowledge base.

Having this, the first TTCN test case defined (Table 6.2) considers the transformation evaluation of steps 2 and 3 of the MDA-based package for interoperability establishment. Step 1 uses a pre-defined Eclipse ATL's execution engine function that injects a XML serialization of the OWL input model into the system. It is possible to test whenever an invalid model is put at input of the step 2 transformation, since Eclipse ATL's execution engine automatically determines if a model is valid against the corresponding meta-model. After a successfully imported XML model, the transformation itself is applied to the injected model and a manual inspection of the results (OWL meta-model instance) must be done in order to validate its correctness accordingly with the original model. A similar manual inspection on the LIM results must be conducted to evaluate the success of step 3 transformation, as well. It is important not only to evaluate if the loss of expressiveness (which can be substantial if the rules are not correctly implemented, since these transformations are translating models to different output modelling languages) is not higher than expected, but also to evaluate that each concept is in fact represented in the desired output as the intended mapping originally defined.

The second TTCN test case defined (Table 6.3) continues with the output of the previous and is meant to evaluate if the implementation of the mapping establishment is following the specifications of MapT and according step 5 of the MDA-based package for interoperability establishment. The initial tests consist on checking if the prototype is able to load and display both models required for the mapping (LIM). After successfully loaded, a manual inspection of the displayed graphs is required to ensure if everything is conforming to the output of the previous step 3 (all model elements should be displayed graphically according to Table 5.5. Finally, the test specification envisages testing the functional behaviour in the definition of the mappings: first the graphical relationships, then the specification of the mapping metadata according to MapT, and finally the storage on the CM so that it can be continued or executed at any time. Again in this last question verification, manual inspection on the physical knowledge base file (output) is required.

Table 6.4 completes the definition of TTCN test cases of scenario 1, enabling to test the actual data exchange from one enterprise ("A") to another ("D"). This test considers successful any transformation that translates input data into the destination's LIM in a meaningful and coherent way. The transformation cycle from LIM to the actual format used in the destination organisation will be target of evaluation in another test case. Nevertheless, previous events are described and need to be tested to ensure that step 5 transformation of LIM_A to LIM_D is valid, namely loading a data file compliant with the information model of "A" and appending the result of step 4 transformation (passing through the similar tests as the ones of Table 6.2) to an existing LIM_A. As before, manual inspection is recommended to ensure the coherency checking of some results.

Table 6.2: TTCN Specification of TC1.2 - OWL Language Transformation

Test Case:	Language transformation τ(OWL,LIMM)		
Group:	Scenario 1		
Purpose:	Check if the OWL language transformation specified, correctly transform all mapped concepts		
Comments:	MDA-based Package for Interoperability Establishment – steps 2 and 3		
Behaviour		**Constraints**	**Verdict**
! Load a OWL/XML model			
? Valid XML model and not empty			
! Apply XML injection to XML meta-model			
! Apply step 2 transformation (to OWL meta-model)			
? Output is coherent with input model			
! Apply step 3 transformation (to LIMM)			
? Output is coherent with input model			Success
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Inconclusive

Table 6.3: TTCN Specification of TC1.3 and TC3.1 - LIM Model Mapping

Test Case:	Model mapping θ(LIM_A, LIM_D) and Model mapping θ(LIM_D, LIM_{Collab})		
Group:	Scenario 1 and scenario 3		
Purpose:	Check if the mapping tool correctly imports LIM files and enables to define a model mapping among all supported concepts		
Comments:	MDA-based Package for Interoperability Establishment – step 5		
Behaviour		**Constraints**	**Verdict**
! Load a LIM model (enterprise A or enterprise D)			
? Valid LIM and not empty			
! Load another LIM model (enterprise D or collaborative environment)			
? Valid LIM and not empty			
? Displayed model graphs are coherent with the input			
! Define graphical relationships among the different model elements			
? Able to view and edit mapping metadata (MapT information)			
! Store progress in the CM			
? CM coherent with mappings defined			Success
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Inconclusive
OTHERWISE			Inconclusive

Table 6.4: TTCN Specification of TC1.4 - Data Transformation (Enterprise - LIM)

Test Case:	Data transformation $\tau(A,LIM_D)$
Group:	Scenario 1
Purpose:	Check if the LIM transformation specified, correctly transform all mapped concepts
Comments:	MDA-based Package for Interoperability Establishment – steps 4 and 5

Behaviour	Constraints	Verdict
! Load a LIM model (enterprise A)		
? Valid LIM and not empty		
! Load XML data (enterprise A)	Compliant with A's model	
! Apply step 4 language transformation appending results to LIM (enterprise A)		
? Transformation was successful		
? Output is coherent with input data		
! Load a LIM model (enterprise D)		
? Valid LIM and not empty		
! Load mappings (LIM A -> LIM D)	Stored in A's CM	
! Apply step 5 transformation		
? Output is coherent with input data		Success
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive

Table 6.5: TTCN Specification of EXPRESS Language Transformation

Test Case:	Language transformation τ(EXPRESS,LIMM)
Group:	Scenario 2
Purpose:	Check if the EXPRESS language transformation specified, correctly transform all mapped concepts
Comments:	MDA-based Package for Interoperability Establishment – steps 2 and 3

Behaviour	Constraints	Verdict
! Load a EXPRESS model	XML serialised	
? Valid XML model and not empty		
! Apply XML injection to XML meta-model		
! Apply step 2 transformation (to EXPRESS meta-model)		
? Output is coherent with input model		
! Apply step 3 transformation (to LIMM)		
? Output is coherent with input model		Success
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive

Table 6.6: TTCN Specification of TC2.2 - LIM Model Versioning

Test Case:	versioning mapping θ(LIM_B, LIM_{AP236})
Group:	Scenario 2
Purpose:	Check if the mapping tool correctly imports LIM files and enables to define a model versioning among all supported concepts
Comments:	MDA-based Package for Interoperability Establishment – step 5

Behaviour	Constraints	Verdict
! Load a LIM model (enterprise B)		
? Valid LIM and not empty		
! Load another LIM model (AP236)		
? Valid LIM and not empty		
? Displayed model graphs are coherent with the input		
! Define graphical relationships among the different model elements		
? Able to view and edit mapping metadata (MapT information)		
! Store progress in the CM		
? CM coherent with mappings defined		Success
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

Table 6.7: TTCN Specification of TC2.3 and TC4.2 - Detection to Internally Caused Harmonization Breaking

Test Case:	Harmonization breaking(B,network) and Harmonization breaking(C,network)
Group:	Scenario 2 and scenario 4
Purpose:	Check if MIRAI detects an internally provoked network harmonization breaking situation
Comments:	MAS Package for Sustaining Systems Interoperability– step 2

Behaviour	Constraints	Verdict
! Request for activities log (enterprise B or enterprise C)	MIRAI is active	
? Log displayed and readable		
? All agents are alive		
! Apply versioning morphism (enterprise B or enterprise C)		
? Agent monitor mediator detected changes in the CM time stamp (CM_B or CM_C)		
? Agent monitor mediator detected new versioning's		
! Request for activities log		
? Log displayed and readable		
? Versioning's registered are coherent with CM		Success
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

Table 6.8: TTCN Specification of TC2.3 and TC4.2 - Reaction to Internally Caused Harmonization Breaking

Test Case:	Reaction(B) or Reaction(C)
Group:	Scenario 2 and Scenario 4
Purpose:	Check if MIRAI reacts to an internally provoked network harmonization breaking situation
Comments:	MAS Package for Sustaining Systems Interoperability– step 3

Behaviour	Constraints	Verdict
! Request for activities log (enterprise B or enterprise C)	MIRAI is active	
? Log displayed and readable		
? All agents are alive and harmonization breaking was detected		
? Agent MoMo detected interoperability problems		
? Adaptations are presented to the user (B -> A or C -> X, for X= B or D)		Success
OTHERWISE		Failure
OTHERWISE		Failure (scenario 2 or 4)
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

Table 6.9: TTCN Specification of TC2.4 - Decision and Learning

Test Case:	decision (B,new $M2M_{BA}$)		
Group:	Scenario 2		
Purpose:	Check if MIRAI accepts and learns from user's corrections		
Comments:	MAS Package for Sustaining Systems Interoperability– step 4		
Behaviour		**Constraints**	**Verdict**
! Request for activities log (enterprise B)		MIRAI is active	
? Log displayed and readable			
? All agents are alive and adaptation(s) are pending user's approval			
! Reject and provide correction(s) to the suggested morphisms (B -> A)			
? CM was updated			
? CM coherent with mappings defined and MatchClass weights			Success
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Inconclusive
OTHERWISE			Inconclusive

Table 6.10: TTCN Specification of TC2.3 and TC4.2 - Reaction to Externally Caused Harmonization Breaking

Test Case:	Reaction(X) while X is a partner enterprise of the one causing the harmonization breaking		
Group:	Scenario 2 and Scenario 4		
Purpose:	Check if MIRAI detects and reacts to an externally provoked network harmonization breaking situation		
Comments:	MAS Package for Sustaining Systems Interoperability– step 3		
Behaviour		**Constraints**	**Verdict**
! Request for activities log (enterprise X)		MIRAI is active	
? Log displayed and readable			
? All agents are alive			
? Agent monitor mediator detected external harmonization breaking			
! Request for activities log			
? Log displayed and readable			
? Versioning's received (B -> X or C -> X) are coherent with X's elements			
? Agent MoMo detected interoperability problems			
? Adaptations (A -> X or C->X) are presented to the user			Success
OTHERWISE			Failure
OTHERWISE			Failure (scenario 2 or 4)
OTHERWISE			Failure
OTHERWISE			Inconclusive
OTHERWISE			Failure (scenario 2 or 4)
OTHERWISE			Inconclusive
OTHERWISE			Inconclusive

As the test case previously defined to evaluate the translation of OWL models into LIMM, the TTCN test case specification of Table 6.5 also defines a language transformation. All the testing events (actions and questions) are similar, but here the envisaged modelling language is EXPRESS (ISO TC184/SC4, 2004). The POC implementation envisages a step 0 where EEP, a well known EXPRESS model validator, serialises EXPRESS models in XML. However, since this is an ISO recognised software and there is no interest whatsoever to further test its outputs, the test case already considers a XML serialised model input. The test case of Table 6.6 follows the exact same procedure as the one of Table 6.3 since the versioning is a specific case of mapping, i.e. instead of relating models of two different organisations, it is meant to link models of the same one.

TC2.3 intended to simulate the detection and reaction of the network to an harmonization breaking caused by the implementation of the standard AP236, for furniture product data representation, directly in enterprise “B” information system. Due to its long range of activities, it was split in 3 sub test-cases (Table 6.7, Table 6.8 and Table 6.10) to evaluate separately the different agents' intelligence. Moreover, TC2.4 was defined in between them (Table 6.9) because they are closely interlinked in terms of functionality and would be easier to test activities in the same workflow as they are specified in the corresponding activity diagrams (see section 4.5). Also, concerning the evaluation events specified in each of these 4 TTCN test tables, most are of type “questions”, since the SUT is a multi-agents system which is permanently executing and needs, or accepts, little actions from the user's side. Nevertheless, to achieve a successful test in all four cases, one needs to make sure MIRAI is executing properly with all agents alive, and that logs being displayed on request are readable and track the flow of actions registered.

In short, for the detection of an internally provoked harmonization breaking (Table 6.7), after inducing new versioning morphisms in the CM, the system must detect them, and a manual inspection will provide assurances of success or failure. The second part of TC2.3 evaluates if *Agent MoMo* is working as expected, and after being informed of that harmonization breaking, proposes a set of adaptations to re-establish interoperability with “B” partner, i.e. “A”. Then (Table 6.9), the user should reject that proposal, submit a different morphism and, this way, test the system's learning functionality. Again, to be sure everything is running as expected a manual inspection on the communication mediator of B is required to check if all activities have been properly registered and the user action weighted. A similar line of actions is expected in case an external harmonization breaking is detected (Table 6.10).

The final TTCN test case definition for scenario 2 is depicted in Table 6.11. Intending to illustrate data exchange among two organisations, this test can be complementary to TC1.4 since it evaluates the actions needed to transform data from the LIMM format into a specific organisation, in this case XML (compliant with EXPRESS). Before executing the real transformation (step 5 transformation of LIM_A to LIM_B) it is required to load both LIM models and the mapping stored in the CM. If the output of the data transformation is coherent, then the inverse path of transformations (from LIM to XML) needs to be followed. As usual, input data loading is validated by the Eclipse

Table 6.11: TTCN Specification of TC2.5 - Data Transformation (LIM - Enterprise)

Test Case:	Data transformation τ(LIM_A,B)	
Group:	Scenario 2	
Purpose:	Check if the updated LIM transformations, correctly transform all mapped concepts	
Comments:	MAS Package for Sustaining Systems Interoperability – steps 5, and inverse steps 4, 2, 1 MDA-based Package for Interoperability Establishment – step 5	

Behaviour	Constraints	Verdict
! Load a LIM model (enterprise A)	Instantiated with data	
? Valid LIM and not empty		
! Load another LIM model (enterprise B)		
? Valid LIM and not empty		
! Load mappings (LIM A -> LIM B)	Stored in A's CM	
! Apply step 5 transformation		
? Output is coherent with input data		
!Apply inverse step 4 transformation (to EXPRESS)		
? Output is coherent with input data		
! Apply inverse step 2 transformation (to XML)		
! Apply XML extraction		
? Output coherent with input data		Success
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

ATL’s executor engine against the existing meta-models (first EXPRESS and then XML) and the outputs should be verified manually. The last action is the data extracting into a XML tagged file.

Application scenario number three is composed by a majority of test cases illustrating the different kinds of mapping morphisms, from model to conceptual and data. Since the type of activity is the same, their execution steps are all very similar, loading and validating the two LIM files required to define a mapping in the implemented prototype, defining the graphical relationships and then editing the tuple metadata that will be stored in the CM. In fact the TTCN definition for TC3.1 (model mapping) is described together with TC1.3 in Table 6.3, as the only difference is the input data. The conceptual mapping of TC3.2, corresponding to step 6 of the MDA-based package for interoperability establishment, is described in Table 6.12. Since the ontology harmonization process is very dynamic, this specific type of mapping would probably require a manual insertion of concepts, but that is not contemplated in the developed version of the mapping tool prototype, thus cannot be tested. Finally, the data mapping test case is available in Table 6.13, and the major difference for the previous is the fact that the input LIM models need to be instantiated with data. All test cases require manual inspection at several intermediate tests but, most importantly, on the physical knowledge base file (output).

Table 6.14 illustrates the execution events for test case TC3.4, which envisages data transformation with terminology translation using the conceptual mappings previously defined. Being a data transformation envisaged in the MDA-based package for interoperability establishment, this TTCN table is similar to the one presented for TC2.5 and even more with TC1.4 since this one stops the simulation at LIM level. It starts by loading both LIM models (input and output) and, when successfully loaded, ATL transformations can be applied using the mappings duly stored in the communication mediator of enterprise “A”. First is applied step 6 transformation where the data format remains the same and the terminology used is translated to the one of the reference ontology. After inspecting the result, data transformation according to step 5 is applied to become available in the collaborative environment used in scenario 3. The test case is considered to be successful if the output of these transformations cycle is coherent with the input data.

Scenario 4 is, as described in section 5.1 composed of TC4.1 and TC4.2. The first TC, described in the TTCN Table 6.15, is a data versioning. Very similar to many tests of mapping already presented, namely the ones of Table 6.13 – data mapping, and Table 6.6 – model versioning, it needs as input two instantiated LIM models, where the second LIM needs, in this case, to be a versioning of the first. After successfully loading these LIM’s, the mapping tests follow the same flow and the instance groups can be related as evolutions of one another, and stored in the CM so that the MIRAI can detect and respond to the harmonization breaking (TC4.2). The TTCN test descriptions corresponding to TC4.2 are divided in three and described together with the TC2.3 (Table 6.7, Table 6.8, and Table 6.10) which also simulated another harmonization breaking situation where detection and response are relevant. In this case (TC4.2), the response is less meaningful but is still important to notify the user about a potential problematic situation and prevent the propagation of a problem.

Table 6.12: TTCN Specification of TC3.2 - LIM Conceptual Mapping

Test Case:	conceptual mapping $\theta(LIM_{D\ ONTOLOGY}, LIM_{Collab\ ONTOLOGY})$
Group:	Scenario 3
Purpose:	Check if the mapping tool correctly imports LIM files and enables to define a conceptual mapping among all concepts
Comments:	MDA-based Package for Interoperability Establishment – steps 6

Behaviour	Constraints	Verdict
! Load a LIM model (enterprise D)		
? Valid LIM and not empty		
! Load another LIM model (Ontology)		
? Valid LIM and not empty		
? Displayed model graphs are coherent with the input		
! Define graphical relationships among the different concepts	Manual insertion of concepts would be useful but is not contemplated	
? Able to view and edit mapping metadata (MapT information)		
! Store progress in the CM		
? CM coherent with mappings defined		Success
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

Table 6.13: TTCN Specification of TC3.3 - LIM Data Mapping

Test Case: Data mapping θ(LIM_D, LIM_C) Group: Scenario 3 Purpose: Check if the mapping tool correctly imports LIM files and enables to define a data mapping among all instances Comments: MDA-based Package for Interoperability Establishment – steps 5		
Behaviour	**Constraints**	**Verdict**
! Load a LIM model (enterprise D)	Instantiated with data	
? Valid LIM and not empty		
! Load another LIM model (enterprise C)	Instantiated with data	
? Valid LIM and not empty		
? Displayed model graphs are coherent with the input		
! Define graphical relationships among the different instances		
? Able to view and edit mapping metadata (MapT information)		
! Store progress in the CM		
? CM coherent with mappings defined		Success
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

Table 6.14: TTCN Specification of TC3.4 - Data Transformation with Terminology Translation (Enterprise - LIM)

Test Case:	Data transformation $\tau(D, LIM_{CollabD})$		
Group:	Scenario 3		
Purpose:	Check if the LIM transformation specified, correctly transform all mapped concepts		
Comments:	MDA-based Package for Interoperability Establishment – steps 5 and 6		

Behaviour	Constraints	Verdict
! Load a LIM model (enterprise A)	Instantiated with data	
? Valid LIM and not empty		
! Load a LIM model (enterprise Collaborative Environment)		
? Valid LIM and not empty		
! Load conceptual mappings (LIM D_ONTOLOGY -> LIM Collab_ONTOLOGY)	Stored in A's CM	
! Apply step 6 transformation		
? Output is coherent with input data		
! Load model mappings (LIM A -> LIM Collab)	Stored in A's CM	
! Apply step 5 transformation		
? Output is coherent with input data		Success
OTHERWISE		Failure
OTHERWISE		Failure
OTHERWISE		Inconclusive
OTHERWISE		Inconclusive

Table 6.15: TTCN Specification of TC4.1 – LIM Data Versioning

Test Case:	Data versioning mapping θ(LIM_C,LIM_C')		
Group:	Scenario 4		
Purpose:	Check if the mapping tool correctly imports LIM files and enables to define a data versioning among all instances		
Comments:	MDA-based Package for Interoperability Establishment – steps 5		
Behaviour		**Constraints**	**Verdict**
! Load a LIM model (enterprise C)		Instantiated with data	
? Valid LIM and not empty			
! Load another LIM model (enterprise C)		Instantiated with new data	
? Valid LIM and not empty		New version of the first	
? Displayed model graphs are coherent with the input			
! Define graphical relationships among the different instances			
? Able to view and edit mapping metadata (MapT information)			
! Store progress in the CM			
? CM coherent with mappings defined			Success
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Inconclusive
OTHERWISE			Inconclusive

Table 6.16: TTCN Specification to Evaluate MIRAI Robustness

Test Case:	Robustness		
Group:	Isolated test		
Purpose:	Check if the persisting agents are able to detect and "resurrect" dead agents		
Comments:	MAS Package for Sustaining Systems Interoperability		
Behaviour		**Constraints**	**Verdict**
! Launch MIRAI, activating agent persistor police			
? Detects dead agents			
? Agent persistor activated?			
? Detects dead agents			
? Other MIRAI agents activated			
! Kill agent			
? Agent is resurrected			
OTHERWISE			Success
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Failure
OTHERWISE			Failure

To conclude the POC functional testing specification, Table 6.16 details a robustness test to evaluate the lifecycle monitor block of the multi-agents system implemented within the package for sustaining systems interoperability. This test launches *Agent Persistor Police* which, according to the activities specification of Figure 4.33, is capable of detecting whether *Agent Persistor* is alive, and launching it if necessary. In turn, this agent does the same for all the other agents of the MIRAI platform. To finalise the test, one can kill any agent at choice and if that agent is resurrected, the test is successful.

6.2.3.1. Critical Analysis of Results

Most of the tests specified for the functional testing have been performed using data from the industrial validation described on section 6.4 “Industrial Acceptance”. Nevertheless, when that was not possible, simulation with examples has been conducted.

Analysing the output of each of them, each TTCN test case has been evaluated as a success. Hence, the implementations described in chapter 5 are indeed following the specifications of chapter 4 and the envisaged application scenarios. The different types of mappings defined proved to be capable of specifying transformations using the ATL language and, their storage in a communication mediator knowledge base, help in the reconstruction of the morphims during the interoperability harmonization breaking response.

The usage of the language independent meta-model requires more transformations than if the mappings were specified directly at the enterprise model, however, that does not imply significant loss of efficiency and is compensated by the generalisation of modelling concepts used by the mapping tool prototype. Since this collaborative tool is integrated with the CMs involved in the process, all the metadata used later by the MIRAI POC is controlled by the user and does not require extra effort than the one required in any other mapping performed outside the framework proposed. In fact, it is the opposite due to the generalised concepts of LIMM.

Concerning the two modelling languages applied, EXPRESS and OWL, all intricate relations and most important model representation forms are preserved, but analysing the inherent loss of expressiveness, a few remarks can be enumerated. For EXPRESS:

- A concept defined initially as being of “AggregationType”, can have the semantic connotation of being “SETType”, “LISTType”, “BAGType” and “ARRAYType”. In LIMM, they are exclusively mapped to *Aggregation_Type*;
- Another loss of expressiveness occurs regarding abstract properties, since LIMM is only able to deal with abstraction of *Entity_Concept* concepts;
- The “USE” and “REFERENCE” links to foreign schemas are mapped to LIMM as *Module* references, hence, its semantic value is also lost.

Concerning OWL, the losses are considerably larger as it is originally a semantic language and as explained before, LIMM is only concerned with the information model itself. Thus, not explicitly enumerating here all concepts, knowledge such as class equivalences, annotations, intersection

rules, transitivity, etc. are lost in the translation. Nevertheless, that can be compensated when defining the conceptual mappings for reference ontologies.

Having the mappings correctly defined, MIRAI has proven its value answering to the requirements specified, i.e. being able to detect automatically any interoperability harmonization situation and by using environment knowledge, the system can self-propose a form of reorganisation. Nevertheless, automation proved sometimes to be complicated which indicates that, despite proving the self-sustainable interoperability concept proposed in this dissertation, implementations could benefit from more intelligence. Detection of harmonization breaking is yet dependant on a tool, i.e. if the model versioning is not performed on the mapping tool, it cannot be detected. Plausible choices are presented but automation in the ATL reconstruction still needs further work. And, in the case where versioning morphisms enlarge the scope addressed by an organisation information model, new mappings could be proposed to relate concepts not yet associated.

Moreover, some technical limitations have been found in the multi-agents technology adopted. In fact, as reported, during the POC implementation JADE’s message transport protocol proved to be of difficult adoption and the communication among different agent platforms (different MIRAI) had to be simulated manually. For instance, the transition among tests reported in Table 6.9 where a decision is taken and implemented, and the ones of Table 6.10, where these adaptations are disseminated through the network, is not automated as envisioned.

However it is believed these issues do not question the validity of the concepts proposed. They just emphasize that further technological developments are still required to fully support a self-sustainable interoperability.

6.2.4. POC Non-Functional Testing

Regarding non-functional testing, reliability, efficiency and portability can be addressed in order to give a glimpse of what can be evaluated.

Reliability is the ability of software to perform a required function under given conditions for a given time interval. In general, software that is not executing what it is intended to is unreliable (I-Surf Partners, 2009). Analysing the reliability of a system is not a simple task, yet in what regards to the MDA-based package for interoperability establishment, it either supports the mapping definition and applies the transformation or it does not. The same paradigm applies for the MAS package for sustaining interoperability. Or it detects an harmonization breaking and tries to resolve it proposing intelligent options to the user or not. Given that the tests executed proved their purpose and, except for the problems already reported, both can be considered reliable.

So far, no critical “bug” (i.e. a problem that aborts or halts a transformation) was detected in the transformations workflow which, considering that the mapping tool is just a small prototype and ATL is currently in growing development on the open source community, could be expected to demonstrate some problems. No crashes have been registered when defining and executing mappings or transformations, and to test reliability furthermore, the full STEP AP236 information model (which is very large declaring more than 700 concepts) was put to the test successfully,

importing it to a LIM and exporting to an XML serialised file. MIRAI's reliability tests have not been so exhaustive because the time required to fully map such large models would prove to be unbearable for this PhD work, nevertheless with the exception of the problem reported with the communication among JADE platforms, the proof-of-concept implementations worked smoothly around the test case examples (either industrial or simulated).

Regarding the efficiency of the implementation, it can be evaluated from the measurement of the average time consumption scenario, though the framework for self-sustaining interoperability is not expected to be time critical, i.e. is not necessary to react to real-time events on-the-fly. Mapping establishment is a time consuming activity where the majority of time is spent on human activities, while harmonization breaking is not something that occurs on a daily basis, thus detection on the following day is considered reasonable (*Agent Monitor Mediator* scans the CM once a day).

Table 6.17: MDA-based Package for Interoperability Establishment Average Time Consumption

<table>
<tr><th>MDA-Package for Interoperability Establishment</th><th colspan="3">Small size information models (< 20 concepts)</th><th colspan="2">Very large information models (> 500 concepts)</th></tr>
<tr><th>Language transformation</th><th>Forward</th><th colspan="2">Reverse</th><th>Forward</th><th>Reverse</th></tr>
<tr><td>Transformation to language meta-model XMI format (or back to XML)</td><td>0,065 sec.</td><td colspan="2">0,025 sec.</td><td>15,581 sec.</td><td>0,498 sec.</td></tr>
<tr><td>Transformation to LIM (or back to language meta-model)</td><td>0,021 sec.</td><td colspan="2">0,023 sec.</td><td>0,939 sec.</td><td>1,892 sec.</td></tr>
<tr><td>Total Process (inc. external tools as EEP or the XML injection)</td><td>0,449 sec.</td><td colspan="2">0,323 sec.</td><td>18,506 sec.</td><td>3,08 sec.</td></tr>
<tr><th>Mapping / Transformation</th><th>Loading</th><th colspan="2">Saving[64]</th><td></td><td></td></tr>
<tr><td>Data/Model loading from LIM</td><td>0,733 sec.</td><td colspan="2">-</td><td></td><td></td></tr>
<tr><td>Mapping Definition</td><td>Manual</td><td colspan="2">-</td><td></td><td></td></tr>
<tr><td>Mappings loading</td><td>0,879 sec.</td><td>8,139 sec.</td><td>4,444 sec.</td><td></td><td></td></tr>
<tr><td>Transformation among LIMs</td><td colspan="3">0,016 sec.</td><td></td><td></td></tr>
</table>

Nonetheless some metrics have been taken and presented on Table 6.17 and Table 6.18, where, as observed, time measurements are well within the accepted parameters with the longest operation taking about 15 seconds to transform larger information models to the XMI conforming to the language meta-model (Table 6.17), or 12-13 seconds for the *Agent MoMo* to propose the various morphism adaptations (Table 6.18). The MIRAI system takes no more than 1 second to set-up and similar amount of time to resurrect dead agents, with the exception of the case where all die and it cannot recover. One fact that has been noted is that even though MIRAI agents are restored quite quickly after death (about 1 second), the others continue to work, which may cause the loss of some messages. It is a very rare event but in future implementations that situation

[64] Saving is slower when one hasn't already saved the file before in the same application run, therefore the saving of mappings was tested in 2 separate ways

needs to be carefully addressed, adding more redundancy to the MAS. Also, as explained earlier, the external communicator block could not be evaluated due to implementation problems.

Table 6.18: MAS-Package for Sustaining Interoperability Average Time Consumption

MAS-Package for Sustaining Interoperability (times are measured through the log, thus have latency)				
Lifecycle Monitor		**1 Dead**	**3 Dead**	**6 Dead**
MIRAI step-up	< 1sec.	-		
MIRAI robustness		< 1sec.	< 1sec.	Doesn't recover
Intelligent Supervisor and Administrator		**1 versioned element**	**10 versioned elements**	**20 versioned elements**
Harmonization breaking detection		2 sec.	2 sec.	2 sec.
MoMo proposition		13,66 sec.	12,33 sec.	12,66 sec.
Decision		Manual	Manual	Manual
External Communicator				
Networks of 2 enterprises		unable to measure	unable to measure	unable to measure
Networks of 3 enterprises				
Networks of 4 enterprises				

Finally, the last non-functional test regards portability. It is defined as the degree of independence a software product, POC, or prototype has from any particular hardware and/or operating system platform (I-Surf Partners, 2009). This proof-of-concept implementation depends on a number of JAVA developed applications and the EEP application for the initial step of the EXPRESS language transformations. Nevertheless, these applications can be compiled and made available for Windows, Linux and Mac OS X, as is the case of EEP. Although it was not implemented, Eclipse's ATL executor engine can be detached from the Eclipse installation by isolating the correct libraries, which means that it is possible to run, in all of the above operating systems.

6.3. Acceptance by Scientific Community

Peer review of the research developed, both at final and intermediate stages is an important metric to assess the quality of the work. As envisaged in the last step of the adopted research method (see section 1.2.1), several scientific publications in recognised[65] conferences and scientific journals, are envisaged during a PhD work. In fact, research is to be disseminated at national and international level through publications in relevant scientific journals (with SCI[66] impact factor) and presentations in international conferences, workshops and seminars with recognised CPSI[67]. Also, international and national RTD projects are valid sources of case studies and validation, absorbing some of the results.

---

[65] Thomson Reuters (formerly ISI) Web of Knowledge (isiknowledge.com/)
[66] SCI- Science Citation Index (isiknowledge.com/).
[67] CPCI - Conference Proceedings Citation Index (isiknowledge.com/).

During this research, the author of this dissertation has joined a number of international discussion groups, research projects and communities where the accumulated knowledge contributed for common advances. Besides, he has been and will continue to be deeply involved in the activities of the host institution[68]. In detail, the most important milestones achieved with inputs from this work are:

- 4 publications (two of them are already published and two are accepted for publishing) in scientific journals addressing the self-sustainable interoperability concept, namely the CAS-SIF framework, the IPyM model, or more specific contributions with propositions to enhance interoperability (Jardim-Goncalves et al., 2009; Jardim-Goncalves, Agostinho & Steiger-Garcao, 2012; Jardim-Goncalves, Grilo, et al., 2012; Jardim-Goncalves, Agostinho, Sarraipa, et al., 2012);
- 13 other scientific publications (1 poster, 1 book chapter and 12 conference papers), including 1 poster with the baseline for the PhD, its evolution to paper format on EI2N'2009 workshop of the On The Move Conferences, developments at the packages levels, namely the MDA-based package for interoperability establishment and the MAS package for sustaining systems interoperability, and other specific contributions (Sarraipa, Agostinho, et al., 2009; Agostinho & Jardim-Goncalves, 2009; Sarraipa, Vieira, et al., 2009; Jardim-Goncalves et al., 2010; Agostinho et al., 2010; Agostinho, Sarraipa, et al., 2011; Agostinho, Jardim-Goncalves, et al., 2011; Lampathaki et al., 2011; Beca et al., 2011; Jardim-Goncalves et al., 2011; Ferreira et al., 2011; Ferreira et al., 2012);

6.3.1. Collaboration in International Research

The most common form of achieving peer acceptance is via scientific publications as explained above. However, the participation in international discussion groups and research projects is also a form of validation. This way, this PhD research is contributing for (thus being validated in) four international research projects (detailed in the following sections), enabling industrial validation and a large cooperation with other research teams. Furthermore, the author has been actively involved in:

- InteropVlab[69] Task Force 2 "Model-Driven Interoperability/ Service-Oriented Architecture" towards keeping the state-of-the-art in the areas of MDI and SOA updated, and providing a basis for implementations and training. Contributions to the task force are focused on the proof-of-concept implementations and new challenges identification.
- Interop-Vlab Task Force 3 "EI Practice and Reference Applications" aiming at helping industry understanding the benefits of EI via the illustration of methods and tools towards EI. Contributions to the task force are focused on the new methods and tools towards EI.

---

[68] Group for Research in Interoperability of Systems (GRIS) of the CTS – UNINOVA (www.uninova.pt/cts/)
[69] Interop-Vlab - The International Virtual Laboratory for Enterprise Interoperability (interop-vlab.eu)

- The PESSOA[70] bilateral initiative with CRAN-UHP[71], promoting active research cooperation among the author's institution and CRAN-UHP. Contributions include the development of co-authored publications.
- The funStep community, an interest group that aims at achieving interoperability among software solutions independently of the place they are used within the company and/or in different companies along the furniture supply chain (FunStep, 1998). Contributions include the development of domain oriented architectures and frameworks, as well as the implementation of proof-of-concepts.

6.3.1.1. CRESCENDO

"Collaborative and Robust Engineering using Simulation Capability Enabling Next Design Optimisation" (CRESCENDO) is an EU co-funded research and technology project launched in May 2009 (CRESCENDO IP, 2009). Led by Airbus[72], it brings together 59 organisations from 13 different countries, including major aeronautics industry companies, service and IT solution providers, research centres and academic institutions, with the ambition to make a step change in the way that modelling and simulation activities are carried out, by multi-disciplinary teams working as part of a collaborative enterprise, in order to develop new aeronautical products in a more cost and time efficient manner.

This collaborative project is already in its final year and specific contributions from this PhD included the introduction of the self-sustainable interoperability framework, and usage of some of the package components (e.g. the LIMM and the CM) to the development of the model-based interoperability architecture and services for the collaborative framework introduced by the project, i.e. the Behavioural Digital Aircraft (BDA) (CRESCENDO Partners, 2010). As described on the industrial acceptance section 6.4.1, CRESCENDO also provided an industrial test environment for some of the TCs addressed along the dissertation, namely the ones of scenario 3 and 4 (see section 5.1.2 "Collaborative Product Development in System Engineering Networks").

6.3.1.2. ENSEMBLE

The ENSEMBLE (Envisioning, Supporting and Promoting Future Internet Enterprise Systems Research through Scientific Collaboration) collaboration project, funded by the EC since May 2010, aims to coordinate and promote research activities in the domain of FInES and to systematically establish EI as a science. ENSEMBLE combines systemic approaches, scientific multi-disciplinarity and innovative Web 2.0 collaboration tools with a community-driven mentality, in order to significantly increase the impact of the Future Internet Systems (ENSEMBLE CSA, 2011).

The project is not industry-driven so it doesn't fall under the category of industrial acceptance. Nevertheless it is validating many of the work and concepts proposed, by incorporating them in the

---

[70] PESSOA program (alfa.fct.mctes.pt/apoios/cooptrans/pessoa/)
[71] Centre de Recherche en Automatique de Nancy (CRAN) of l'Université Henri Poincaré, Nancy 1 (www.cran.uhp-nancy.fr/)
[72] Airbus (www.airbus.com/)

EI science base (EISB) problem and solution spaces formulation, in particular the theory that EI problems and solutions are related to neighbouring scientific disciplines, and that these could be specified and formalised by incorporating methods from their areas of research (motivator to main research question of section 1.2.3). This PhD is contributing with the identification and analysis of the complexity science state-of-the-art in regards to EI, as well as the specification of the EI complexity analyser tool of the initial EISB toolset, based on the ideas the transient evaluation package for minimizing positive feedback loops (section 4.6) (ENSEMBLE Partners, 2012).

6.3.1.3. MSEE

Manufacturing Service Ecosystem (MSEE) is an ICT integrated project, also funded by the 7th Framework Programme of the European Commission. By merging concrete needs and short-term requirements from European manufacturing industry with a long-term vision of Future Internet (FI) enterprises and enterprise systems, MSEE aims to create new Virtual Factory Industrial Models, where service orientation and collaborative innovation will support a new renaissance of Europe in the global manufacturing context (MSEE IP, 2011). Having started on October 2011, this project is just in the early stages and the author's involvement is contributing to the definition of a model based service engineering framework to support end-to-end integrated ICT solutions that enable innovation and higher management efficiency in networked enterprise operations (MSEE Partners, 2012). These contributions follow the same philosophy used on the MDA-based packages for interoperability establishment, and include the use of a communication mediator and the knowledge enriched mapping tuple (MapT) (section 4.4) to represent the mapping of service model specifications.

6.3.1.4. IMAGINE

IMAGINE (Innovative End-to-end Management of Dynamic Manufacturing Networks) is a research and development project, funded by the European Commission under the "Virtual Factories and Enterprises" theme of the 7th Framework Programme. The project targets the development and delivery of a novel comprehensive methodology and the respective platform for effective end-to-end management of dynamic manufacturing networks in an innovative plug and produce approach, and aims at supporting the emergence of a powerful new production model, based on community, collaboration, self-organisation and openness rather than on hierarchy and centralised control (IMAGINE IP, 2011).

Effective end-to-end management of dynamic manufacturing networks is consistently advertised as a top priority for manufacturing enterprises that need to strive to improve their efficiency, adaptability and sustainability of their production systems. Moreover, it is a crucial prerequisite for achieving the envisaged goal. Thus, contributing to project's novel comprehensive methodology for the management of dynamic manufacturing networks is the CAS-SIF framework and the MIRAI distributed network. However, as in MSEE, IMAGINE is yet in the early stages, thus validation is at the level of the discussions and propositions, rather than real implementations as is the case of CRESCENDO. For this reason these last two projects are not included under the industrial acceptance section, next.

6.4. Industrial Acceptance

Industrial acceptance and validation is as important as the acceptance by peers. If industry does not approve the results, most likely they will never be used. This way, since the beginning, this research has been scoped along two industry-led initiatives from two different sectors. Each of them has its own prototyping scenarios, which have been adapted to the application scenarios that scope the POC implementations, as described in the earlier chapter 5.

6.4.1. CRESCENDO's Behavioural Digital Aircraft

In this initiative, research is validated in a large enterprise-ruled sector where the usage of product data standards is already a reality but the diversity of requirements demand more complex products developed with shorter lead times and more cost effectiveness, while using business models involving multiple partners for several activities, e.g. applying novel technologies to aircraft parts, answering to airliner requirements, collaborating with suppliers world-wide, etc.

Therefore, the aeronautical industry is today transforming its operations from a design process based approach to a modelling and simulation based approach in the extended enterprise. An early involvement of production and support teams in down stream design activities, using the digital aircraft, is being stimulated. However, the current approach of modelling and simulation is multi-domain and dispersed in as many "islands" as disciplines involved in the aircraft development programme (structure, thermal and manufacturing engineering, aerodynamics, control, etc.). Each of them develops its own models/simulations when the need arises instead of sharing them among teams and/or organisations. Either due to lack of cooperation or to technical interoperability barriers, this situation leads to redundant work where models have to be rebuilt when needed and prevents effective change management among disciplines (see Figure 6.2). CRESCENDO proposes to solve this situation with the Behavioural Digital Aircraft (BDA), managing the evolution of the aircraft behavioural dataset from concept to certification.

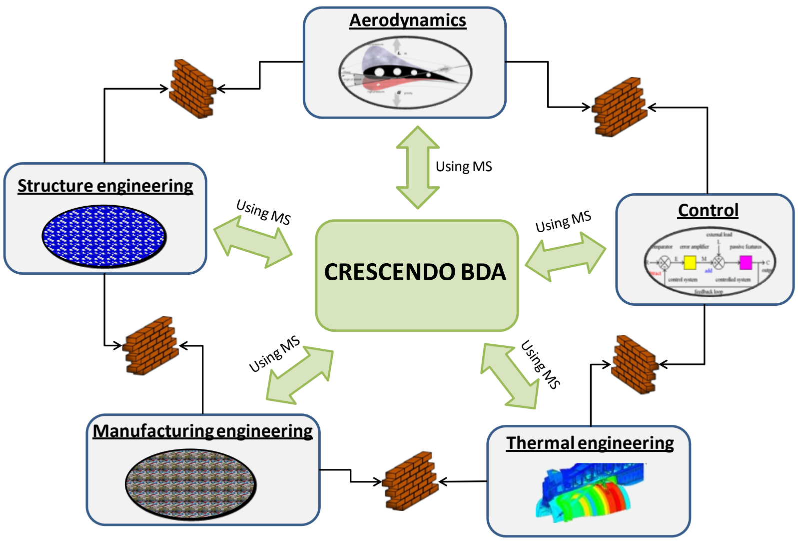


Figure 6.2: Current Barriers and Envisaged CRESCENDO Solution

Among the several components of the BDA, the Model Store (MS) is the core technical result that satisfies the CRESCENDO objective to organize modelling and simulation for multidisciplinary purposes. It proposes a common architecture of models that can be configured to provide the complete behavioural and functional view of the product, enabling to organize and share all models in the Extended Enterprise network. The Model Store delivers innovation in terms of meta-level model architectures, supported by a semantic technology based modelling and simulation dictionary, together with adapted modelling, management and control processes. These results allow models associativity, evolution and context to be managed more effectively throughout the lifecycle. More specifically, the Model Store results will illustrate (CRESCENDO Partners, 2010):

- How to link models developed for different purposes (or by different organisations), in a secure, federated, retrieval and storage collaborative system, so that the combined models can be used to simulate larger applications;
- How to link models in a hierarchical manner to provide consistent views of a product at different levels of abstraction or life cycle stages;
- How to allow a model to understand its own behaviour, input needs, and output capabilities, such that when a new element is added to the system, it will negotiate with the models of all the other elements of the system and fit in the overall common models architecture.

6.4.1.1. Testing in Industry Use Case

During CRESCENDO, several industrial use cases have been defined for validation of the BDA capabilities on a public live demo at the end of the project. One of these use cases is related with aircraft powerplant integration, focusing on product integration chain using a collaborative environment capable of supporting performance requirements and reference powerplant models, as well as keeping a complete traceability of activities and data through the product design lifecycle, from preliminary design optimisation to virtual testing. The use case demonstrates improved simulation set-up processes and multi-partner collaborative workflows, counting with the participation of a number of industrial partners not only from the user's side but also from software development.

Some of the activities and test cases to which this work contributed are illustrated in the application scenario 3, i.e. the idea is that companies involved in the collaborative development of the aircraft porwerplant can, through the use of a common collaboration model and environment, improve the design optimization processes and simulation set-up. In such scenario, a sustainable interoperability is of major importance, as it is the key not only to get different companies working together in an interoperable manner, but also to acknowledge, as soon as possible, the several changes on the product models. This way, rework is reduced through the collaboration network and the partners don't keep working with deprecated model versions (similar objective as the MBSE paradigm presented in the review of methodologies for optimizing complex systems in section 3.6.4).

Using the powerful set of methods for explaining non-linear, emergent behaviour in complex systems, such as CAS, the demonstration of the powerplant integration (prototypes are in the final stages of development) will apply some parts of the complexity framework here addressed, namely the full package for interoperability establishment (including LIMM), and also the communication mediator of the CAS-SIF, which is being presented in a separate demonstrator as a future step in the BDA research.

6.4.2. funStep Initiative

In the furniture sector, this PhD work is contributing to the developments proposed in the funStep initiative where research is validated in a SME enterprise-ruled sector where the usage of product data standards is not yet a reality and electronic partnerships still work on a P2P basis. Industries characterized by a strong presence of small and medium sized enterprises (SMEs), mostly family owned and diversified, are generally flexible and quick in adapting to market changes. However, despite of being fairly ICT equipped, these types of industries face relevant constraints and barriers that are preventing them to evolve and apply that ICT to support e-commerce and e-business, and ultimately to be interoperable with its business partners.

In this sector, companies are frequently guided by their distribution chains and business partners which are using a wide range of systems from different software houses (usually bespoke applications). As an example, furniture products within the Home, Kitchen, Bedroom and/or Bathroom market are becoming more and more complex as the value chain is driven to offer consumers better quality and services. The phenomena has mushroomed the number of choices, and despite accounting with very good software applications, namely CAD/CAM systems, most are incompatible, not being able to "talk" to each other. This diffuse range of systems inhibits interoperability and the development of network-based trading partnerships using effective e-business, thus restricting innovation and development of the sector (INNOVAFUN Partners, 2008).

6.4.2.1. Testing in Industry Use Case

Even though no specific supporting projects were active during most of the PhD period, developments have been validated through the long-term partnership existing among UNINOVA (the author's hosting organisation) and the funStep group of companies. The major focus of the contributions have been towards facilitating the adoption, as a mediation format or the direct implementation, of the AP236 standard for furniture product data exchange (ISO TC184/SC4, 2006). Therefore, from the self-sustaining interoperability framework proposed, the most used package has been the MDA-based package for interoperability establishment where specific translators have been developed to manage the integration among proprietary systems and the corresponding parts of the AP236 standard format (situations similar to some of the ones described in the application scenarios 1 and 2 – section 5.1.1 "Supply Chain in Manufacturing Networks").

However, recently the funStep initiative managed to get support from the EC as part of the IMAGINE project (already introduced), thus enabling a future continuation of the technology transfer to this sector, namely the MIRAI multi-agents system and the CAS-SIF conceptual

framework. IMAGINE is supporting a specific industrial use case in the furniture sector through the development of a living lab demonstrator that addresses the furniture supply chain interactions (IMAGINE Partners, 2012). Companies should be able to connect to the IMAGINE platform that uses blueprint models to interface the services provided and the requests of the various organisations. The key goal of the blueprint model is to gather and modularize the knowledge needed for managing enterprise resources, product lifecycles, partner relationships, operational planning, manufacturing process execution, or safety issues, along the dynamic manufacturing network lifecycle. Blueprinting will provide “intelligence” and improved visibility for an integrated supply chain in a variety of ways, and as some of these blueprints are to be developed based of the AP236 standard (which is already mapped to some industrials using the communication mediator result of the package for interoperability establishment), MIRAI is also being considered for demonstration in this living lab as it can support evolutions and help sustaining interoperability with the IMAGINE platform.

6.5. Hypothesis Validation

In section 1.4 the hypothesis drawn for this dissertation was defined following the research questions and the background analysis. To better recall and revise it, it is here included again:

- *“If the model morphisms, defined at the time of interoperability establishment between the systems’ nodes operating within collaboration networks, are monitored by expert agents capable of triggering CAS-like self-organisation procedures when interoperability harmonization breaking is detected, then dynamic business networks could display sustainability of their systems interoperability”*.

Based on the observations gathered during implementations and feedback received at testing stage, it is possible to conclude that the hypothesis has been validated. Indeed, if the mapping morphisms defined at the time of interoperability establishment are stored in a knowledge intensive repository that enables to trace internally (within the enterprise) all changes registered, then expert agents, in the form of software agents, can conduct a set of activities towards the thesis arguments. In fact, it has been demonstrated that they can monitor the existing model morphisms, detecting harmonization breaking situations independently of the type of system change, i.e. structure, concept, or data. Also, these agents are able to use the meta-information gathered at earlier systems’ activities and auto-propose responses, thus triggering CAS-like self-organisation routines that include human agents’ feedback for the decision process (just as represented in the conceptual solution proposed - Figure 4.7).

Yet, as duly noted along the functional testing analysis of section 6.2.3.1, automation proved sometimes to be complicated, which indicates that despite supporting the self-sustainable interoperability concept, implementations would require more intelligence in CAS-SIF to generate better quality proposals. The transient analysis package for minimizing positive feedback loops is part of that intelligence, but also mappings need always to be strictly defined at both structural and conceptual levels, so that reasoning algorithms can reduce the human involvement. The idea is not

to take the human decision maker out of the loop because that would be unacceptable, but to minimize the time spent in improving the morphisms re-adaptation proposals.

In terms of external consensus, scientific experts have recognised and accepted different publications targeting the many contributions developed along this work, and industrials, via collaboration in international projects, also had the chance to validate and contribute to ideas and implemented POCs. Furthermore, the inclusion of Complexity science as an important neighbouring domain in the initial EISB formulation, as proven to be a very important validation of the hypothesis, recognizing that methods such as the ones inherent to CAS can be (re)used to develop and formalise interoperability solutions in dynamic business networks.

7. Final Considerations and Future Work

This dissertation addresses applied research to endorse the vision that, in the future, innovation and developments on manufacturing, services and ICT-related enterprises can be supported by a self-sustainable interoperability. This enables the evolution of businesses, models, and tools, while maintaining enterprise networked environments interoperable. In fact, interoperability related issues, normally inherent to system changes and network nonlinearity, can be identified and mitigated enabling companies and the networks they belong to, to evolve at their own pace according to different scales of interoperability maturity. Sustained by a novel framework based on complexity, this strategy towards seamless industrial ecosystems becomes feasible, analysing networks as "white boxes" of communication enabling relationships and, applying model-driven technologies and multi-agent systems to manage industrial data, providing companies with effective monitoring and decision support capabilities.

This chapter summarises the results achieved, highlighting the main contributions of the research developed and, proposing future work directions.

7.1. Synthesis of the work

Typically, each company has its own information systems, specific business requirements and, suffers external influences that, even when part of robust collaboration networks, might lead to a harmonization breaking phenomena. In these environments, the interoperability maturity can evolve from *slack* to *unregulated*, *standard-based* or *semantic* interoperability practices. In all, electronic communication is common, and the efficiency of each of them can be calculated based on the proposed mathematical formalisation models. However, this efficiency can vanish whenever

the network begins evidencing interoperability problems. Long term cooperation is, therefore, a challenge, and organisations demand for a *sustainable interoperability*, as well as solutions that support them in the update of business models, software or, information structures, in response to new requirements while minimizing the harmonization breaking impact.

To address this issue and support systems sustainable interoperability, it is required to analyse how intervention strategies on the network evolution, namely how attempts to shape local interaction patterns and mappings, affect the network interoperability sustainability.

This dissertation responds to that need, presenting a conceptual framework (divided in three packages) for a self-sustainable interoperability based in complexity, a discipline that has been largely used as an analytic framework for organisational management and, has also been acknowledged in the design of information systems. Offering a powerful set of methods, complexity science provides a metaphorical language for explaining behaviour in terms of CAS. Based on this, it was possible to describe MIRAI, a “docking” system capable of monitoring the internal environment of the organisation and creating an adequate response to manage harmonisation breaking, involving both humans and information systems as an adaptive single resource. With it, enterprises achieve a state of self-organised criticality without a blueprint or external control mechanism, evolving to the edge of chaos where it they are optimised through adaptation. As envisaged by Kirshbaum (2002) and Couture (2007), the number of interactions is great enough so that novel change, innovation, and evolution can occur, and still not become totally unstable (i.e., quasi-equilibrium state).

The proposed conceptual framework envisages that enterprises willing to join a collaboration network do not have to change their legacy software. Using the MDA/MDI paradigm, the package for interoperability establishment abstracts from organisation’s modelling languages and technology/implementation details, enabling the participation of domain experts in the definition of dedicated model morphisms represented in the form of knowledge enriched tuples (MapT). By using these peer-to-peer and peer-to-reference model mappings to regulate enterprise communications, managers can choose what information they are willing to share, and to whom they will make it available (e.g. restricted to a limited number of enterprises, but not to all network).

The package for sustaining interoperability and the transient evaluation package for minimizing feedback loops, that together form CAS-SIF, respond to the need to capture environment knowledge and relate the human choices and preferences, using monitoring, feedback, and decision support. The mapping morphisms defined at the time of interoperability establishment are stored in a knowledge intensive repository (communication mediator) that enables to trace internally (within the enterprise) all changes registered and detect harmonization breaking before causing real damage in businesses. Also, using that stored knowledge, learning with past actions, and analysing enterprise/network transients with cost-benefit estimations, enable users to make informed decisions and readjustments to be managed more effectively towards re-establishing data exchange as fast, automatically and with less disruption as possible to the network. Also, despite not being approached as such in the dissertation, the transient calculation could support in the

decision regarding the evolution of internal information structures and/or software used to answer to new requirements (model versioning).

The current implementation of CAS-SIF, through MIRAI, has been validated in industrial scenarios, managing the evolution of the networked system nodes, and the entire network dynamics. Using expert agents in the form of software agents, MIRAI monitors all the existing mappings among the several concepts and models used by business partners in the same collaborative network. It controls changes, by warning and proposing new mappings whenever necessary, thus preventing interoperability problems that could cause a destabilization of the network harmony. Enterprises' privacy is assured since each company has its own MIRAI associated to an internal CM storing only the morphims it maintains with its direct partners.

To conclude, sustainable interoperability does not appear from scratch. It is technically built upon previous practices such as standard and semantic interoperability, complementing them with sustainability techniques. Therefore, it also has some benchmark references in terms of efficiency for interoperability establishment and data exchange. The efficiency calculation equations presented as part of the Interoperability Efficiency Pyramid Model (IPyM), can be used in comparison with results obtained with sustainable interoperability, which adds some extra time on the total time spent on the communications (*Cs*) between two organisations (eq.36). In addition to equations 11 (*standard-based interoperability*) or 16 (*semantic interoperability*), it needs to account for the time spent on the sustainability recovery cycle, where for each potential interoperability problem (*p*) at harmonization breaking detection, one needs to account for the time to propose a mapping adaptation (*mapAdapt*), for the human interventions (*ha*), and also for the transient simulation ($\Delta t(transient\ simul)$). However, on the long-term it should pay off, since the network becomes protected from external phenomena that otherwise would decrease efficiency and, produce serious costs associated to production lines and, supply chains becoming paralysed if a major player was affected by the harmonization breaking.

$$\Delta t(Cs) = \sum_{i=0}^{n} \Delta t(m_i) + \Delta t(recl) \quad \textbf{(36)}$$

$$\Delta t(recl) = \begin{cases} 0, & p = 0 \\ \sum_{i=1}^{p} \big(mapAdapt(x)_i + \Delta t(ha_i)\big) + \Delta t(transient\ simul), & p > 0 \end{cases} \quad \textbf{(37)}$$

Both systems and complexity theory approaches have been used in the proof-of-concept implementations. The first tends to focus on exploratory analysis to examine characteristics such as the symbolic content of communication (metadata) while systems theory favours confirmatory analysis or "problem-solving" using feedback-based models. This mixed approach has been contributing for an EI science base (EISB) not only with the framework proposed and technical developments, but also with the analysis conducted relating both research domains (complexity and EI). In fact, as highlighted by Charalabidis et al. (2010), despite of the importance of enterprise interoperability in the global economy, there is yet no established scientific base for EI, and is of

paramount importance to define an EISB that, embodying lessons learnt from scientific neighbouring domains, can help formalising interoperability problems and solutions.

7.2. Main Scientific and Technical Contributions

Main contributions of this thesis can be split in three groups, as follows, and have been included in the publications enumerated in section 6.3 “Acceptance by Scientific Community”.

7.2.1. Conceptual Contributions and Frameworks

- **Sustainability of Systems Interoperability** or in a shorter version, **Sustainable Interoperability**, is a concept introduced as a novel dimension to extend the established interoperability research practices. As evidenced in the EI research roadmaps, today enterprise interoperability has evolved from a technical business systems interconnection problem to a larger domain with multiple dimensions and multidisciplinary issues, that needs to be addressed using a more systemic and holistic methods. Therefore, the concept proposed appears as a natural requirement of the increasingly complexity of the EI domain, addressing the robustness of network relationships and the capacity to absorb disturbances caused by the natural evolution of enterprises.
- The concept of **Harmonization Breaking**, which complements the one of sustainable interoperability, identifying the exact point where a relationship, previsouly stable, begins experiencing interoperability problems. It is a new term proposed for the interoperability domain, equivalent to the “symmetry breaking” term, from classical complexity science.
- The **Self-Sustainable Interoperability Conceptual Framework** is the major contribution of this PhD, drawing concepts from complex systems science, namely CAS, to provide enterprise systems the capacity of dealing internally with the network dynamics, auto-rearranging themselves to face interoperability disruptions. This high-level framework is detailed into two, also novel, (sub)frameworks that separate concerns in:
    - Interoperability establishment - **Modelling Language Independent Framework for Interoperability Establishment**. As a precondition, the conceptual framework needs that enough meta-information about the systems’ relationships is available. This (sub)framework helps defining the necessary mappings for an organisation to enter a collaborative network and become interoperable, without having to switch existing legacy software and enabling the abstraction of technology and implementation details, to allow domain experts to take part on the mapping definitions;
    - Adaptive loop for interoperability (re)establishment - **CAS-based Framework to Support Sustainable Interoperability (CAS-SIF)**, responding to the need to capture environment knowledge and relate the human choices and preferences, using monitoring, feedback and decision support. CAS-SIF involves both humans and information systems as an adaptive single resource.

- The **identification and analysis of the Complexity Science State-of-the-Art in regards to EI**, which not only inspired the conceptual framework proposed, but also enabled the work and concepts developed to be incorporated in the EISB problem and solution spaces formulation. Evaluated by the EI research community, together with researchers from neighbouring scientific domains, these achievements are recognised as encouraging, as they are providing a steady direction for the establishment of a science for EI.

7.2.2. Contributions in the form of New Models and Methodologies

- The proposal of a reference model - **Interoperability Efficiency Pyramid Model (IPyM)** - for the estimation of the efficiency of systems interoperability within networked enterprises, is another relevant contribution. IPyM provides a five-level categorisation of interoperability practices (counting with sustainable interoperability), including formal expressions that enable to classify the different levels of interoperability maturity in these environments.
- The **Language Independent Meta-model (LIMM)**, which serves as an abstract interface on top of enterprises' information models. Through the usage of LIMM, becomes possible to abstract the technology and implementation details associated with the different modelling languages, and thus, enlarge the scope of users involved in a traditional mapping definition activity. Having manager and domains experts involved in this process increases the quality of the mappings that will enable interoperable relationships. It is therefore, an essential part of the modelling language independent framework for interoperability establishment. In comparison to most modelling languages, it is intended to enable as little loss of expressiveness as possible, but at the same time, be simple and generic to support multiple language mappings.
- The **Communication Mediator (CM) Knowledge Base Model** was designed to support the entire self-sustainable interoperability conceptual solution and, addresses traceability as the ability to interrelate the uniquely identifiable object versions in a way that can be processed by either a human or a system.
- The **Knowledge Enriched Tuple for Mappings Representation (MapT)** to tackle the formalisation and coverage gap which has been lacking in MoMo. Some existing formalisation techniques are ideal for structural issues (e.g. model management), others for semantics providing good human traceability (e.g. semantic mismatches), while others are more formal and mathematical based, but none really covered the mixed needs of users and systems.
- The proposed methodological **Activity for Transient Simulation of Adaptations** prepares ground for future research in this direction, even though no specific POC implementations have been developed.
- The **algorithm** proposed **for the Generation of New Mapping Morphisms**, used in the implementation of CAS-SIF. Applying mathematics transitivity principle of binary relations, it is possible to calculate the constituents of MapT (Exp and the derived ATL code) that

enable to realise data exchanges after interoperability is re-established. This algorithm constitutes a formal generalization, thus there are some exceptions duly noted.

7.2.3. Technological Contributions

Although not as scientifically relevant as the contributions previously outlined, the technological achievements made remain, due to their novelty, important contributions of this PhD work.

- The application of **MDA and MDI technology in the layered architecture for interoperability establishment**. It is not very common to see in industry targeted applications of this technology, thus by integrating the multiple inherent concepts, the LIMM, and the CM in a functional POC, this PhD work has contributed to shorten the gap between research and industry.
- The **EXPRESS-LIMM** and **OWL-LIM** mappings are results that endure into future implementations of the self-sustainable interoperability conceptual framework. They can be provided to any person that desires to take up the ideas (all or just some) proposed in this dissertation. With these mapping definitions, the corresponding language meta-models have been validated and, a correction for the W3C OWL meta-model registered.
- Lastly, the **MIRAI Architecture** and the **application of multi-agent technology** in industrial scenarios to prove that concepts of complexity defined **within CAS-SIF** can be implemented in EI solutions.

7.3. Future work

As with any PhD thesis this work cannot be considered completely finalised, and it has given rise to new ideas for further research or in need of further validation.

The choice of MDA/MDI as the enabling technology for the interoperability establishment package was motivated by morphisms modularity and repeatability through the existing landscape of tools available to support horizontal and vertical transformations. Depending on the initial situation (i.e. already having a legacy system, or wanting to develop a new one), either of these transformation methods can prove to be the more efficient to establish interoperability, thus allowing a seamless exchange of information among partners. As evidenced in the framework proposed and, verified by implementations, horizontal transformations may solve interoperability establishment problems. Vertical transformations were not addressed, but could be used to create multiple software and services from scratch; all based on the same requirements, which would enable a nearly automatic mapping definition. This branch of applied research could be explored in the future, checking the feasibility of creating smaller, more parameterized software or services developed specifically for managing temporary business relationships. Nevertheless, since there are scarce implementations of transformations from context independent models (CIM), where the business requirements are specified, to platform independent models (PIM), where the information structure is detailed, new concepts, methods and tools are demanded to cover this gap.

Considering the proposed framework capabilities, at the system microscopic level (i.e., network node), prediction of future behaviour could be seen as a possibility to improve the automatic adaptation of the network morphisms. Thus, to support global interoperability sustainability, CAS-SIF envisages at complementing the diagnosis mechanisms addressed in the presented research (and implemented through MIRAI), with prognosis, based on the history of operations and interventions on the systems involved.

On a complementary line of work, the framework envisages at including more advanced learning capabilities based on the operations history and interventions of the involved systems. The weighting algorithm here used is very simple, and more advanced heuristics for monitoring the individual evolution of each networked system, as well as the dynamics of the entire network could minimize the need of human intervention. Also, automatic detection of harmonization breaking in the cases where model versionings are not updated in the CM knowledge base (e.g. due to forgetfulness) is desired. For example, an interesting line of future research would be the monitoring of real-time communications and analysis of success parameters to determine when a particular system might be experiencing ineffective interoperability.

In fact, the current automatism of the frameworks' implementation concerning the mappings proposition in response to harmonization breaking has some restrictions, which could probably be improved if the conceptual mappings would be used to complement the structural ones. Instead of creating structural mappings based exclusively on structural mappings, or conceptual based on conceptual, the process would benefit from more dynamism. In fact, in the current algorithm for the generation of new mapping morphisms, the composed ATL and Exp (of MapT) used in the new morphism is valid only if $\tau(x, g^{-1})$ is a lossless transformation, otherwise, if conceptual mappings are not considered, data will be lost by this functional composition.

A similar situation occurs in the interoperability establishment, where ATL code could be dynamically generated at the mapping establishment using the knowledge specified by MapT. Although author's belief that is possible, currently there are no ATL libraries available to enable the technical execution of the activity, thus it is being explicitly written by the mapping expert. Moreover, and back to the response to harmonization breaking, if the versioned elements are decomposable (not of KMType = "InstantiableElement"), the new ATL rules should gather automatically the ones redefined by the expert in the sub-elements.

The research concerning the transient evaluation package for minimizing network impact was left at methodological level and, despite integrated in a CAS-SIF layer, no dedicated POC implementation to support massive testing was developed. For future work, besides promoting that validation activity, it would be interesting to analyse the network response transients to a new node addition, as well as the reaction of a network to a node deletion (e.g. due to enterprise insolvency). Up to now, the only possible response is the removal of all morphims connected to that specific node. However, a more helpful and adequate response would require metadata to describe the node's business activities and involve searching of possible replacements in neighbouring networks. This impacts also MIRAI, which future research should enhance the proposed system

and architecture, with components able to manage the dynamic inclusion or exclusion of enterprises in the network.

Concerning the communication mediator knowledge base, its validation with other types of morphisms, such as composition of models, merging operations, etc, would be relevant for future work, especially if one desires to harmonize ontologies, where mappings are not exclusively P2P and can be formed of concepts coming from multiple sources.

Finally, and foreseeing the application scope enlargement, in an electronic world, where business are progressively becoming virtual and business–to-consumer transactions are, many times, executed online, e-marketplaces would be expected to play a relevant role. Nevertheless, current e-marketplace platforms are characterized by low degrees of flexibility to adapt to new business demands, which can only be overcome by greater degrees of interoperability between companies and the electronic platforms, enabling organisations to integrate internal applications and systems with disparate e-platforms. The need for this evolution becomes more acute as the new generation of e-commerce/business platforms are combining the transactional and collaboration functions, merging commercial, managerial, and product/service technical information within clouds (Grilo & Jardim-Goncalves, 2011). Therefore, the framework for sustaining systems interoperability scope could be extended to the cloud, e.g. MIRAI and the MDA-based mapping tool could be extended as SaaS or PaaS, and the communication mediator provided as IaaS. The full conceptual framework could even be studied to regulate inter-cloud communications.

8. References

Aberer, K., Cudré-Mauroux, P., Ouksel, A.M., Catarci, T., Hacid, M.S., Illarramendi, A., Kashyap, V., Mecella, M., Mena, E., Neuhold, E.J., Troyer, O.D., Risse, T., Scannapieco, M., Saltor, F., De Santis, L., Spaccapietra, S., Staab, S. & Studer, R. (2004) Emergent Semantics Principles and Issues. In: *Database Systems for Advanced Applications 9th International Conference (DASFAA 2004)*.

Ackoff, R.L. (1998) A systemic view of transformational leadership. *Systemic Practice and Action Research*, 11 (1), pp.23–36.

Ackoff, R.L. (1971) Towards a system of systems concepts. *Management Science*, 17 (11), pp.661–671. Available from: <http://www.jstor.org/stable/2629308> [Accessed 3 May 2011].

Adams, W.M. (2006) *The future of sustainability: Re-thinking environment and development in the twenty-first century (Report of the IUCN Renowned Thinkers Meeting)*. IUCN - The World Conservation Union. Available from: <http://cmsdata.iucn.org/downloads/iucn_future_of_sustanability.pdf> [Accessed 5 May 2011].

Agostinho, C., Correia, F. & Jardim-Goncalves, R. (2010) Interoperability of Complex Business Networks by Language Independent Information Models. In: *17th ISPE International Conference on Concurrent Engineering (CE2010)*. Cracow, Poland, Springer-Verlag. Available from: <http://www.springerlink.com/content/q775010855762526/>.

Agostinho, C., Dutra, M., Jardim-Goncalves, R., Ghodous, P. & Steiger-Garcao, A. (2007) EXPRESS to OWL morphism: making possible to enrich ISO10303 Modules. In: *14th ISPE International COnference on Concurrent Engineering*. São José dos Campos, Brazil. Available from: <http://www.springerlink.com/index/u7677m72547lg535.pdf> [Accessed 19 April 2011].

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, Springer, pp.194–204. Available from: <http://www.springerlink.com/index/X47W854738585840.pdf> [Accessed 19 April 2011].

Agostinho, C., Jardim-Goncalves, R. & Steiger-Garcao, A. (2011) Using neighboring domains towards setting the foundations for Enterprise Interoperability science. In: *International Symposium on Collaborative Enterprises (CENT 2011) in the context of The 15th World-Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2011*. Orlando, Florida, USA.

Agostinho, C., Sarraipa, J., D'Antonio, F. & Jardim-goncalves, R. (2007) Enhancing STEP-based Interoperabity Using Model Morphisms Model Morphisms - MoMo. In: *3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA'07)*. Funchal, Madeira.

Agostinho, C., Sarraipa, J., Goncalves, D. & Jardim-Goncalves, R. (2011) Tuple-Based Semantic and Structural Mapping for a Sustainable Interoperability. In: *2nd Doctoral Conference on Computing, Electrical and Industrial Systems (DOCEIS'11)*. Caparica, Portugal, Springer.

Ambler, S.W. (2003) Agile Model Driven Development Is Good Enough. *IEEE Software*, 20 (5), pp.71 – 73.

Ambler, S.W. (2011) Effective Practices for Modeling and Documentation [Internet]. Available from: <http://www.agilemodeling.com/> [Accessed 24 January 2011].

Amin, A. & Cohendet, P. (2004) *Architectures of knowledge: firms, capabilities, and communities*. Oxford University Press.

Anderson, C. & Bartholdi, III, J.J. (2000) Centralized versus decentralized control in manufacturing: lessons from social insects. In: I. P. McCarthy & T. Rakotobe-Joel eds. *Complexity and Complex Systems in Industry*. The University of Warwick, pp.92–105.

Anderson, P.W., Arrow, K. & Pines, D. (1988) *The Economy as an Evolving Complex System (Santa Fe Institute Studies in the Sciences of Complexity)*. Reading, MA, USA, Addison-Wesley Longman.

Aronson, D. (1996) Overview of systems thinking [Internet]. Available from: <http://www.thinking.net/Systems_Thinking/OverviewSTarticle.pdf> [Accessed 3 May 2011].

Asnina, E. (2009) Use of Business Models within Model Driven Architecture. *Scientific Journal of Riga Technical University. Computer Sciences*, 38 (38), pp.119–130.

Athena (2005) *Deliverable D.A5.1: Perspectives in Service-Oriented Architectures and their Application in Environments that Require Solutions to be Planned and Customisable*. Athena IP Project (FP6 IST-507849). Available from: <http://interop-vlab.eu/ei_public_deliverables/athena-deliverables/A5/d-a5.1/> [Accessed 7 September 2011].

Athena IP (2007) Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application (FP6 IST-507849) [Internet]. Available from: <http://interop-vlab.eu/ei_public_deliverables/athena-deliverables/list-of-public-deliverables-submitted-during-the-athena-project> [Accessed 30 May 2011].

Athena IP (2006) ATHENA Interoperability Framework (AIF) [Internet]. Available from: <http://modelbased.net/aif/index.html> [Accessed 20 January 2012].

Athena Partners (2007) *Deliverable D.A4.2: Specification of Interoperability Framework and Profiles, Guidelines and Best Practices*. Athena IP Project (FP6 IST-507849). Available from: <http://interop-vlab.eu/ei_public_deliverables/athena-deliverables/A4/d-a4.2>.

Athena Partners (2005) *Deliverable D.B4.3: Scenarios Mapped with Interoperability issues*. Athena IP Project (FP6 IST-507849).

Axelrod, R. (1997) *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. P. W. Anderson, J. M. Epstein, D. K. Foley, S. A. Levin, & G. Mayer-Kress eds. Princeton University Press.

Bak, P., Tang, C. & Wiesenfeld, K. (1988) Self-organized criticality. *Physical Review A*, 38 (1), pp.364–374. Available from: <http://link.aps.org/doi/10.1103/PhysRevA.38.364>.

Barata, J. (2003) Coalition Based Approach for Shop Floor Agility – A Multiagent Approach (PhD Thesis). Faculty of Sciences and Technology, Universidade Nova de Lisboa (FCT/UNL). Available from: <http://run.unl.pt/handle/10362/2483>.

Barrat, A., Barthélemy, M. & Vespignani, A. (2008) *Dynamical Processes on Complex Networks*. Cambridge University Press.

Baïna, K., Benatallah, B., Casati, F. & Toumani, F. (2004) Model-driven Web service development. *Lecture Notes in Computer Science*, 3084, pp.527–543.

Beca, M.F., Agostinho, C., Sarraipa, J. & Jardim-Goncalves, R. (2011) Tuple-based Morphisms For E-procurement Solutions. In: *ASME 2011 International Design Engineering Technical Conference & Computers and Information in Engineering Conference (IDETC/CIE 2011)*. Washington, DC, USA.

Beck, P. (2005) Collaboration vs. Integration: Implications of a Knowledge-Based Future for the AEC Industry. *DesignIntelligence*. Available from: <http://www.di.net/articles/archive/2437/> [Accessed 5 September 2011].

Bellifemine, F., Poggi, A. & Rimassa, G. (2001) Developing multi-agent systems with a FIPA compliant agent framework. *Software: Practice and Experience*, 31 (2), pp.103–128.

Bellinger, G., Castro, D. & Mills, A. (2004) Data, Information, Knowledge, and Wisdom [Internet]. Available from: <http://www.systems-thinking.org/dikw/dikw.htm> [Accessed 7 September 2011].

Benbya, H. & Mckelvey, B. (2006) Toward a Complexity Theory of Inormation Systems Development. *Information Technology and People*, 19 (1), pp.12–34.

Bentley, R.A. & Maschner, H.D.. (2007) Complexity Theory. In: R. A. Bentley, H. D. . Maschner, & C. Chippendale eds. *Handbook of Archaeological Theories*. AltaMira Press, pp.245–270.

Berners-Lee, T. & Fischetti, M. (1999) *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. HarperCollins Publishers.

Berners-Lee, T., Hall, W., Hendler, J.A., O'Hara, K., Shadbolt, N. & Weitzner, D.J. (2006) A Framework for Web Science. *Foundations and Trends® in Web Science*, 1 (1), pp.1–130. Available from: <http://www.nowpublishers.com/product.aspx?product=WEB&doi=1800000001>.

Bernstein, P.A. (2003) Applying Model Management to Classical Meta Data Problems. In: *First Biennial Conference on Innovative Data Systems Research*. California.

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, Madeira, Springer.

Berre, A.-J., Liu, F., Xu, J. & Elvesæter, B. (2009) Model Driven Service Interoperability through Use of Semantic Annotations. In: *2009 International Conference on Interoperability for Enterprise Software and Applications (I-ESA'09)*. Beijing, China, IEEE, pp.90–96.

Bertalanffy, L. Von (1976) *General system theory: foundations, development, applications*. revised ed. George Braziller.

Black, J.A. & Edwards, S. (2000) Emergence of virtual or network organizations: fad or feature. *Journal of Organizational Change Management*, 13 (6), pp.567–576. Available from: <http://www.emeraldinsight.com/10.1108/09534810010378588>.

Blaha, M. & Premerlani, W. (1996) A catalog of object model transformations. In: *Proceedings of the 3rd Working Conference on Reverse Engineering (WCRE '96)*. Monterey, CA , USA. Available from: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=558881>.

Bosin, A., Dessi, N., Fugini, M.G., Liberati, D. & Pes, B. (2005) Applying Enterprise Models to Design Cooperative Scientific Environments. In: *Business Process Management Workshops'2005*. pp.281–292.

Boudjlida, N. & Panetto, H. (2007) Enterprise Semantic Modelling for Interoperability. In: *Proceedings of 12th IEEE Conference on Emerging Technologies and Factory Automation (ETFA'2007)*. Patras, Greece, pp.847–854.

Boulding, K.E. (1956) General systems theory-the skeleton of science. *Management science*, 2 (3), pp.197–208. Available from: <http://www.jstor.org/stable/2627132> [Accessed 3 May 2011].

Breiter, A. (2004) Decision Support Systems in Schools-from Data Collection to Decision Making. In: *Proceedings of the Tenth Americas Conference on Information Systems*. New York.

Brown, S.L. & Eisenhardt, K.M. (1997) The art of continous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations. *Administrative Science Quarterly*, 42 (1).

Brunnermeier, S.B. & Martin, S.A. (1999) *Interoperability cost analysis of the US automotive supply chain (NIST 99-1 Planning Report)*. Gaithersburg, MD, USA, NIST.

BuildingSMART (2010) Industry Foundation Classes (IFC) [Internet]. Available from: <http://buildingsmart.com/standards/ifc> [Accessed 7 September 2011].

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. & Brandic, I. (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25 (6), pp.599–616.

Bézivin, J. (2005) Model Driven Engineering: An Emerging Technical Space. In: *Generative and Transformational Techniques in Software Engineering, International Summer School (GTTSE 2005)*. Braga, 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*. p.50.

C4ISR Architecture Working Group (1997) *C4ISR Architecture Framework (version 2.0)*. U.S. Department of Defense.

Calvano, J. V., Alves, V.C. & Lubaszewski, M. (1999) Fault detection methodology for second order filters using compact test vectors transient analysis. In: *Proceedings of Third International Design of Mixed-Mode Integrated Circuits and Applications Workshop*.

Camarinha-Matos, L.M. (2009) Scientific Research Methodologies and Techniques [Internet]. Available from: <www.uninova.pt/~cam/teaching/srmt.htm> [Accessed 1 July 2010].

Camarinha-Matos, L.M. & Afsarmanesh, H. (2004) *Collaborative networked organizations: a research agenda for emerging business models*. Springer.

Camarinha-Matos, L.M. & Afsarmanesh, H. (2005) Collaborative networks: a new scientific discipline. *Journal of Intelligent Manufacturing*, 16 (4-5), pp.439–452. Available from: <http://www.springerlink.com/index/10.1007/s10845-005-1656-3>.

Camarinha-Matos, L.M. & Afsarmanesh, H. (1999) The Virtual Enterprise Concept. In: *Proceedings of the IFIP TC5 WG5.3 / PRODNET Working Conference on Infrastructures for Virtual Enterprises: Networking Industrial Enterprises*.

Campos, C., Grangel, R., Chalmeta, R. & Coltell, Ó. (2005) Requirements to improve the synchronization of inter-enterprise models. In: *Business Process Management Workshops*. Springer, pp.353–362. Available from: <http://www.springerlink.com/content/y8w826k0524h40n6/>.

Carpi, A. & Egger, A. (2008) CHED 392-Teaching scientific process and the nature of science. In: *Proceedings of: 236th National Meeting of the American-Chemical-Society*. Philadelphia PA, USA.

Carpi, A. & Egger, A. (2010) The Process of Science [Internet]. Available from: <http://www.visionlearning.com/process_science.php> [Accessed 5 September 2011].

CEN/ISSS (2006) *eBusiness Roadmap: addressing key eBusiness standards issues 2006-2008*. CEN/ISSS. Available from: <ftp://ftp.cen.eu/cen/Sectors/List/ICT/FocusGroups/eBusRoadmap.pdf> [Accessed 20 January 2012].

Chalmeta, R., Campos, C. & Grangel, R. (2001) References architectures for enterprise integration. *The Journal of Systems and Software*, 57 (3), pp.175–191.

Charalabidis, Y., Jardim-Goncalves, R. & Popplewell, K. (2010) Towards a Scientific Foundation for Interoperability. In: Y. Charalabidis ed. *Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business*. IGI Global.

Checkland, P. (1999) *Systems Thinking, Systems Practice: Includes a 30-Year Retrospective*. 1st editio. Wiley.

Chen, D. & Doumeingts, G. (2004) European Initiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap. *Journal of Annual reviews in Control*, 27 (2), pp.151–160.

Chen, D. & Doumeingts, G. (1996) The GRAI-GIM reference model, architecture and methodology. In: P. Bernus, L. Nemes, & T. J. Williams eds. *Architectures for Enterprise Integration*. London, UK, Chapman & Hall.

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. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0166361508000365> [Accessed 27 August 2010].

Chen, D. & Vernadat, F. (2002) Enterprise Interoperability: A Standardisation View. In: *Proceedings of IFIP TC5/WG5.12 International Conference on Enterprise Integration and Modeling Technique: Enterprise Inter- and Intra-Organizational Integration: Building International Consensus*.

Chen, L. & Li, S. (2004) Analysis of decomposability and complexity for sedign problems in the context of decomposition. In: *Proceedings of DETC'04 ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Salt Lake City, Utah, USA.

Chestnut, H. (1965) *Systems Engineering Tools*. Wiley.

Choi, T., Dooley, K.J. & Rungtusanatham, M. (2001) Supply networks and complex adaptive systems: control versus emergence. *Journal of Operations Management*, 19 (3), pp.351–366. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0272696300000681>.

Chu, C.-H., Chang, C.-J. & Cheng, H.-C. (2006) Empirical Studies on Inter-Organizational Collaborative Product Development. *Journal of Computing and Information Science in Engineering*, 6 (2), pp.179–188.

Clark, Thea & Jones, R. (1999) Organisational Interoperability Maturity Model for C2. In: *Proceedings of the Command And Control Research And Technology Symposium (CCRTS)*. Newport, RI, USA. Available from: <http://www.sei.cmu.edu/library/assets/orginteroper.pdf>.

Clark, Tony, Evans, A. & Kent, S. (2002) Engineering modelling languages: A precise meta-modelling approach. In: *FASE '02 Proceedings of the 5th International Conference on Fundamental Approaches to Software Engineering*. Grenoble, France, Springer. Available from: <http://www.springerlink.com/index/6pt85yvb0v8prgya.pdf> [Accessed 3 May 2011].

Clauset, A., Shalizi, C.R. & Newman, M.E.J. (2009) Power-law distributions in empirical data. *SIAM Review*, 51 (4), pp.661–703.

Clemens, M. (2004) Complexity Illustrations [Internet]. Available from: <www.idiagram.com/examples/complexity.html> [Accessed 7 September 2011].

Comptia (2004) *White Paper - European Interoperability Framework: ICT Industry Recommendations*. Brussels, Comptia.

Courtney, J., Merali, Y., Paradice, D. & Wynn, E. (2008) On the study of complexity in information systems. *International Journal of Information Technologies and The Systems Approach*, 1 (1), pp.37–48.

Couture, M. (2007) *Complexity and chaos - State-of-the-art; Overview of theoretical concepts (Technical Memorandum DRDC Valcartier TM 2006-453)*. Canada, Valcartier.

CRESCENDO IP (2009) Collaborative and Robust Engineering using Simulation Capability Enabling Next Design Optimisation (FP7-234344) [Internet]. Available from: <http://www.crescendo-fp7.eu/> [Accessed 7 May 2013].

CRESCENDO Partners (2009a) *Deliverable D5.2.1: Model Store SOTA & Vision*. CRESCENDO IP Project (FP7-234344).

CRESCENDO Partners (2010) *Deliverable D5.2.4: BDA Model Store Specifications Dossier (Intermediate Version)*. CRESCENDO IP Project (FP7-234344).

CRESCENDO Partners (2009b) *Deliverable D5.5.1: Enterprise Collaboration State-of-Art Review Report*. CRESCENDO IP Project (FP7-234344).

Czarnecki, K. & Helsen, S. (2006) Feature-based survey of model transformation approaches. *IBM Systems Journal*, 45 (3), pp.621–645.

Dauben, J.W. (1979) *Georg Cantor: His Mathematics and Philosophy of the Infinite*. Harvard University Press.

Davis, R. & Smith, R.G. (1983) Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, 20 (1), pp.63–109.

Deakin, a. G., Rallis, I., Zhang, J., Spencer, J.W. & Jones, G.R. (2006) Towards holistic chromatic intelligent monitoring of complex systems. *Sensor Review*, 26 (1), pp.11–17. Available from: <http://www.emeraldinsight.com/10.1108/02602280610640616> [Accessed 3 May 2011].

Dekkers, R. (2009) Collaborations in Industrial Networks: the Co-Evolutionary Perspective. In: R. Dekkers ed. *Dispersed Manufacturing Networks*. Springer.

Delgado, M. (2008) Harmonisation of STEP and MDA conceptual models using Model Morphisms (MSc Thesis). Faculty of Sciences and Technology, Universidade Nova de Lisboa (FCT/UNL).

Delgado, M., Agostinho, C., Malo, P. & Jardim-Goncalves, R. (2006) A framework for STEP-based harmonization of conceptual models. In: *3rd International IEEE Conference Intelligent Systems (IEEE IS'06)*. London, UK, IEEE, pp.781–785.

Diestel, R. (2010) *Graph Theory (Graduate Texts in Mathematics, Volume 173)*. 4th ed. Springer-Verlag, Heidelberg.

DoD (2011) *Department of Defense Dictionary of Military and Associated Terms (Joint Publication 1-02)*. U.S. Department of Defense. Available from: <http://www.dtic.mil/doctrine/new_pubs/jp1_02.pdf> [Accessed 20 January 2012].

DoD (1977) DoD Directive 2010.6 "Standardization and Interoperability of Weapon Systems and Equipment within the North Atlantic Treaty Organization (NATO)."

Dodig-Crnkovic, G. (2002) Scientific methods in computer science. In: *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden*. Skövde, Sweden, pp.126–130. Available from: <http://www.mrtc.mdh.se/~gdc/work/cs_method.pdf> [Accessed 3 May 2011].

Doumeingts, G., Vallespir, B. & Chen, D. (2006) GRAI grid decisional modelling P. Bernus, K. Mertins, & G. Schmidt eds. *International Handbook of Information Systems*, 1, pp.321–346. Available from: <http://www.springerlink.com/content/mql0264427327855/>.

Durfee, E.H. (2004) Challenges to scaling-up agent coordination strategies. In: T. A. Wagner ed. *An Application Science for Multi-Agent Systems*. Dordrecht, Netherlands, Kluwer Academic Publishers, pp.113–132.

Durresi, A., Durresi, M. & Barolli, L. (2009) Heterogeneous Multi Domain Network Architecture for Military Communications. In: *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS '09)*.

Dyer, J.H. (2000) *Collaborative advantage: Winning through extended enterprise supplier networks*. Oxford University Press.

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.

Eclipse foundation (2011) ATL Transformation Language [Internet]. Available from: <http://eclipse.org/atl/> [Accessed 7 September 2011].

Edgington, T., Choi, B., Henson, K., Raghu, T.S. & Vinze, A. (2004) Adopting Ontology to Facilitate Knowledge Sharing. *Communications of the ACM*, 47 (11), pp.85–90.

Elvesæter, B., Hahn, A., Berre, A. & Neple, T. (2005) Towards an interoperability framework for model-driven development of software systems. In: *1st International Conference on Interoperability for Enterprise Software and Applications (I-ESA'05)*. Springer. Available from: <http://www.springerlink.com/index/L10NU4306N054T6G.pdf> [Accessed 3 May 2011].

ENSEMBLE CSA (2011) Envisioning, Supporting and Promoting Future Internet Enterprise Systems Research through Scientific Collaboration (FP7-ICT-257548) [Internet]. Available from: <http://www.fines-cluster.eu/fines/jm/ENSEMBLE-Public-Category/ensemble-project.html> [Accessed 31 May 2011].

ENSEMBLE Partners (2012) *Deliverable D2.4: EISB Models & Tools Report*. ENSEMBLE CSA Project (FP7-ICT-257548). Available from: <http://www.fines-cluster.eu/fines/jm/Publications/ENSEMBLE-Deliverables/View-category.html>.

Enterprise Interoperability Cluster (2008) *Enterprise Interoperability Research Roadmap, Version 5.0*. European Commision. Available from: <http://cordis.europa.eu/fp7/ict/enet/ei-research-roadmap_en.html> [Accessed 3 May 2011].

Estefan, J.A. (2007) *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. INCOSE. Available from: <http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf> [Accessed 7 September 2011].

European Comission (2006) Information Society Technologies (IST) [Internet]. Available from: <http://cordis.europa.eu/ist/projects/projects.htm> [Accessed 1 July 2009].

European Comission (2011) Towards the Internet of Services [Internet]. Available from: <http://cordis.europa.eu/fp7/ict/ssai/home_en.html> [Accessed 27 January 2012].

European Science Foundation (2010) Future Internet and Society: A Complex Systems Perspective [Internet]. Available from: <www.esf.org/index.php?id=6633> [Accessed 7 September 2011].

Farinha, F., Steiger-Garcao, A. & Jardim-Goncalves, R. (2007) Integration of cooperative production and distributed design in AEC. *Advances in Engineering Software*, 38 (11-12), pp.772–779.

Farlex (2011) proof of concept [Internet]. Available from: <http://encyclopedia2.thefreedictionary.com/proof+of+concept>.

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., Agostinho, C., Sarraipa, J. & Jardim-Goncalves, R. (2011) Monitoring morphisms to support sustainable interoperability of enterprise systems. In: *On the Move to Meaningful Internet Systems: OTM 2011 Workshops (EI2N 2011)*. Hersonissos, Crete, Greece.

FInES Research Roadmap Task Force (2010) *Future Internet Enterprise Systems (FInES) Research Roadmap: version 3.0*. European Commission. Available from: <http://cordis.europa.eu/fp7/ict/enet/documents/task-forces/research-roadmap/fined-researchroadmap.pdf>.

Foster, I., Kesselman, C., Nick, J.M. & Tuecke, S. (2002) Grid Services for Distributed Systems Integration. *IEEE Computer*, 35 (6).

Franconi, E. (2004) Using Ontologies. *IEEE Intelligent Systems*, 19 (1), pp.76–79.

Frankel, D. (2003) *Model Driven Architecture – Applying MDA to Enterprise Computing*. 1st editio. OMG Press.

Freetutes.com (2007) Systems Analysis and Design - Complete Introductory Tutorial for Software Engineering [Internet]. Available from: <www.freetutes.com/systemanalysis/ > [Accessed 7 September 2011].

Frei, R., Barata, J. & Serugendo, G.D.M. (2007) A complexity theory approach to evolvable production systems. In: *Internacional Conferecene in informatics and control, automation and robotics*. Angers, France. Available from: <http://www.dcs.bbk.ac.uk/~dimarzo/papers/MARS07.pdf>.

Friedenthal, S., Moore, A. & Steiner, R. (2008) *A Practical Guide To SysML: The Systems Modeling Language*. Morgan Kaufmann.

Friedman, T. (2005) *The World is Flat: a Brief History of the Twenty-First Century*. Farrar, Straus & Giroux.

Fuentes-Fernández, L. & Vallecillo-Moreno, A. (2004) An Introduction to UML Profiles. *UPGRADE European Journal for the Informatics Professional*, V (2), pp.6–13.

FunStep (1998) funStep Initiative [Internet]. Available from: <http://www.funstep.org/> [Accessed 5 September 2011].

Gallaher, M.P., O'Connor, A.C., Dettbarn, J.L. & Gilday, L.T. (2004) *Cost analysis of inadequate interoperability in the US capital facilities industry (NIST GCR 04-867)*. Gaithersburg, MD, USA, NIST.

Ganeshan, R. & Harrison, T.P. (1995) An Introduction to Supply Chain Management [Internet]. Available from: <http://lcm.csa.iisc.ernet.in/scm/supply_chain_intro.html> [Accessed 15 September 2011].

Gasser, U. & Palfrey, J.G. (2007) *Case Study: Mashups Interoperability and eInnovation (Berkman Center Research Publication No. 2007-10)*. Berkman Center. Available from: <http://cyber.law.harvard.edu/interop/pdfs/interop-mashups.pdf> [Accessed 7 September 2011].

Gharajedaghi, J. (2005) *Systems Thinking: Managing Chaos and Complexity: A Platform for Designing Business Architecture*. Butterworth-Heinemann.

Gielingh, W. (2008) An assessment of the current state of product data technologies. *Computer-Aided Design*, 40 (7), pp.750–759.

Gilbert, N. (2007) *Agent-based models*. Sage Publications, Inc.

Giunchiglia, F., Yatskevich, M. & Shvaiko, P. (2007) Semantic matching: Algorithms and implementation. *Journal on Data Semantics IX*, pp.1–38.

Gleick, J. (2008) *Chaos: Making a new science*. Penguin.

Van Gorp, P. (2008) Model-driven development of model transformations. In: *Proceedings of the 4th international conference on Graph Transformations (ICGT '08)*. Leicester, UK, Springer. Available from: <http://www.springerlink.com/index/RL725074894W241J.pdf> [Accessed 3 May 2011].

Gorton, I. (2006) *Essential Software Architecture*. Berlin Heidelberg, Springer-Verlag.

Gottschalk, P. (2009) Maturity levels for interoperability in digital government. *Government Information Quarterly*, 26 (1), pp.75–81.

Grangel, R., Bigand, M. & Bourey, J.-P. (2008) A UML Profile as Support for Transformation of Business Process Models at Enterprise Level. In: *MDISIS'08 - Model Driven Interoperability for Sustainable Information Systems held in conjunction with CAiSE'08*. Montpellier, France. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.143.1830>.

Greenfield, J., Short, K., Cook, S. & Kent, S. (2004) *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. Wiley.

Grilo, A. & Jardim-Goncalves, R. (2011) Challenging Electronic Procurement in the AEC Sector: A BIM-based Integrated Perspective. *Automation in Construction*, 20 (2), pp.107–114.

Grilo, A. & Jardim-Goncalves, R. (2010) Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*, 19 (5), pp.522–530.

Gruber, T.R. (1993) A translation approach to portable ontology specifications. *Knowledge acquisition*, 5 (2), pp.199–220.

Gruber, T.R. (2007) Ontology of Folksonomy: A Mash-up of Apples and Oranges. *International Journal on Semantic Web & Information Systems*, 3 (2).

Gruber, T.R. (1995) Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43 (5-6), pp.907–928.

Guarino, N. (1995) Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, 43 (5-6), pp.625–640.

Guarino, N., Oberle, D. & Staab, S. (2009) What Is an Ontology? In: S. Staab & Rudi Studer eds. *Handbook on Ontologies*. Berlin, Heidelberg, Springer Berlin Heidelberg. Available from: <http://www.springerlink.com/index/10.1007/978-3-540-92673-3> [Accessed 1 July 2010].

Gunasekaran, A. & Ngai, E.W.T. (2005) Build-to-order supply chain management: a literature review and framework for development. *Journal of Operations Management*, 23 (5), pp.423–451. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0272696304001184>.

Hailpern, B. & Tarr, P. (2006) Model-driven development: The good, the bad, and the ugly. *IBM Systems Journal*, 45 (3).

Haykin, S.S. (2008) *Neural networks and learning machines*. Prentice Hall.

Heylighen, F. & Joslyn, C. (1992) What is Systems Theory? [Internet]. Available from: <http://pespmc1.vub.ac.be/systheor.html> [Accessed 7 September 2011].

Ho, M.T. (1998) Systems Analysis. In: J. M. Stellman ed. *Encyclopaedia of Occupational Health and Safety (Volume 2)*.

Holland, J.H. (1992) Complex adaptive systems. *Daedalus*, 121 (1), pp.17–30.

Holland, J.H. (1998) *Emergence: From Chaos to Order*. Perseus Books. Available from: <http://video.yahoo.com/watch/111582/992708>.

Holland, J.H. (1996) *Hidden Order: How Adaptation Builds Complexity*. Perseus Books.

Honour, E. (2008) Systems Engineering and Complexity. *INCOSE Insight*, 11 (1), p.20.

Huhns, M.N. & Stephens, L.M. (1999) Multiagent systems and societies of agents. In: M. Weiss ed. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press.

I-Surf Partners (2009) *Deliverable 8.1.1 – Functional and Non-Functional Evaluation Criteria for i-Surf Components and Integrated Platform*. i-Surf STREP Project (FP7 213031). Available from: <http://www.srdc.com.tr/projects/isurf/>.

I-Surf STREP (2010) An Interoperability Service Utility for Collaborative Supply Chain Planning across Multiple Domains Supported by RFID Devices (FP7 213031). *Public Website*. Available from: <http://www.srdc.com.tr/projects/isurf/> [Accessed 2 May 2012].

Ide, N., Pustejovsky, J., Calzolari, N. & Soria, C. (2009) The SILT and FlaReNet international collaboration for interoperability. In: *Proceedings of the Third Linguistic Annotation Workshop on - ACL-IJCNLP '09*. Morristown, NJ, USA, Association for Computational Linguistics, pp.178–181. Available from: <http://portal.acm.org/citation.cfm?doid=1698381.1698415>.

IDEAS Partners (2003) *Deliverable D2.4: The Vision for 2010*. IDEAS Network (FP5 IST-2001-37368). Available from: <http://interop-vlab.eu/ei_public_deliverables/ideas-deliverables/d-2-4-the-vision-for-2010>.

IDEAS Thematic Network (2003) Interoperability Development for Enterprise Applications and Software - Roadmaps (FP5 IST-2001-37368) [Internet]. Available from: <http://interop-vlab.eu/ei_public_deliverables/ideas-deliverables> [Accessed 30 May 2011].

IEEE (1990) IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries (IEEE 610-1990).

IMAGINE IP (2011) Innovative End-to-end Management of Dynamic Manufacturing Networks (FP7 285132) [Internet]. Available from: <http://www.imagine-futurefactory.eu> [Accessed 8 May 2012].

IMAGINE Partners (2012) *Deliverable D1.1.1 - Compendium on virtual manufacturing management*. IMAGINE IP (FP7 285132).

INCOSE & OMG DSIG (2011) Model Based Systems Engineering [Internet]. Available from: <http://www.omgwiki.org/MBSE/doku.php> [Accessed 26 February 2012].

INCOSE, T.O. (2007) *Systems Engineering Vision 2020 (INCOSE-TP-2004-004-02)*. INCOSE. Available from: <www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf> [Accessed 9 April 2012].

Informal Study Group on Value Proposition for Enterprise Interoperability (2008) *Unleashing the Potential of the European Knowledge Economy Value Proposition for Enterprise Interoperability*. European Comission. Available from:

<http://cordis.europa.eu/documents/documentlibrary/100123101EN6.pdf> [Accessed 3 May 2011].

INNOVAFUN Partners (2008) *Deliverable 1.3: Innovation impact on furniture organisations resulting from funStep standard implementation*. INNOVAFUN CA project (FP6 031139).

Interedition (2008) Interedition (ESF COST Action IS0704) [Internet]. Available from: <www.interedition.eu/> [Accessed 7 September 2011].

INTEROP Partners (2006a) *DELIVERABLE D.A3.3: Semantic Annotation language and tool for Information and Business Processes Apendix F: User Manual*. INTEROP NoE Project (FP6 IST-1-508011).

INTEROP Partners (2006b) *Deliverable D12.4: Report on Future research to support interoperability for SMEs*. INTEROP NoE Project (FP6 IST-1-508011). Available from: <http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/wp12-service-take-up-towards-smes/d12-4>.

INTEROP Partners (2006c) *Deliverable DI.2: Enterprise Interoperability-Framework and knowledge corpus-Advanced report*. INTEROP NoE Project (FP6 IST-1-508011). Available from: <http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/di-domain-interoperability/di-2-enterprise-interoperability-framework-and-knowledge-corpus/> [Accessed 5 September 2011].

INTEROP Partners (2005a) *Deliverable DTG2.1: REPORT ON MODEL ESTABLISHMENT*. INTEROP NoE Project (FP6 IST-1-508011). Available from: <http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/tg2-model-driven-interoperability/DTG2.1/>.

INTEROP Partners (2007a) *Deliverable DTG2.3: REPORT ON MODEL DRIVEN INTEROPERABILITY*. INTEROP NoE Project (FP6 IST-1-508011). Available from: <http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/tg2-model-driven-interoperability/dtg2-3-report-on-model-driven-interoperability/>.

INTEROP Partners (2005b) *Deliverable DTG3.1 (MoMo.2): MoMo Roadmap*. INTEROP NoE Project (FP6 IST-1-508011). Available from: <http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/tg3-model-morphisms/DTG3.1/>.

INTEROP Partners (2007b) *Deliverable DTG3.3: Extended toolbox definition and dissemination report*. INTEROP NoE Project (FP6 IST-1-508011). Available from: <http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/tg3-model-morphisms/dtg3-3_extended-toolbox-definition-and-dissemination-report/>.

InterOP-VLab (2008) Model Driven Interoperability for Sustainable Information Systems Workshop (MDISIS'08) [Internet]. Available from: <http://mdisis08.ec-lille.fr/> [Accessed 29 January 2012].

ISA, I.S. for E.P.A. (2010) *European Interoperability Framework (EIF) for European public services*. Available from: <http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf>.

ISO TC184/SC4 (2004) Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual (ISO 10303-11:2004).

ISO TC184/SC4 (2006) Industrial automation systems and integration -- Product data representation and exchange -- Part 236: Application protocol: Furniture catalog and interior design (ISO 10303-236:2006). Available from: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=42340>.

ISO TC184/SC4 (1994) Standard for the Exchange of Product Data - ISO 10303 (STEP). Available from: <http://www.tc184-sc4.org> [Accessed 1 July 2009].

ISO TC184/SC5 (2011) Advanced automation technologies and their applications — Part 1 : Framework for enterprise interoperability (ISO/DIS 11354-1:2011).

ISO TC184/SC5 (2000) Industrial Automation Systems—Requirements for Enterprise-reference Architectures and Methodologies (ISO 15704:2000). Available from: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=28777>.

ISO/IEC (2004) Information Technology - Metadata registries (MDR) - Part 1: Framework (ISO/IEC 11179-1:2004). Available from: <http://metadata-standards.org/11179/>.

ISO/IEC (1999) Information technology—Specification and standardization of data elements— Part 1: Framework for the specification and standardization of data elements (ISO/IEC 11179-1:1999).

ISO/IEC (2001a) Software engineering – Product evaluation – Part 6: Documentation of evaluation modules (ISO/IEC 14598-6).

ISO/IEC (2007) Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements (ISO/IEC CD 25030).

ISO/IEC (2001b) Software Engineering-Software product quality-Part 1: Quality model (ISO/IEC 9126-1).

Jalali, S. (2009) *(White Paper) Collaborative Product Development –Challenges and Best Practices*. Tata Consultancy Services Limited. Available from: <http://www.tcs.com/SiteCollectionDocuments/White Papers/HiTech_WhitePaper_Collaborative Product Development_04_2010.pdf>.

Jardim-Goncalves, R., Agostinho, C., Malo, P. & Steiger-garcao, A. (2007) Harmonising technologies in conceptual models representation. *International Journal of Product Lifecycle Management*, 2 (2), pp.187–205.

Jardim-Goncalves, R., Agostinho, C., Sarraipa, J., Grilo, A. & Mendonça, J.P. (2012) Reference Framework for Enhanced Interoperable Collaborative Networks in Modern Furniture Industrial Organizations: The funStep Case Study. *Approved and in publication at International Journal of Computer Integrated Manufacturing*.

Jardim-Goncalves, R., Agostinho, C., Sarraipa, J., Roca-Togores, A., Nunez, M.J. & Panetto, H. (2011) Standards Framework for Intelligent Manufacturing Systems Supply Chain. In: S. Renko ed. *Supply Chain Management - New Perspectives*. InTech. Available from: <http://www.intechopen.com/articles/show/title/standards-framework-for-intelligent-manufacturing-systems-supply-chain>.

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*, (February), pp.1–19. Available from: <http://www.tandfonline.com/doi/abs/10.1080/0951192X.2011.653831> [Accessed 14 February 2012].

Jardim-Goncalves, R., Agostinho, C. & Steiger-Garcao, A. (2010) Sustainable systems' interoperability: A reference model for seamless networked business. In: *2010 IEEE International Conference on Systems Man and Cybernetics (SMC),*. Istambul, Turkey, IEEE.

Jardim-Goncalves, R., Figay, N. & Steiger-Garcao, A. (2006) Enabling interoperability of STEP Application Protocols at meta-data and knowledge level. *International Journal of Technology Management*, 36 (4), pp.402–421.

Jardim-Goncalves, R., Grilo, A., Agostinho, C., Lampathaki, F. & Charalabidis, Y. (2012) Systematisation of Interoperability Body of Knowledge: The foundation for EI as a science. *Enterprise Information Systems*, (Special Issue "Information Systems for Enterprise Integration, Interoperability and Networking: Theory and Applications").

Jardim-Goncalves, R., Grilo, A. & Steiger-Garcao, A. (2006) Challenging the interoperability industry with MDA between computers in and SOA. *Computers in Industry*, 57 (8-9), pp.679–689.

Jardim-Goncalves, R., Sarraipa, J., Agostinho, C. & Panetto, H. (2009) Knowledge framework for intelligent manufacturing systems. *Journal of Intelligent Manufacturing*, 22 (5), pp.725–735. Available from: <http://www.springerlink.com/index/10.1007/s10845-009-0332-4> [Accessed 8 April 2011].

Jardim-Goncalves, R., Silva, J.P.M.A., Steiger-Garcao, A. & Monteiro, A. (2007) Framework for Enhanced Interoperability Through Ontological Harmonization of Enterprise Product Models in book Ontologies: A Handbook of Principles, Concepts and Applications. *Integrated Series in Information Systems*, 14 (XXIV).

Jardim-Goncalves, R. & Steiger-Garcao, A. (2002) Implicit Multilevel Modeling in Flexible Business Environments. *Communications of the ACM*, 45 (10), pp.53–57. Available from: <http://portal.acm.org/citation.cfm?id=570907.570931>.

Jennings, N.R., Sycara, K. & Wooldridge, M. (1998) A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1 (1), pp.7–38.

Jouault, F. & Kurtev, I. (2007) On the interoperability of model-to-model transformation languages. *Science of Computer Programming*, 68, pp.114–137.

Juric, M.B., Loganathan, R., Sarang, P. & Jennings, F. (2007) *SOA Approach to Integration: XML, Web services, ESB, and BPEL in real-world SOA projects*. Packt Publishing.

Kamrani, A.K. (2008) Collaborative Design Approach in Product Design and Development. In: A. K. Kamrani & E. A. Nasr eds. *Collaborative Engineering: Theory and Practice*. Springer.

Kauffman, S. (1996) *At home in the universe: The search for laws of self-organization and complexity*. Oxford University Press.

Kauffman, S. & Clayton, P. (2006) On emergence, agency, and organization. *Biology Philosophy*, 21 (4), pp.501–521. Available from: <http://www.springerlink.com/index/10.1007/s10539-005-9003-9>.

Kelley, T.D. (2003) Symbolic and Sub-symbolic Representations in Computational Models of Human Cognition. What Can be Learned from Biology? *Theory & Psychology*, 13 (6), pp.847–860.

Kempler, S., Lynnes, C., Vollmer, B., Alcott, G. & Berrick, S. (2009) Evolution of Information Management at the GSFC Earth Sciences (GES) Data and Information Services Center (DISC): 2006–2007. *IEEE Transactions on Geoscience and Remote Sensing*, 47 (1), pp.21–28. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4694057>.

Kephart, J.O. & Chess, D.M. (2003) The vision of autonomic computing. *Computer*, 36 (1), pp.41–50.

Kern, E.-M. & Kersten, W. (2007) Framework for internet-supported inter-organizational product development collaboration. *Journal of Enterprise Information Management*, 20 (5), pp.562–577.

Khayari, R.E.A., Sadre, R., Haverkort, B.R. & Ost, A. (2004) The pseudo-self-similar traffic model: application and validation. *Performance Evaluation - Dependable systems and networks-performance and dependability symposium*, 56 (1-4).

Kirshbaum, D. (2002) Introduction to Complex Systems [Internet]. Available from: <http://www.calresco.org/intro.htm> [Accessed 7 September 2011].

Komatsoulis, G.A., Warzel, D.B., Hartel, F.W., Shanbhag, K., Chilukuri, R., Fragoso, G., Coronado, S., Reeves, D.M., Hadfield, J.B., Ludet, C. & Covitz, P.A. (2008) caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. *Journal of biomedical informatics*, 41 (1), pp.106–123.

Krause, F.-L. & Kaufmann, U. (2007) Meta-Modelling for Interoperability in Product Design. *CIRP Annals - Manufacturing Technology*, 56 (1), pp.159–162.

Kundert, K. (2003) Transient analysis. In: K. Kundert ed. *The Designer's Guide to Spice and Spectre*. Springer US, pp.129–250.

Lampathaki, F., Koussouris, S., Agostinho, C., Jardim-Goncalves, R., Charalabidis, Y. & Psarras, J. (2011) Towards an Interoperability Science : Cultivating the Scientific Foundations for Enterprise Interoperability. In: *International Symposium on Collaborative Enterprises (CENT 2011) in the context of The 15th World-Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2011*. Orlando, Florida, USA.

Ledesma, P. & Usaola, J. (2005) Doubly fed induction generator model for transient stability analysis. *IEEE Transactions on Energy Conversion*, 20 (2), pp.388–397.

Levy, D.L. (2000) Applications and limitations of complexity theory in organization theory and strategy. In: J. Rabin, G. Miller, & W. B. Hildreth eds. *Handbook of strategic management*. M. Dekker, pp.67–87.

Lewin, D.L. (1999) Application of Complexity Theory to Organization Science. *Organization Science*, 10 (3), p.215.

Lewis, G. (2007) Systems Thinking About Purpose. *Australasian Journal of Information Systems*, 10 (1), pp.50–58.

Lewis, R. (1994) From Chaos to Complexity: Implications for Organizations. *Executive Development*, 7 (4), pp.16–17.

Lubell, J., Peak, R.S., Srinivasan, V. & Waterbury, S.C. (2004) STEP, XML, and UML: Complementary Technologies. In: *DETC 2004: ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Salt Lake City, Utah USA, Asme. Available from: <http://link.aip.org/link/ASMECP/v2004/i46970/p915/s1&Agg=doi>.

Macal, C.M. & North, M.J. (2010) Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4, pp.151–162.

Mainzer, K. (1996) *Thinking in Complexity: The Complex Dynamics of Matter, Mind, and Mankind*. Springer-Verlag.

Mandel, T. (2011) So You Want To Know What A System Is? [Internet]. Available from: <http://www.isss.org/primer/primer.htm> [Accessed 7 September 2011].

Manson, S.M. (2001) Simplifying complexity: a review of complexity theory. *Geoforum*, 32 (3), pp.405–414.

Mason, H. (2007) Enabling e-business. *ISO Focus, July-August*. Available from: <http://www.iso.org/iso/ru/iso-focus-index/iso-focus_2007/iso-focus_2007-07.htm>.

Mathew, G.E. (2002) *The State-of-the-Art Technologies and Practices for Enterprise Business Collaboration – An Overview*. Infosys Technologies.

Matlack, C. (2006) Airbus: First, Blame the Software. *BusinessWeek*. Available from: <http://www.businessweek.com/globalbiz/content/oct2006/gb20061005_846432.htm> [Accessed 5 September 2011].

Maturana, H. & Varela, F. (1980) *Autopoiesis and cognition*. D. Reidel Pub. Co. Available from: <http://www.jstor.org/stable/20127678>.

MBE project (2009) Model Based Enterprise: The next Generation of Defense Supply Chain [Internet]. Available from: <http://model-based-enterprise.org> [Accessed 29 January 2012].

McElroy, M.W. (2000) Integrating complexity theory, knowledge management and organizational learning. *Journal of Knowledge Management*, 4 (3), pp.195–203. Available from: <http://www.emeraldinsight.com/10.1108/13673270010377652>.

Mckelvey, B. (1999) Complexity Theory in Organization Science: Seizing the Promise or Becoming a Fad? Bottom-Up Organization Science. *Emergence*, 1 (1), pp.5–32.

Mentzer, J.T., DeWitt, W., Keebler, J.S., Min, S., Nix, N.W., Smith, C.D. & Zacharia, Z.G. (2001) Defining supply chain management. *Journal of Business logistics*, 22 (2), pp.1–26.

Merali, Y. (2006) Complexity and Information Systems: the emergent domain. *Journal of Information Technology*, 21, pp.216–228.

Merali, Y. & McKelvey, B. (2006) Using Complexity Science to effect a paradigm shift in Information Systems for the 21st century. *Journal of Information Technology*, 21 (4), pp.211–215.

Merriam-Webster (2011) Semantics [Internet]. Available from: <www.merriam-webster.com/dictionary/semantics> [Accessed 7 September 2011].

Mika, P. (2005) Ontologies Are Us: A unified model of social networks and semantics. In: *International Semantic Web Conference*. pp.522–536. Available from: <http://www.cs.vu.nl/~pmika/research/papers/ISWC-folksonomy.pdf>.

Miller, J.G. & Miller, J.L. (1982) The earth as a system. *Behavioral Science*, 27 (4), pp.302–322.

Missikoff, M., Schiappelli, F. & Taglino, F. (2003) A controlled language for semantic annotation and interoperability in e-business applications. In: *Proceedings of the Second International Semantic Web Conference (ISWC-03)*. Sanibel Island, Florida, pp.1–6.

Morjaria, M. & Santosa, F. (1996) Monitoring complex systems with causal networks. *Computational Science & Engineering*, 3 (4), pp.9–10.

Morris, C.G. (1992) *Academic Press Dictionary of Science and Technology*. 1st editio. Academic Press.

Mosterman, P.J. & Vangheluwe, H. (2004) Computer Automated Multi-Paradigm Modeling: An Introduction. *SIMULATION*, 80 (9), pp.433–450.

MSEE IP (2011) Manufacturing Service Ecosystem (FP7 284860) [Internet]. Available from: <http://www.msee-ip.eu/> [Accessed 8 May 2012].

MSEE Partners (2012) *Deliverable D1.1.1: Service concepts, models and method at CIM-PIM-PSM level*. MSEE IP (FP7 284860).

Munro, D.A. (1991) *Caring for the Earth: A Strategy for Sustainable Living*. Gland, Switzerland, Published in partnership by IUCN, UNEP, WWF.

Myers, G. (1979) *The Art of Software Testing*. John Wiley & Sons Inc.

Mylopoulos, J. (1998) Information modeling in the time of the revolution. *Information Systems*, 23 (3-4), pp.127–155.

Nagurney, A. (2006) *Supply chain network economics: dynamics of prices, flows and profits*. Edward Elgar Publishing, Inc.

Nallon, J. (2003) An Overview of AP-233: The Systems Engineering Data Exchange Standard based on STEP (ISO-10303). *PLM World Annual Conference*. Available from: <http://www.slideserve.com/Mercy/an-overview-of-ap-233-the-systems-engineering-data-exchange-standard-based-on-step-iso-10303> [Accessed 9 April 2012].

Narasimhan, V.. L. & Hector, A. (2004) A New Ontology for Software Agent Technology. In: *Intl. Conf. on Information Technology 2004 (ICIT)*. Istanbul, Turkey.

NATO, A.D.P. 34 (ADatP-34) (2003) *NATO C3 Technical Architecture (NC3TA), Version 4.0*. NATO ISSC Open Systems Working Group.

Nicolis, G. & Prigogine, I. (1989) *Exploring Complexity: An Introduction*. W.H. Freeman. Available from: <http://www.amazon.com/dp/0716718596>.

NIST (2011) Sustainable and Lifecycle Information-based Manufacturing Program [Internet]. Available from: <www.nist.gov/mel/msid/dpg/slim.cfm> [Accessed 7 September 2011].

Nonaka, I., Konno, N. & Toyama, R. (2001) Emergence of “Ba”. In: I. Nonaka & T. Nishiguchi eds. *Knowledge Emergence: Social, Technical, and Evolutionary Dimensions of Knowledge Creation*. Oxford University Press, pp.13–29.

Noori, H. & Lee, W.B. (2009) Dispersed Network Manufacturing: an Emerging Form of Collaboration Networks. In: R. Dekkers ed. *Dispersed Manufacturing Networks: Challenges for Research and Practice*. Springer.

Nuseibeh, B. & Easterbrook, S. (2000) Requirements Engineering: A Roadmap. In: *Proceedings of the Conference on the Future of Software Engineering (ISCE ’00)*. New York.

Nwana, H.S. (1996) Software Agents: An Overview. *Knowledge Engineering Review*, 11 (3), pp.1–40.

Ogren, I. (2000) On Principles for Model-Based Systems Engineering. *Systems Engineering Journal*, 3 (1), pp.38–49.

OMG (2003) *MDA Guide Version 1.0.1 (omg/2003-06-01)*. Object Management Group. Available from: <http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf> [Accessed 7 September 2011].

OMG (2008) Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification - Version 1.0 (formal/2008-04-03). Available from: <http://www.omg.org/spec/QVT/1.0/PDF/>.

OMG (2011a) Model Driven Architecture [Internet]. Available from: <http://www.omg.org/mda/> [Accessed 24 January 2012].

OMG (2010a) Object Constraint Language - version 2.2 (formal/2010-02-01). Available from: <http://www.omg.org/spec/OCL/2.2/PDF/>.

OMG (2011b) OMG Unified Modeling LanguageTM (OMG UML), Infrastructure - version 2.4.1. Available from: <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>.

OMG (2010b) Reference Metamodel for the EXPRESS Information Modeling Language Specification - version 1.0 (formal/2010-10-01). Available from: <http://www.omg.org/spec/EXPRESS/1.0/>.

OMG (2011c) Unified Modeling Language [Internet]. Available from: <http://www.uml.org/> [Accessed 24 January 2012].

Orr, G., Schraudolph, N. & Cummins, F. (1999) CS-449: Neural Networks [Internet]. Available from: <http://www.willamette.edu/~gorr/classes/cs449/ann-overview.html> [Accessed 8 September 2012].

Panetto, H. (2007) Towards a classification framework for interoperability of enterprise applications. *International Journal of Computer Integrated Manufacturing*, 20 (8), pp.727–740.

Panetto, H., Jardim-Goncalves, R. & Pereira, C. (2006) E-Manufacturing and Web-Based Technology for Intelligent Manufacturing and Networked Enterprise. *Journal of Intelligent Manufacturing*, 17 (6), pp.715–724. Available from: <http://oa.uninova.pt/1319/>.

Panetto, H. & Molina, A. (2008) Enterprise integration and interoperability in manufacturing systems: Trends and issues. *Computers in Industry*, 59 (7), pp.641–646. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0166361508000353> [Accessed 5 April 2011].

Papadimitriou, C.H. (1994) *Computational complexity*. Addison-Wesley.

Papazoglou, M.P. (2003) Service-oriented computing: Concepts, characteristics and directions. In: *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03)*. Roma, Italy, IEEE.

Park, J. & Ram, S. (2004) Information systems interoperability: What lies beneath? *ACM Transactions on Information Systems*, 22 (4), pp.595–632.

Park, K. & Willinger, W. (2005) *The Internet as a large-scale complex system*. Oxford University Press, USA.

PDES Inc. (2006) STEP success stories [Internet]. Available from: <pdesinc.aticorp.org/success_stories.html> [Accessed 5 September 2011].

Peppard, J. & Rylander, A. (2006) From Value Chain to Value Network::: Insights for Mobile Operators. *European Management Journal*, 24 (2-3), pp.128–141. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0263237306000156> [Accessed 2 May 2011].

Peristeras, V. & Tarabanis, K. (2006) The Connection, Communication , Consolidation, Collaboration Interoperability Framework (C4IF) For Information Systems Interoperability. *Interoperability in Business Information Systems*, 1 (1), pp.61–72.

Phelan, S.E. (1999) A note on the correspondence between complexity and systems theory. *Systemic Practice and Action Research*, 12 (3), pp.237–246. Available from: <http://www.springerlink.com/index/R5362G375621H218.pdf> [Accessed 3 May 2011].

Phelan, S.E. (2001) What Is Complexity Science, Really? *Emergence*, 3 (1), pp.120–136.

Pine, B.J. & Gilmore, J.H. (1999) *The experience economy: work is theatre & every business a stage*. Harvard Business Press.

Pratt, M. (2001) Introduction to ISO 10303—the STEP Standard for Product Data Exchange. *Journal of Computing and Information Science in Engineering*, 1 (1), pp.102–103.

Pustejovsky, J. (1993) *Semantics and the Lexicon*. Springer.

Ray, S.R. (2002) Interoperability Standards in the Semantic Web. *Journal of Computer and Information Science in Engineering*, 2 (1), pp.65–70.

Ray, S.R. & Jones, A.T. (2006) Manufacturing interoperability. *Journal of Intelligent Manufacturing*, 17 (6), pp.681–688. Available from: <http://www.springerlink.com/index/10.1007/s10845-006-0037-x> [Accessed 22 March 2011].

Rogerjiao, J., You, X. & Kumar, A. (2006) An agent-based framework for collaborative negotiation in the global manufacturing supply chain network. *Robotics and Computer-Integrated Manufacturing*, 22 (3), pp.239–255.

Rooke, J., Molloy, E., Sinclair, M., Koskela, L., Kagioglou, M., Siriwardena, M. & Siemieniuch, C. (2007) Models and Metaphors: Some Applications of Complexity Theory for Design, Construction and Property Management. In: *CIB World Building Congress 2007*. Cape Town, South Africa.

Rosendal, P. (2005) XML 1.1 [Internet]. Available from: <http://www.emn.fr/z-info/atlanmod/index.php/Atlantic> [Accessed 27 April 2012].

Russell, S.J. & Norvig, P. (2003) *Artificial Intelligence: A Modern Approach*. 2nd ed. New Jersey, Prentice Hall.

Saha, D. & Mukherjee, A. (2003) Pervasive computing: a paradigm for the 21st century. *Computer*, 36 (3), pp.25–31.

SAP Research (2009) Why “Internet of Services” [Internet]. Available from: <http://www.internet-of-services.com> [Accessed 27 January 2012].

Sarraipa, J. (2009) Tutorial “SEC – Semantic Enrichment of Product Catalogue” [Internet]. Available from: <http://gris-public.uninova.pt:8080/funStepServices/SEC.jsp> [Accessed 7 September 2011].

Sarraipa, J., Agostinho, C., Panetto, H. & Jardim-Goncalves, R. (2009) Semantic enrichment of standard-based electronic catalogues. In: *13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2009)*. Moscow, Russia. Available from: <http://oa.uninova.pt/1791/> [Accessed 2 May 2011].

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

Sarraipa, J., Vieira, H., Agostinho, C. & Goncalves, R. (2009) Methodology for Seamless Supply Chain Planning. In: *Proceedings of 1st International Conference on Software, Services & Semantic Technologies (S3T)*. Sofia, Bulgaria.

Schafersman, S.D. (1997) An Introduction to Science: Scientific Thinking and the Scientific Method [Internet]. Available from: <http://www.geo.sunysb.edu/esp/files/scientific-method.html> [Accessed 5 September 2010].

Schary, P. (2007) Supply chain management: the challenge of systems. In: D. Waters ed. *Global Logistics: New Directions in Supply Chain Management, ed.5*. Kogan Page Publishers.

Scheer, A.-W. & Schneider, K. (2006) Architecture of Integrated Information Systems (ARIS). In: P. Bernus, Kai Mertins, & G. Schmidt eds. *Handbook on Architectures of Information Systems*. Springer.

Schmidt, D. (2006) Guest Editor’s Introduction: Model-Driven Engineering. *Computer*, 39 (2), pp.25–31. Available from: <http://www.computer.org/portal/web/csdl/doi/10.1109/MC.2006.58>.

Schrage, M. (1990) *Shared Minds: The New Technologies of Collaboration*. Random House.

Schroth, C. & Janner, T. (2007) Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. *IT Professional*, 9 (3).

Schubert, L., Jeffery, K. & Neidecker-Lutz, B. (2010) *The Future of Cloud Computing: Opportunities for European Cloud Computing Beyond 2010 (Expert Group Report version 1.0)*. European Commision. Available from: <cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf> [Accessed 27 January 2012].

Seidewitz, E. (2003) What models mean. *IEEE Software*, 20 (5), pp.26–32. Available from: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1231147> [Accessed 3 May 2011].

Selic, B. (2003) The pragmatics of model-driven development. *IEEE Software*, 20 (5), pp.19–25. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1231146>.

Selik, M., Baraniuk, R. & Kruzick, S. (2010) System Classifications and Properties [Internet]. Available from: <http://cnx.org/content/m10084/latest/> [Accessed 7 September 2011].

Senge, M.P. (1994) *The Fifth Discipline: The Art and Practice of the Learning Organization*. London, UK, Random House.

Shannon, C.E. & Weaver, W. (1998) *The mathematical Theory of Communication*. The University of Illinois Press.

Shor, M. (2005) Dictionary of Game Theory Terms, Game Theory .net [Internet]. Available from: <http://www.gametheory.net/dictionary/Non-CooperativeGame.html> [Accessed 3 March 2011].

Simon, H.A. (1995) Near decomposability and complexity: How a mind resides in a brain. In: H. J. Morowitz & J. L. Singer eds. *The mind the brain and complex adaptive systems*. Addison-Wesley, pp.25–44.

Simon, H.A. (1962) The Architecture of Complexity. *Proceedings Of The American Philosophical Society*, 106 (6), pp.467–482. Available from: <http://www.jstor.org/stable/985254>.

Snodgrass, R. (1988) A relational approach to monitoring complex systems. *ACM Transactions on Computer Systems*, 6 (2), pp.157–195. Available from: <http://portal.acm.org/citation.cfm?doid=42186.42323>.

Software Engineering Institute (2002) *Software Technology Roadmap*. Carnegie Mellon University. Available from: <http://www.sei.cmu.edu/str/index.html>.

Sorensen, S. (2009) *The Sustainable Network: The Accidental Answer for a Troubled Planet*. O'Reilly Media.

Stergiou, C. & Siganos, D. (1996) Neural Networks [Internet]. Available from: <http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html> [Accessed 8 April 2012].

Sugihara, G. & May, R.M. (1990) Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 244 (6268), pp.734–741.

Suranaa, A., Kumarab, S., Greavesc, M. & Raghavan, U.N. (2005) Supply-chain networks: a complex adaptive systems perspective. *International Journal of Production Research*, 43 (20), pp.4235 – 4265.

Syed, A. & Shah, A. (2001) Data, Information, Knowledge, Wisdom: A Doubly Linked Chain? In: *International Conference on Information and Knowledge Engineering (ICIKE 2001)*.

The United States conference of mayors (2004) *Interoperability Survey (192-City Survey)*. Homeland Security Monitoring Center. Available from: <http://www.usmayors.org/72ndannualmeeting/interoperabilityreport_062804.pdf> [Accessed 5 September 2011].

Thomas, D. & Barry, B.M. (2003) Model Driven Development: The Case for Domain Oriented Programming. In: *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA'03)*. ACM Press.

Tolk, A. & Muguira, J.A. (2003) The Levels of Conceptual Interoperability Model. In: *2003 Fall Simulation Interoperability Workshop*. Orlando, Florida, USA, Simulation Interoperability Standards Organiza-tion.

Tretmans, J. (2001) *An Overview of OSI Conformance Testing*. Formal Methods & Tools Group University of Twente.

Truex, D.P., Baskerville, R. & Klein, H. (1999) Growing Systems in an Emergent Organization. *Communications of the ACM*, 42 (8), pp.117–123.

TTCN-3 (2011) Tree and Tabular Combined Notation 3 [Internet]. Available from: <www.ttcn-3.org/> [Accessed 2 May 2012].

Tukey, J.W. (1977) *Exploratory Data Analysis*. Addison-Wesley.

United Nations General Assembly (2005) 2005 World Summit Outcome (Resolution A/60/1).

University of Alberta (2011) Complex Systems [Internet]. Available from: <www.complexityandeducation.ualberta.ca/glossary/g_complexsys.htm> [Accessed 7 September 2011].

USA Secretary of Commerce (1993) Integration Definition for Function Modeling (IDEF0). Available from: <http://www.idef.com/pdf/idef0.pdf> [Accessed 23 April 2012].

Vaquero, L.M., Rodero-Merino, L., Caceres, J. & Lindner, M. (2009) A Break in the Clouds : Towards a Cloud Definition. *ACM SIGCOMM Computer Communication Review*, 39 (1), pp.50–55.

Vidgen, R. (2008) The NKC model [Internet]. Available from: <http://wiki.bath.ac.uk/display/sendero/NKCModel> [Accessed 3 March 2012].

W3C (2008) OWL 2 Web Ontology Language MOF-Based Metamodel. Available from: <http://www.w3.org/2007/OWL/wiki/MOF-Based_Metamodel#ref-owl-2-specification> [Accessed 26 April 2012].

W3C (2009) OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. Available from: <http://www.w3.org/TR/owl2-syntax/>.

W3C (2004) Web Services Architecture. Available from: <http://www.w3.org/TR/ws-arch/>.

W3C (2001) XML Schema (XSD). Available from: <http://www.w3.org/XML/Schema>.

Waldrop, M.M. (1992) *Complexity: the emerging science at the edge of order and chaos*. Simon & Schuster.

Wang, J.X. & Tang, M.X. (2006) A Multi-agent Framework for Collaborative Product Design. In: *Agent Computing and Multi-Agent Systems, 9th Pacific Rim International Workshop on Multi-Agents (PRIMA 2006)*. Guilin, China.

Wang, N.J., Quek, J., Rafacz, T.M. & Patel, S.J. (2004) Characterizing the effects of transient faults on a high-performance processor pipeline. In: *Proceedings of International Dependable Systems and Networks Conference*.

Wang, Wenguang, Tolk, A. & Wang, Weiping (2009) The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S. In: *Spring Simulation Multiconference (SpringSim'09)*. San Diego, CA, USA.

Watson, S. (2008) Material offshore sourcing undergoes standardization. *SCBIZ magazine (www.scbizmag.com)*.

Weiss, A. (2007) Computing in the clouds. *netWorker*, 11 (4), pp.16–25.

Weisstein, E.W. (2011) “Strange Attractor” From MathWorld--A Wolfram Web Resource. Available from: <http://mathworld.wolfram.com/StrangeAttractor.html > [Accessed 7 September 2011].

White, L. (1987) Software Testing and Verification. *Advances in Computers, Academic Press*, 26.

White, W.J., O'Connor, A.C. & Rowe, B.R. (2004) *Economic impact of inadequate infrastructure for supply chain integration (NIST Planning Report 04-02)*. Gaithersburg, MD, USA, NIST.

Wiener, N. (1965) *Cybernetics, Second Edition: or the Control and Communication in the Animal and the Machine*. MIT Press.

Wikipedia (2012) Interoperability [Internet]. Available from: <http://en.wikipedia.org/wiki/Interoperability#_note-0> [Accessed 20 January 2012].

Wilkinson, I. & Young, L. (2002) On cooperating: firms, relations and networks. *Journal of Business Research*, 55 (2), pp.123–132.

Wilson, E.O. (1999) *Consilience: The Unity of Knowledge*. New York, Vintage.

Wolfs, F.L.H. (1996) Introduction to the Scientific Method [Internet]. Available from: <http://teacher.pas.rochester.edu/phy_labs/appendixe/appendixe.html> [Accessed 5 September 2011].

Wooldridge, M. (2009) *An Introduction to MultiAgent Systems*. 2nd ed. John Wiley and Sons.

Wycisk, C., McKelvey, B. & Hülsmann, M. (2008) “Smart parts” supply networks as complex adaptive systems: analysis and implications. *International Journal of Physical Distribution & Logistics Management*, 38 (2), pp.108–125.

Yahia, E. (2011) Contribution à l’évaluation de l’interopérabilité sémantique entre Systemes d'Information d'Entreprises: Application aux systemes d'information de pilotage de la production (PhD Thesis). École Supérieure d’Informatique et Applications de Lorraine de l'Université Henri Poincaré, Nancy 1. Available from: <http://www.scd.uhp-nancy.fr/docnum/SCD_T_2011_0049_YAHIA.pdf>.

Yahia, E., Lezoche, M., Aubry, A. & Panetto, H. (2012) Semantics enactment for interoperability assessment in enterprise information systems. *Annual Reviews in Control*, 36 (1), pp.101–117. Available from: <http://www.sciencedirect.com/science/article/pii/S1367578812000090>.

Zelm, M. & Kosanke, K. (2007) Standardisation in Interoperability (Presentation at the IMS Workshop) [Internet]. Available from: <www.cimosa.de/Modelling/Standardisation_in_Interoperability.pdf> [Accessed 24 April 2011].