Carlos Eduardo Dias Coutinho

Licenciado em Engenharia Eletrotécnica, Sistemas e Comunicações

NEGOSEIO: Framework for the Sustainability of Model-oriented Enterprise Interoperability

Dissertação para obtenção do Grau de Doutor em
Engenharia Eletrotécnica e Computadores
Especialidade Sistemas de Informação Industriais

Orientador: Ricardo Luís Rosa Jardim Gonçalves,
Professor Auxiliar com Agregação, FCT/UNL

Co-orientadora: Adina-Georgeta Bratu Creţan,
Assistant Professor, "Nicolae Titulescu"
University of Bucharest, Romania

Júri:

Presidente: Prof. Doutor José Alberto Cardoso e Cunha

Arguentes: Prof. Doutor Hervé Panetto
Prof. Doutor Manuel Barata

Vogais: Prof. Doutor Adolfo Steiger-Garção
Prof. Doutor João Pedro Mendonça
Prof. Doutora Parisa Ghodous
Prof. Doutor António Grilo
Prof. Doutora Claudia-Melania Chituc

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro de 2012

Carlos Eduardo Dias Coutinho

Licentiate in Electrical Systems and Communications Engineering

NEGOSEIO: Framework for the Sustainability of Model-oriented Enterprise Interoperability

Dissertation to obtain the degree of Doctor of Philosophy in

Electrical and Computer Engineering

(Industrial Information Systems)

Supervisor: Ricardo Luís Rosa Jardim Gonçalves,
Auxiliary Professor with Aggregation, FCT/UNL

Co- supervisor: Adina-Georgeta Bratu Creţan,
Assistant Professor, "Nicolae Titulescu"
University of Bucharest, Romania

Evaluation Board:

President: Prof. Doutor José Alberto Cardoso e Cunha

Opponents: Prof. Doutor Hervé Panetto

Prof. Doutor Manuel Barata

Members: Prof. Doutor Adolfo Steiger-Garção

Prof. Doutor João Pedro Mendonça

Prof. Doutora Parisa Ghodous

Prof. Doutor António Grilo

Prof. Doutora Claudia-Melania Chituc

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September 2012

NEGOSEIO: Framework for the Sustainability of Model-oriented Enterprise Interoperability

To Linha, Vido & Nina,

for being the sunshine of my smile

(and for letting me stay awake at night)

ACKNOWLEDGEMENTS

After all those years of hard work and research, it is amazing to finally be able to make a brief pause to recollect what I humbly call “the course to illumination”. I recall with admiration all the places I’ve been to, the mixed feelings of fear, frustration, luck and gratitude, the growth from nothing into, well, something, and especially the opportunities and the incredible people I’ve had the privilege and honour to share thoughts with. I’m not sure to be able to pay credit to them all, but here’s an attempt.

First of all, my everlasting admiration to my supervisor, Professor Ricardo Jardim-Gonçalves, a whole section of this thesis wouldn’t be enough to praise your work. Thank you for being so stubborn, for trusting and believing in me against all odds, when my case seemed inevitably lost (so many times) “sustainably interoperating” through days and nights, pushing me to work harder, to reach farther.

Then my big “mulțumesc” to Professor Adina Creţan for your bright ideas, continuous support, counselling, and hard work. In spite of the physical distance, for always being there when I needed.

A word of deep appreciation to Professor Manuel Barata and Professor Steiger-Garção for inspiring me into starting this amazing journey, so many years ago. Also to journey mates, Carlos Agostinho, João Sarraipa, David Inácio, Vasco Gomes, José Lima, Pedro Magalhães & Tiago Santos.

I would like to acknowledge the support of the European Commission through the funding of the FP7 ENSEMBLE, UNITE, MSEE and IMAGINE projects, and the ESA-CDF for their enormous support and contribution in the development of the business case that was presented in this dissertation. I particularly wish to highlight the support of Massimo Bandecchi, Sam Gerené and Arne Matthyssen for so many brainstorming discussions about the ESA-CDF working environment, and to colleagues Ana Relvas, Arne Tøn, Tore Christiansen, Ray Richardson, and Jochen Haenisch (and many others!) for their fellowship and support on the OCDS project. Oh, and for teaching me how to ski. Also a thank you note to Hans-Peter de Koning, Carlos Corral and Tiago Soares for their expert judgement & time.

Moving into my inner circle, I wish to thank Mum Luz, for your restless help and encouragement through the various ups and downs of this hazard way. Also to my in-laws Domingos, Manuela, and Sónia for always being there when I needed the most (and for taking such great care of the kids). I’ll never thank you enough for your support, care, and especially, for the precious time you provided me.

To my family, parents, brothers, cousins, uncles… well, as a Goan I have a really large and tight family, and they were all supportive and patient towards my constant unavailability. A special word for my cousin Florêncio, who taught me how to learn, without whom, I would certainly not be here today. Also to another cousin, the Most Reverend Archbishop Blasco Collaço, who insisted in meeting me at the Vatican, near my thesis conclusion, and honoured me with his blessing to this research work.

Finally, and most especially, my wife Carla, and my loving children David and Catarina, please accept my eternal apologies for being absent and missing some football, karate and ballet exhibitions. You know I did my very best to be always present, despite always bringing my “appendix” laptop. This outcome is entirely **yours**.

I recall, from my dissertation defence, some words of wisdom from the jury: To my family, especially my little kids, as they managed to stand there through it all, for so many hours of discussion, after also standing there for so many years of work. Professor Steiger-Garção told them to understand it as a good example that life is not easy; perseverance and strength are essential to achieve our goals. Professor Parisa was surprised that I kept smiling throughout several hours of intense discussion and exchanging of ideas on the whole defence. When you are performing as a role model for your children, you should face all challenges with a smile on your face, like the song ("Smile" by Charles Chaplin). That's what I always taught them, so I couldn't do less.

Please keep on smiling; you make me the proudest and happiest man on Earth.

ABSTRACT

This dissertation tackles the problematic of Enterprise Interoperability in the current globally connected world. The evolution of the Information and Communication Technologies has endorsed the establishment of fast, secure and robust data exchanges, promoting the development of networked solutions. This allowed the specialisation of enterprises (particularly SMEs) and favoured the development of complex and heterogeneous provider systems. Enterprises are abandoning their self-centrism and working together on the development of more complete solutions. Entire business solutions are built integrating several enterprises (e.g., in supply chains, enterprise nesting) towards a common objective. Additionally, technologies, platforms, trends, standards and regulations keep evolving and demanding enterprises compliance. This evolution needs to be continuous, and is naturally followed by a constant update of each networked enterprise's interfaces, assets, methods and processes. This unstable environment of perpetual change is causing major concerns in both SMEs and customers as the current interoperability grounds are frail, easily leading to periods of downtime, where business is not possible. The pressure to restore interoperability rapidly often leads to patching and to the adoption of immature solutions, contributing to deteriorate even more the interoperable environment. This dissertation proposes the adoption of NEGOSEIO, a framework that tackles interoperability issues by developing strong model-based knowledge assets and promoting continuous improvement and adaptation for increasing the sustainability of interoperability on enterprise systems. It presents the research motivations and the developed framework's main blocks, which include model-based knowledge management, collaboration service-oriented architectures implemented over a cloud-based solution, and focusing particularly on its negotiation core mechanism to handle inconsistencies and solutions for the detected interoperability problems. It concludes by validating the research and the proposed framework, presenting its application in a real business case of aerospace mission design on the European Space Agency (ESA).

KEYWORDS

Enterprise Interoperability; Cloud; Services; Sustainability; Negotiation.

RESUMO

Esta dissertação aborda a problemática da Interoperabilidade em ambientes empresariais no cenário atual de um mundo interligado. A evolução das Tecnologias de Informação e Comunicação permitiu o estabelecimento de transmissões de dados rápidas, seguras e robustas, promovendo o desenvolvimento de soluções em rede. Isto permitiu a especialização das empresas (especialmente das PMEs) e levou à formação de redes de serviços complexas e heterogéneas. As empresas estão a deixar de ser auto-centradas, trabalhando em conjunto no desenvolvimento de soluções mais completas. Soluções de negócio completas são desenvolvidas integrando várias empresas (e.g., em cadeias de produção, e cooperação empresarial) com vista a um objetivo comum. Além disso, as tecnologias, plataformas, standards e regulamentos evoluem periodicamente, exigindo conformidade por parte das empresas. Esta evolução quer-se contínua, e está naturalmente associada a uma alteração constante das interfaces, conteúdos, métodos e processos de cada empresa. Este ambiente instável de perpétua mudança está a levar a grandes preocupações, quer por parte das PMEs como dos seus clientes, dado que o ambiente de interoperabilidade é frágil e pode facilmente levar a períodos de quebra, em que não é possível realizar-se negócio. A pressão pelo rápido restabelecimento da interoperabilidade leva frequentemente ao recurso a "remendos" e à adoção de soluções imaturas, contribuindo assim para uma deterioração ainda maior do ambiente de interoperabilidade. Esta dissertação propõe a adoção da plataforma NEGOSEIO, uma solução que lida com os problemas de interoperabilidade, desenvolvendo uma sólida base de conhecimento baseada em modelos, e promovendo a melhoria contínua e a adaptabilidade com o objetivo do aumento da sustentabilidade da interoperabilidade de sistemas empresariais. Ela apresenta as motivações para a pesquisa realizada, e os principais elementos da plataforma desenvolvida, incluindo um sistema de gestão de conhecimento baseado em modelos, arquiteturas orientadas a serviços colaborativos, realizada sobre uma solução baseada em tecnologias cloud, e principalmente focando-se na apresentação do seu mecanismo central de negociação para resolução de inconsistências e decisão das soluções a adotar para os problemas de interoperabilidade detetados. O documento conclui validando a pesquisa e a solução proposta através da sua aplicação no caso de negócio real do desenho de missões aeroespaciais na Agência Espacial Europeia (ESA).

PALAVRAS-CHAVE

Interoperabilidade Empresarial; Serviços; Nuvem; Sustentação; Negociação.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

GLOSSARY, DEFINITIONS AND ACRONYMS

- B-CIM: NEGOSEIO's Business CIM layer;
- B-PIM: NEGOSEIO's Business PIM layer;
- B-PSM: NEGOSEIO's Business PSM layer;
- BPEL: Business Process Execution Language;
- BPMN: Business Process Model and Notation;
- CDF: Concurrent Design Facility;
- CIM: Computation-Independent Model;
- CS: NEGOSEIO's Coordination Services;
- EI: Enterprise Interoperability;
- ESA: European Space Agency;
- FP7: European Community's 7th Framework Programme;
- IaaS: Infrastructure as a Service;
- IAMs: Metaphor Interaction Abstract Machines;
- ICT: Information and Communications Technologies;
- INCOSE: International Council on Systems Engineering;
- JADE: Java Agents DEveloping Framework;
- JESS: Java Expert System Shell programming language;
- KLOC: Thousand LOC;
- LOC: Lines Of Code;
- MAS: Multi-Agent System;
- MBSE: Model-Based Systems Engineering;
- MDA: Model-Driven Architecture;
- MDI: Model-Driven Interoperability;
- MDSE: Model-Driven Service Engineering;
- MENTOR: Methodology for Enterprise Reference Ontology development;
- MSEE: Manufacturing SErvice Ecosystem;
- NEGOSEIO: NEGOtiations for Sustainable Enterprise InterOperability framework;
- NM; NEGOSEIO's Negotiation Manager;
- OCDS: Open Concurrent Design Server project;
- OCL: Object Constraint Language;
- OWL: Web Ontology Language;
- Pareto Principle: Law of the vital few or 80/20 rule, states that in many situations it can be roughly observed that 20% of the causes are accountable for 80% of the effects;
- PIM: Platform-Independent Model;
- POC: Proof-Of-Concept;

- PSM: Platform-Specific Model;
- QoS: Quality of Service;
- SaaS: Software as a Service;
- SEI: Sustainable Enterprise Interoperability;
- SIM: System Independent Model;
- SLA: Service Level Agreement;
- SME: Small, Medium and Micro Enterprises;
- SOA: Service Oriented Architecture;
- SSME: Services Sciences, Management and Engineering;
- STEP: STandard for the Exchange of Product model data (ISO 10303);
- SWOT: Strengths, Weaknesses, Opportunities and Threats;
- TIM: Technology Independent Model;
- TSM: Technology Specific Model;
- UEML: Unified Enterprise Modelling Language;
- UML: Unified Modelling Language;
- USDL: Unified Service Description Language;
- VO: Virtual Organisation;
- WS: Web Services.

1. INTRODUCTION

"

If you start me up, if you start me up I'll never stop
If you start me up, if you start me up I'll never stop
I've been running hot, you got me ticking gonna blow my top
If you start me up, if you start me up I'll never stop…

"

Start Me Up – Jagger / Richards

This section introduces the dissertation and contextualises the developed work. It elaborates over the field of eliciting problems regarding interoperability, presenting the research environment and the business case which supported it, detailing the detected problems and setting them as the basis for a set of research questions and consequent hypotheses to be developed in the subsequent sections. The chapter concludes with the outlined structure of the remaining document.

1.1. Background Observation and Motivations

Competitiveness. Innovation. Pragmatism. Effectiveness. Proactivity. Improvement. Leadership. Excellence. Productivity. Environmental-friendliness. Quality. Efficiency. Collaboration. Sustainability. These are some of the buzzwords that can be found on the motto of every enterprise in the current days. This is a direct effect of the advent of having a globally available world. To survive, enterprises have two choices: Either taking a niche market, or going beyond market expectations. Either way, they must be present, available and they must interoperate.

Technology nowadays seems to surpass all concepts related with information exchange. The market demand changed a lot from the first information systems, when interoperation was determined just by being able to consistently exchange data. Since then, Information and Communication Technologies (ICT) have evolved from little more than wired data exchange to being capable of providing complex high-performing systems that already handle data reliability over a large bandwidth.

Thus, market needs are moving towards the next step and shifting to Enterprise Interoperability. Major breakthroughs have been achieved in the near past on this subject: data networking, the Internet and web-services have contributed to the collapse of the concept of enterprise isolation and paved the way for businesses to communicate and interoperate.

This quickly led to the growth of service-provider-consumer businesses and directed companies to specialise and improve their services, moved by the rising competition. The concept of service has been generalised, as all interactions (e.g., complex computer calculations, graphics design, but also carpet cleaning, wall painting, hardware assembly) were able to be virtualised as computer-based services that companies can evoke to perform their business. The globalised world has entered the Service era, where an enterprise produces and consumes services and goods using the internet, whether these are simple services that provide specific calculations to complex platforms like Customer Relationship Management (CRM) systems with distributed databases, to producing cars, houses, tools and other products. Virtualisation and service provisioning are common means of interoperate; however, this interoperability still relies on sets of agreed message syntax exchanges and agreements that are fragile and limited in their scope.

The Future Internet concept [1] shall no longer be concerned about data transaction speeds or data transfer sizes and communication data syntax, but with data reconciliation, service and semantics understanding. Interoperability needs to evolve to more than messages exchanging; it must rely on knowledge and share of the involved business models and semantics, in order to achieve a richer involvement between parties – one that is harder to forget and quicker to be regained, leading to a shorter downtime. Ultimately, this heightens the bonds, trust and confidence between the negotiating parties, leading to fewer errors in communication and concepts.

Emerging paradigms like the Internet of Things (IoT, which is reshaping the world in the form of categorized discoverable items) and the Internet of Services (IoS), together with the evolving cloud computing concepts are gradually transforming the existing reality into a set of available commoditised virtual objects, services, enterprises and networks.

Competition and high ICT availability have also driven enterprises to evolve by creating business nests where several enterprises gather their specialisations in the construction of complete and fully-featured solutions. These business organisations are particularly important when regarding SMEs, and tend to be very dynamic, with frequent inclusion of new enterprises. Strategic partnerships and outsourcing are common practices that frequently compete with in-house development.

The globalised market means companies face the availability of multiple possibilities in terms of suppliers and service providers, partners and customers, but also the growth of competition. Hence, to be able to be competitive, there is the need to continuously interoperate and seek new partnerships. The paradigm for each enterprise is shifting from being the most efficient to building the best-suited business integration. Business success quickly shifts its premises to become a sum of two factors, the SME's performance plus its partners/suppliers' performance. These enterprises thus develop dedicated business areas to handle the seeking for new partnerships for solutions that are aligned with the enterprise's strategy, with its inherent need for interoperation, while maintaining the interoperability regarding the current ones.

Competitiveness is also pressuring enterprises to build better solutions with fewer resources, following new trends and supporting new platforms and methodologies. Business models are recurrently more complex and detailed. Quality improvements and standards evolve more frequently, as well as concerns regarding accessibility, expansion, continuous improvement and upgrade, compliance to new technologies and support of new platforms, frameworks and development paradigms. Local and global legislation and regulations are updating more frequently and deeply, and demanding rapid compliance from enterprises.

This means a scenario of constant change in the interoperability environment. This move from enterprise-concentric to service-dispersed strategies, particularly on SMEs, is leading to concerns about reaching and maintaining the interoperability between customer-provider pairs and its impact on their associated business networks.

The increasing complexity of processes and interactions between businesses, companies, providers and customers means interoperability can no further be seen as a tacit and empiric science, but needs to be well established and founded, with detailed definitions, flows, strategies, procedures and interactions. Plus, the quality standards for service provisioning have risen, which means the price of losing interoperability in the current scope may mean the loss of business and credibility in an irreparable way. Strategies and decisions are increasingly more dependent on interoperability and so are also the investments and the whole business kernel.

These frequent business changes shake all the interoperability links between the enterprises, leading to periods of adaptation where business operation is not possible. The urge to rapidly regain interoperability often leads to unfounded, poorly-chosen solutions, which lead to inefficiency and rework.

This heterogeneity, constant change and subsequent need for interoperability are also worrying traditional business areas like finances (banking, insurance), aeronautics and aerospace, which usually tend to be very conservative towards change on account to accuracy and stability. As an example, the aerospace industry is served by a small set of large enterprises that implement projects and missions, and which then subcontract several Small and Medium Enterprises (SMEs) for supporting their development, thus creating a network of dependencies. The need for interoperability with the other players in these networks is as crucial for staying in business, as the ability to do so while maintaining each company's proprietary business assets protected from the competition.

To large contractors or even final customers like banks and insurance companies, who depend on the performance of this network of SMEs to conduct their business, the improvement coming from the specialisation needs to be balanced with the increase on control of the outcomes that result of multiple sources. The misunderstanding of a concept, a change in a data unit, a mistaken method on a single enterprise in the network can lead to chained mistakes on its counterparts and consequently to errors in the final result that are very difficult to detect and even more difficult to trace and resolve.

It is then essential that more than describing data and interface contracts, the interacting enterprises publish their models, ontologies and methods so that their partners can understand and cooperate with them easier. Moreover, it is important that a controlling entity (e.g., the prime contractor or the customer) is able to control if these models and concepts are aligned with the desired outcome.

This is particularly true in the aerospace segment – a competitive and demanding business supported by numerous applications, which vary from general-purposed, specific, proprietary or open-sourced, all needing to be interoperable and where the interaction between the various business elements should cause the smallest impact on the business itself. The evolution towards a Future Internet [2], [3] can only be possible if a sustainable interoperability between all parties is achieved.

This research describes common problems that are responsible for breaking interoperability in businesses among organisations acting in the same industrial market, and proposes solutions to manage and increase the sustainability of Enterprise Interoperability. It proposes to do so via the application of negotiations when evaluating interoperability solutions, with the objective of reducing the impact of changes in the interoperability of the enterprises and their surrounding environment. It proposes a framework that includes a negotiation mechanism for the management of enterprise-networked environment changes towards seamless business-to-business interactions.

Negotiation regarding the improvement of tasks and jobs implementation is therefore a task that despite not being a company's core business plays a crucial part on its business evolution. Negotiations often involve several heterogeneous parties, external and internal, using different tools, languages, platforms and ICT; several internal company areas compete or collaborate with external parties for the development of a project, or outsource parts of it.

Best practices [4] specify that decision analysis must convey a thorough documentation, evaluation and analysis of the alternative solutions in order to reach the best solution and learn from it, to allow the various parties involved in the negotiation need to be interoperable, sharing business, technical data and information seamlessly.

A strong motivation for dealing with these problems regards the author's large experience as a project manager in IT projects. Project management best practices [5] state that the most important assets in projects are predictability and meeting stakeholders' expectations. This subject is being extrapolated to the field of interoperability, stating that more important than establishing a quick interoperable link, it is better to focus on providing strong and durable interoperability, which will allow the pursuit of better solutions and hence avoid rework.

The proper way to handle this is by using simple project management activities like Risk Management, Scope Management, Time and Cost Management. If all these are applied to interoperability, then the expected result should be to have a proper and mature way to deal with new connections with the target of optimising the cost, effort and time to fulfil the demands of a well-determined scope.

Risk Management adds the capability of mitigation, i.e., the ability to monitor small changes in the interoperable environment and act accordingly with mitigation strategies with the purpose of reducing the probability and impact of interoperability changes.

1.2. Research Context

This research benefited from the knowledge and experience developed and acquired following the participation in the Open Concurrent Design Server (OCDS) project, an European consortium project which aimed to study and improve the interoperability and systems development of the Concurrent Design Facility (CDF) department of the European Space Agency (ESA).

This is an environment which involves the critical development of space missions, split in a multi-disciplinary set and where interoperability is a critical asset, as the entire business is based on the Concurrent Engineering methodology, which has as its base premises knowledge concentration and communication as its premises. The ESA-CDF business case is detailed on section 4.4.

The ESA-OCDS project [6] – which regards to the ECSS-E-TM-10-25 Technical memorandum "System Engineering - Engineering Design Model Data Exchange (CDF)" – outcomes are now being disclosed to other space agencies and the project is heading productisation towards as Open Concurrent Design Tool (OCDT).

1.3. Research Questions

Based on the state of the art on the actual research (section 2) and the problems described on section 1.1, the author formulated the following research question:

- **Can negotiations contribute to improve the sustainability of interoperability in enterprise collaboration?**

Based on this question, some sub-questions were elicited:

- Will the use of negotiations improve interoperability at the cost of destroying business secrecy?
- How can negotiations improve the establishment of a new interoperable ecosystem?
- How can negotiations reduce the time between the loss of interoperability and the reestablishment of the interoperable state?

1.4. Hypotheses

The stated questions lead to the formulation of a set of hypotheses to be developed:

- Sustainability assumes monitoring, control and improvement. The best way to control and improve a dynamic system and learn from it is by formalising the different alternatives. After that, clear criteria must fundament the selection of the best alternative. Applying negotiation processes for this matter formalises another dimension: the information flows and timings. This promotes interoperability solutions to maximise their performance, while capturing the knowledge for future improvements;
- Negotiations allow enterprises to have a clear definition of what is "public" knowledge that can (and should) be shared, and what is "private" knowledge that needs to be secured for the sake of business secrecy. This promotes a higher transparency of the offered solutions, methods and interfaces, favouring the collaboration of new and better partnerships;

- The establishment of negotiation processes from the very beginning of the definition of interoperability solutions favours early analysis of the requirements, best-practices and experience from the outcomes of previous negotiations help to determine if the interoperability is possible, and estimate the time, effort and cost it will take to make it happen. In addition of being able to develop better solutions, this will improve the predictability of the needs and resources to establish the interoperable ecosystem;
- The use of negotiations after the interoperability has been established is important to capture and prioritise the needs and requirements of each participant, which helps on the decision about determining if a determined change should be adopted by the participant and the consequences this decision may carry in an eventual loss of interoperability.
- Negotiating the solutions for re-establishing interoperability forces participants to consider alternative solutions, which favour the achievement of better solutions and thus to reduce the probability of rework.

1.5. Scientific Research Method

The current research was performed following the classical scientific method. This methodology, depicted on Table 1.1, defines several steps for the development of research to achieve the final goal of building a body of knowledge.

Table 1.1: Traditional Research Method steps

Revision / Redefinition

Step	Description	
1	Definition of Research Question / Problem	
2	Background / Observation	
3	Formulation of Hypothesis	
4	Design of Experiments	
5	Test and Validation of Hypothesis	
6	Analysis of Results	Publication of Findings
7	Acceptance by Stakeholders	

1.5.1. Definition of Research Question / Problem

This step concerns a declaration of interest, on stating that there is a science-based topic that the researcher wishes to address in a different way, or which may provide a different perspective. It states the researcher focus and determines the information that is required to be understood, hence, the scope of the research.

In this research, various thematic disciplines are being tackled, including but are not limited to:

- Model-Driven Engineering / Architectures / Interoperability;
- Model-Based Systems Engineering;
- Manufacturing Services;
- Enterprise Interoperability;
- Ontologies;
- Service-Oriented Architectures;
- Cloud Computing;
- Negotiation / Negotiation Strategies;
- Multiple-agent Systems.

1.5.2. Background / Observation

This step concerns the act of analysing the subjected topic on the field or in further research and documentation, to contextualise and fundament the research. Considering the current period of information flood, this step is essential to guide the researcher towards a focused path. In this research, the procedure performed for the concretisation of this step followed the approach proposed in the FInES project [7], which concerns the identification of the relevant authors using two tools for scientific publication analysis (Google Scholar [8] and Harzing's Publish or Perish tool [9]) to collect for each researched discipline:

- The five most cited papers;
- The five most interesting titles in the opinion of the researcher (different of the above);
- The five most recent publications (different from the above);
- The five best-ranked papers (different from the above).

The corresponding analysis resulted in a starting base of twenty papers per analysed discipline; for each, a brief analysis was performed and a classification assigned. Finally, the highest-quoted papers were analysed thoroughly. This created the first knowledge base, after which further analysis was performed by researching specific parts of these papers, related work and their references.

1.5.3. Formulation of Hypothesis

This step concerns the elaboration of an educated guess prediction to respond to the research question. It should be stated in a declarative form, focused and straight to the point. This hypothesis is the subject of the experiments and validation to be carried in the subsequent research. During this research work, several hypotheses have been formulated until the final one was determined.

1.5.4. Design of Experiments

This step aims to determine the testing artefacts that are required to be developed in order to be able to test the proposed hypothesis, always with the vision targeted in answering the research question. In this research this is performed in section 4.5.

1.5.5. Test and Validation of Hypothesis

This step concerns the actual realisation of the designed experiments to test the hypothesis. In this research, this step is described in section 5.1. The hypothesis should be validated and supported by a set of evidence artefacts (performed in section 5.3). There is a possibility that the observations of the artefacts reach a different conclusion than the one submitted as hypothesis, which may lead to the need to adapt, revise or redefine the hypothesis, or even reject it and define a new hypothesis.

1.5.6. Analysis of Results

The steps for result analysis regards the collection and interpretation of the data resulting from the testing and development of conclusions from them (in this research this step was performed on section 5.2), which may provide output findings for new research or to the extension of the current.

1.5.7. Acceptance by Stakeholders / Industry

The final step of the methodology regards the validation and acceptance of the conducted activities and outputs, and the recognition by the interested parts (stakeholders, scientific community) that the research is aligned to their expectations and that its outcomes provides valid conclusions that contribute to the increasing of the body of knowledge in the researched field. This step may include the productisation / industrialisation of the researched artefacts and outputs.

In this research, this step was realised by the elaboration of a report about the developed proof-of-concept which was analysed and validated by several experts of the analysed business case: The main responsible personnel from ESA-CDF, system engineers and other domain experts performed an analysis of the proof-of-concept and corresponding report, thus providing an essential feedback and peer validation over the subject, presented in section 5.4.

1.5.8. Publication of Findings

Finally, this step is executed alongside with the steps of the methodology regarding results analysis and acceptance by the stakeholders, and concerns the proper disclosure of the developed findings. While the research methodology should finish in the acceptance of the research by the industry, academia or other stakeholders, independently of the publication actions, the latter is essential towards providing feedback and peer validation.

In this research, a great care was taken in properly and periodically releasing the developed findings, not only in the scope of academic publications but also in presentations in international conferences. This resulted in the disclosure of its outcomes to a total of 8 scientific publications, including at least one in a relevant scientific journal (with SCI impact factor).

1.6. Dissertation Outline

Section 1 (this section) presents the introduction to this document, where its objectives are stated, together with a clear observation of the researched target and its context, the enumeration and analysis of problems, which then lead to the elaboration of research questions and to the formulation of hypotheses. Finally, an analysis of the research methodology is stated and correlated to the work developed in the other sections.

Section 2 presents the analysis of the main researched areas. This investigation comprises multiple aspects and analysis related to the theme of Enterprise Interoperability, including the multiple visions of interoperability, the literature review over several techniques used to improve it, and presents the author's own research contributions regarding each subject.

Section 3 presents one of the major focuses of this research, negotiation. It then performs a thorough analysis over the related literature and proceeds to define the contributions of this research in the field of negotiations, presenting the models, tools, infrastructure, rules, and processes.

Section 4 describes the proposed framework NEGOSEIO, detailing its motivations, premises, requirements, components and architecture, and describing thoroughly the methodology that supports its application in enterprises. It also introduces the context environment that will be the basis for the validation of the research questions and hypotheses, describing the real problems observed and reported on the business case, and detailing the implementation that was performed on the NEGOSEIO framework to adapt to the real business case, describing the solutions that were performed, and detailing the application of this solution in a proof-of-concept, including the validation scenario plans and test specifications.

Section 5 presents the testing activities and the outcomes that resulted from these tests in terms of consolidated analysis and in terms of the reporting results that were validated by the responsible personnel of the European Space Agency (ESA).

Section 6 presents a synthetic summary of the work performed during the research, highlighting the major contributions to science that were achieved and paving the way to the next challenges that are foreseen to be taken in the current research.

Section 7 presents the list of references to the reviewed literature which contributed for the outcomes of the performed research.

Annex A is an example of an interoperability questionnaire that was submitted to ESA, and whose answers were used to determine the problems and requirements on sections 4.1 and 4.4.1.

Annex B is a survey that was sent to ESA for peer technical validation of a report that was sent presenting the NEGOSEIO framework and regarding its applicability to the presented business case, and whose answers were consolidated in section 5.4.

2. RESEARCH AREAS & LITERATURE REVIEW

> "
> *I want to live, I want to give, I've been a miner for a heart of gold.*
> *It's these expressions I never give*
> *That keep me searching for a heart of gold, and I'm getting old.*
> *Keeps me searching for a heart of gold, and I'm getting old.*
> "

Heart of Gold – Neil Young

2.1. The Multiple Views of an Enterprise

To understand the intricate matters of an enterprise for interoperation, it is necessary to gather and capture the knowledge about it. Considering the enterprise business as a complex system, an interpretation of the General Systems Theory [10] outlines that the business activity of an enterprise can be considered a reflex of the conjunction of a wide set of areas. Hence, the outcome of a project cannot be linearly traced to a single cause. It is instead the result of many internal and external elements and processes.

Consequently, an enterprise (and its business) can be subject to analysis by multiple views, and the knowledge to be determined about it is the result of the added weighted contributions of all the science areas that influence it. While respecting this premise, it is not viable to determine or capture all the contributions of all the different science areas (e.g., hardware, software, tools, interfaces, but also business strategies, personal skills and interactions, and even others like beliefs, trends, and fortune). In the same scope, it can be extrapolated by the Pareto principle that 80% of the enterprise business is defined by 20% of these science areas, i.e., by focusing on its main contributions, it is possible to maximise the ratio between the obtained knowledge and the effort to obtain it.

2.1.1. Literature Review

This section presents the literature review on some of the analysed methodologies and architectures for gathering enterprise business knowledge in a holistic manner, i.e., capturing, inter-relating and consolidating information coming from multiple views and perspectives of a business.

2.1.1.1. Model-Driven Architectures

The term Model-Driven Architectures (MDA) was coined by the Object Management Group (OMG), and promotes the evolution of solutions through successive transformations of higher-level models into lower-level models, which eventually may result in going down to the level of code generation [11]. This represented a change of the undergoing paradigm that professed that system architectures are built by designing and maintaining its code. In this case, the changes are performed in the models, which are then transformed into code.

This means that interoperability may start from the very enterprise foundations, where it is easier to discuss business-related concepts and ideas, and then the progressive steps of transformation into lower-level models may also be synchronised to refine this interoperability, so that the overhead of transforming the concepts into code is performed by automation tools.

The development paradigm of MDA allows the definition of multiple levels of abstraction in the modelling of businesses, using descriptive languages and schemes e.g., UML, OCL [12], and UEML [13] to define the solution foundations. Applications should be designed right from a high-level abstract Computation Independent Model (CIM) where all business related functionalities, objectives, methods, context, requirements and definitions are specified regardless of any implementation (i.e., pure design).

Then, this model shall be subject to transformations into a more detailed Platform Independent Model (PIM), where the business concepts and rules are converted into activities, tasks, ontologies, structures and algorithms, although still independently of the underlying platform.

Finally, other vertical transformations and conversions shall turn the PIM into a Platform Specific Model (PSM), which provides the foundations for the development of the application, now targeted to a specific platform. Using the proposed framework, changes to any model (CIM, PIM) may trigger alterations in the other parties' models, which then, by transformation towards new PSMs, swiftly change the application towards compliance with the new model.

2.1.1.2. Model Based Systems Engineering (MBSE)

Model-Based Systems Engineering is a science initiative that has been evolving in the industry area, supported by the International Council on Systems Engineering (INCOSE) and the OMG, which envisions an object-oriented interpretation of a system. Traditional approaches towards the development of projects and systems determine the building of documentation to describe the evolution of the system: requirements, analysis, design, assembly, verification and validation. MBSE envisages the shift from this document-centric approach to a model-centric approach, based on an extension of the Unified Modelling Language (UML) known as OMG Systems Modelling Language (SysML [14]).

Over the last years, the Object Management Group (OMG) and other research entities have been developing standards for performing model-based engineering of systems, which allow the determination of models for each domain and for the whole system [15], [16]. The purpose is to use SysML to build models of systems which allow the specification for each system component, thus allowing the specialisation of the system and the exchange of systems information throughout the different systems domains and tools [17].

The use of these techniques in the studied business case aims to promote a clear understanding of the complexities in the ESA-CDF environment, in order to facilitate reuse and a proper understanding of the systems, one that is more semantically aware and oriented, able to define much more than the traditional hardware and software specifications. It needs to include information about e.g., organisations, personnel, processes and procedures, quality objectives, requirements, behaviours, hierarchies and structures (see Figure 2.1) [18].

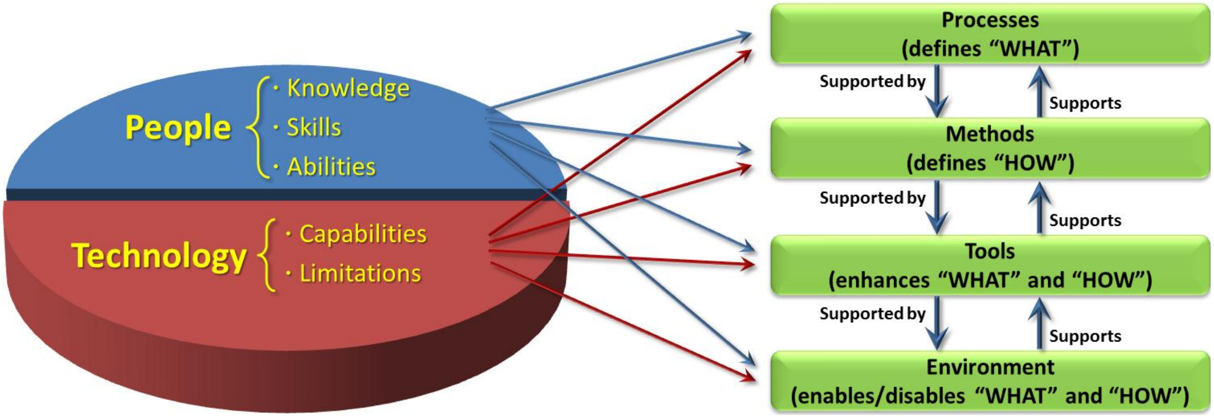


Figure 2.1: The influence of People and Technology in MBSE

Although its purpose is not to define abstraction levels, but to describe systems as a whole, the current trends in MBSE models are to narrow views with MDA models [19], thus defining multiple levels of abstraction in the design, description and development of systems, using SysML instead of UML, as can be seen on Figure 2.2. Within this progressive refinement methodology, a system is defined in a Systems-Independent Model (SIM), where basilar requirements and capabilities are specified, as well as enterprise environmental factors, business governance and similar. This SIM model is thus refined into a Technology-Independent Model (TIM), where a logical model of the system is developed, together with statistical and mathematical parametric models of the system behaviour [20]. Finally, the TIM model is transformed into a Technology-Specific Model (TSM), concerning implementation, automatic system documentation, code and other specific realisations.

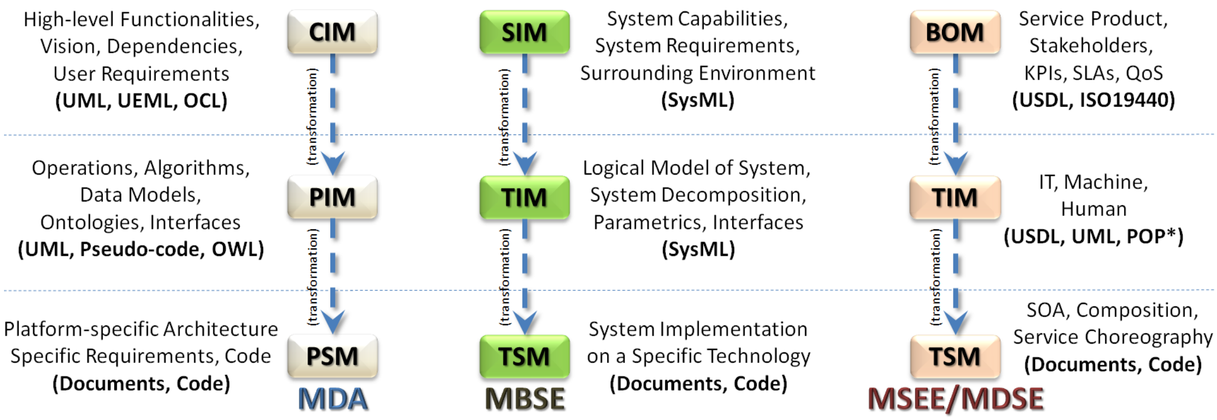


Figure 2.2: Relation between MDA, MBSE and MSEE/MDSE

While this standardisation is evolving in ESA (e.g., with the definition of standard information models e.g., ECSS-E-TM-10-25A [21], and the implementation of the Open Concurrent Design Server/Tool [22]), the transition from document-based engineering to model-based engineering needs to go beyond the simple architecture and specification. It must involve people, habits, training, customisation and progressive evolution; otherwise it shall remain a conceptual resolution that will not be able to be effective in the real-case utilisation. The challenge to build these models is as big as the difficulty of afterwards maintaining and evolving them.

2.1.1.3. Services Sciences, Management and Engineering

Traditional approaches for analysing and designing a business system focus mainly on its tangible areas or its direct action items, i.e., functional and non-functional requirements, architecture, interfaces, databases, specifications and regulations.

The constant evolution of the technology brought an increase of computing capability and storage, leading to a gradual servitisation of the world. This started with the development and availability of specific services for performing calculations and other processing tasks, and this concept has been generalised to the development of architectures based on services (Service Oriented Architectures, see section 2.3.1.3), and evolved to the servitisation of entire IT structures and departments, enterprises and environments. Hardware premises have been virtualised and made available as a commodity in the successful business concept of Cloud Computing (section 2.3.1.4) and eventually this virtualisation can be (and is being) generalised to a holistic vision where everything can be seen as a commodity and virtualised to a service. The initiatives towards the Internet of Things [23], [24], and the Internet of Services [25], [26] are major contributions towards a global availability.

The term Services Sciences, Management and Engineering (SSME) was coined by IBM to deal with this holistic approach, determining business to be the result of a set of services – the conjunction of people, technology, and organisations to create value [27], towards becoming very adaptive and flexible, reusable and commoditised. The SSME aims to improve the sustainability of the development processes, monitoring and controlling assets e.g., the quality, productivity and innovation of services and the exchange and widespread of services.

SSME vision states that to define a business, more than dealing only with its tangible assets (hardware, software, and related documentation) – hence Technology, businesses should also be analysed according to their processes, environment, procedures, quality standards, towards achieving the business optimisation that is needed for being competitive.

SSME also notes that an important asset of businesses is the human factor, i.e. the capabilities of its human resources and their interactions determine the agility and flexibility of a business. Issues like motivation, skills, team building and development, leadership, personal involvement and achievements are leading the priorities of enterprises.

All these aspects must be developed in the scope of a business vision and strategy, which itself can be analysed, studied and optimised by statistical methods and Ishikawa (cause-and-effect) diagrams and analysis towards the creation of servitised strategies that can be reused as business development frameworks. The overall SSME vision can be depicted on Figure 2.3.

Figure 2.3: The SSME vision

While the SSME group is still active [28], it is currently more focused on academic activities and initiatives, developing research towards information systems as a service and information systems-based services.

2.1.1.4. Manufacturing Service Ecosystem

Other initiatives have developed over the SSME concepts, like the Manufacturing SErvice Ecosystem (MSEE), a consortium project of the ICT Work Programme, of the European Community's 7th Framework Programme (FP7) [29]. MSEE targets to pave the way for service development in Europe, with the creation of virtual manufacturing factories (Factories of the Future) which shall make use of extended servitisation for the shift from product-centrism to product-based services, distributed in virtual organisations and ecosystems.

This project proposed a Model-Driven Service Engineering (MDSE) architecture, largely inspired in the concepts of MDA and SSME, which accounts enterprise services to be modelled into three major aspects (views): IT, Machine (and operation) and Human Resources (Figure 2.4). This modelling relies also on multiple abstraction layers with a progressive refinement. In this case, the abstractions are developed to define services.

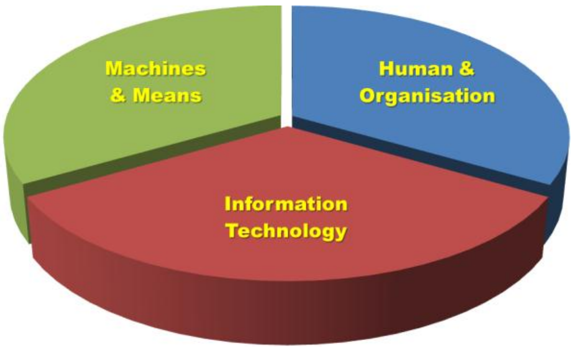


Figure 2.4: The MSEE/MDSE vision

The servitised products (Figure 2.5) are then modelled in a top abstraction layer regarding Business Oriented Models (BOM). This model describes the information about the service's product, the various stakeholders, service levels and indicators, Quality of Service (QoS), human aspects for organisation, hierarchies, standards and processes. The BOM model is then detailed in a Technology-Independent Model (TIM), which describes the three aspects (IT, Machine, Human) independently of the technology which will be used to realise the service. The TIM model is then specialised in a Technology-Specific Model (TSM) which realise the service in terms of technology, platform, specifications and documentation (Figure 2.2).

Figure 2.5: The servitisation of products (Product as a Service)

The MDSE models are developed using various specifications, e.g., the EN/ISO 19440 standard [30], the GRAI modelling language [31], the POP* language [32] and the Unified Service Description Language (USDL [25]).

2.1.2. Research Contribution

The literature review presents a set of initiatives to model an enterprise, which have in common the vision of defining their modelled subjects in highly abstract specifications which then perform a progressive specialisation (possibly with the aid of automated tools). While MDA is an established discipline that is already on the field for some time and is accurate to model technology/IT, the MBSE/SSME/SMEE views are still evolving and represent an attempt to extend this modelling to other areas, related to business, strategy and people.

This document thus proposes the vision that an enterprise collaborative environment is also split in two complementing views, as depicted in Figure 2.6. Reaching and maintaining interoperability means to capture knowledge about the businesses and detect changes that occur in the collaborative environment. In this quest, the perception should not only be accounting for changes that are needed in the technology, but also in business and people interaction. Business changes should actually be most accountable, as they usually convey deep-impacting changes on the supporting technology. Human behaviours, flows and interactions acting on the enterprise also play an important role in the interoperability that is required towards other enterprises.

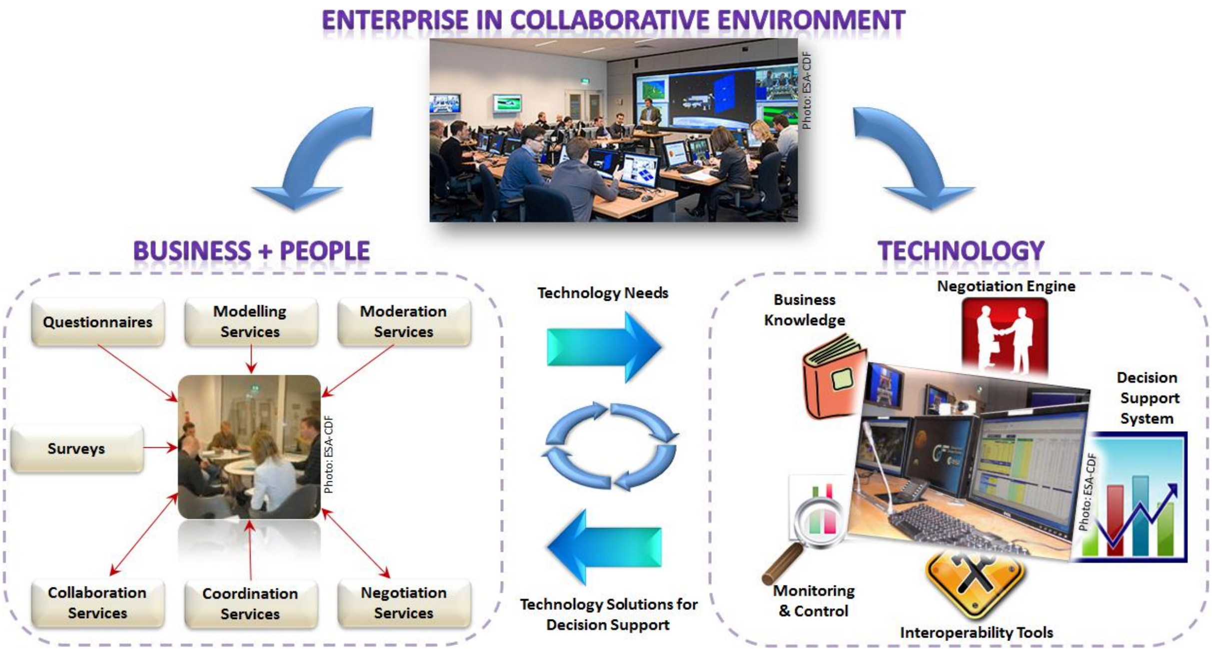


Figure 2.6: Collaborative environment views: Technology and Business + People

In this scenario, the Business + People view provides knowledge about strategies, business needs, quality requirements, concepts, hierarchies, human relationships, motivation, perception and understanding, decisions, directions and other intangible definitions. As these are tacit concepts, it is very difficult to extract specific conclusions directly from them. These then provide the core needs for the development and shall be captured and consolidated by the technology in the shape of information systems, ontologies (section 2.3.1.2), Enterprise Resource Planning systems, Customer Relationship Managers, documents, spreadsheets and other media, defined in the Technology view.

These systems then shall consolidate the tacit information and gather it with other data regarding functional requirements, architectures, statistics, mathematical statements, technology specifications and limitations, and hence provide solutions that are able to aid organisations to improve, whether by being able to respond to technical specifications and solicitations and also by providing the Business + People with solutions that consolidate the tacit knowledge and support decisions, which in turn will recursively become new needs to be fulfilled.

The proposed enterprise knowledge gathering activities comprise a holistic approach for modelling both the Technology knowledge into an MDA architecture and the Business + People knowledge into a Business-MDA (B-MDA) architecture. In this scope of view, Technology shall be abstracted into a MDA CIM, which will model system functionalities, user requirements, policies and constraints, and the Business + People view shall be abstracted into a B-MDA B-CIM, which will model organisational assets, organisation hierarchies, strategy, processes and quality requirements. The CIM shall be then transformed into PIM, modelling algorithms and other requirements, and the B-CIM into a B-PIM, modelling assignments, practices and workflows. Finally, the PIM shall be transformed to PSM, generating code, documentation and specifications, and the B-PIM to B-PSM, regarding documentation, training materials, directives and process activities. The overall picture can be described by Figure 2.7.

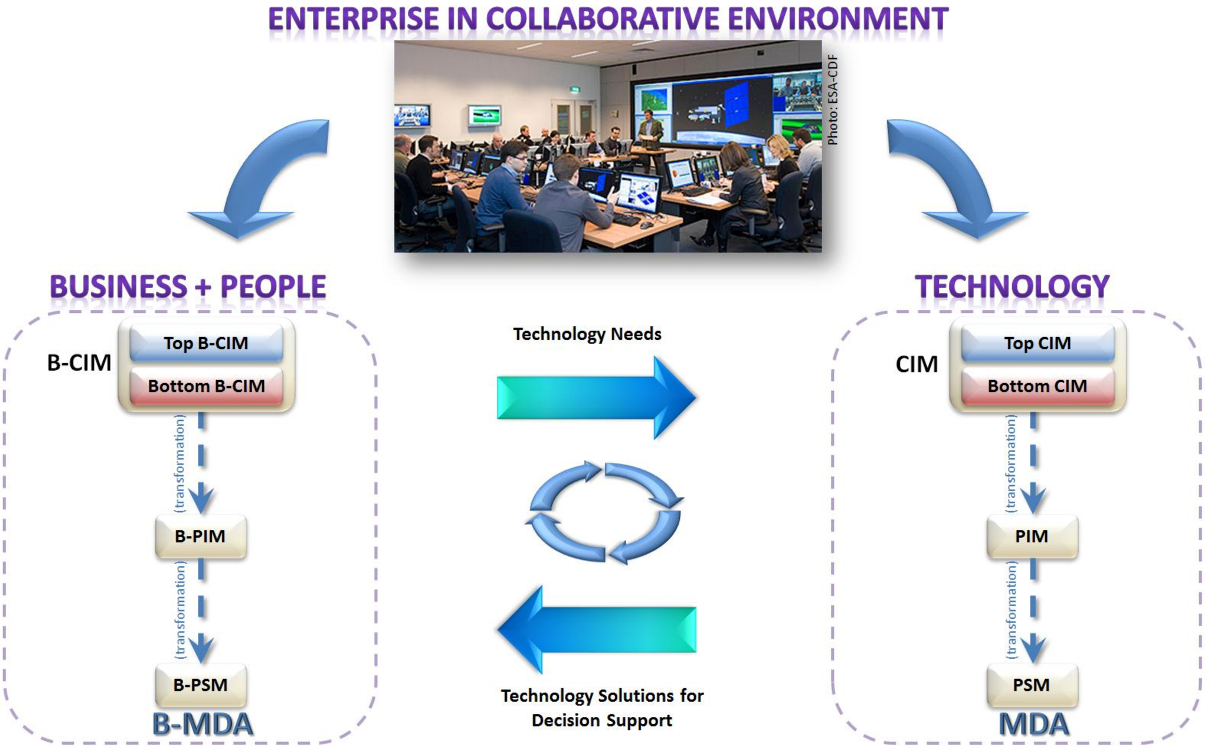


Figure 2.7: Modelling the Enterprise collaborative environment into MDA and B-MDA

Due to the complexity concerning the definition of the CIM and B-CIM models (regarding both the numerous issues to take into account on the modelling and the transformation of the CIM to PIM), the CIM and B-CIM layers were each split into two layers [33]: A Top-Level CIM (and B-CIM), which concerns strategies and abstract functionalities, where a top level specification can be achieved using enterprise modelling languages such as GRAI; and a Bottom-Level CIM (and B-CIM) which concerns system requirements, while still independent of any technology, which can be modelled using UML and thus becomes easier to be transformed to PIM (and B-PIM).

With the analysis of both views of an enterprise, it becomes clear that though they are different, both can benefit from servitisation, i.e., from the definition of services that when combined and properly orchestrated perform the desired functionalities. While the benefits of servitisation on the Technology view are often described on literature, little is said on the Business and People view. Servitisation in this context encompasses the adoption of standard behaviours, best-practices [4], better management and technical decisions, alternatives analysis and especially, predictability over the human factor, which is essential for decisions.

The gathering of knowledge and a proper modelling of this knowledge is an essential step towards understanding the needs and specificities of each business. This action has many applications for the future:

- It allows proper understanding of each own business, thus promoting control, improvements, better estimations, reuse, lessons-learned, statistic and trends, definitions of strategies, visions and new business opportunities. It also allows the perception of what is public data (which may be disclosed) and what is private data (which must be kept secure);

- It is essential for knowledge preservation, i.e., for the development of a corporate knowledge base which may also be used for the purposes stated on the previous item, but here more focused on knowledge transfer, business continuity, disaster recovery and resilience;
- It allows the proper definition and harmonisation of a common ontology for the different parties to have the same perception of the reality. Thus, it promotes a better understanding of each other's business which enables a stronger interoperation, trust and creates new opportunities for business;
- It allows the definition of intelligent agents (see section 2.5.1) which are able to detect (by monitoring e.g., data structures, values, sensors, and indicators) changes in the interoperable environment, and be trained to decide whether these changes are significant enough to endanger the interoperability between two parties.

2.2. Enterprise Interoperability

Interoperability is difficult. Not only because it is a kind of tongue-twister, but especially because the concept behind it is so broad, complex, and has many possible approaches and implications. While computerised interoperability exchanges can deal with "zillions" of bytes of information, it is still way behind the ability of the natural world to exchange information.

This is because the common conception of interoperating in the industrial world is still concerned about interfacing tangible assets, while nature and particularly, the business nature has several other aspects and abstractions that surpass this limited concept.

The term "interoperability" relates to:

- Communication, interconnection, interactions, data/information/knowledge exchange between two or more entities;
- Mutual understanding in the generation of ideas, negotiations, and establishment of agreements;
- According to section 2.1, it concerns a holistic approach regarding legal/regulation aspects, standards, and multiple-levelled abstraction of understanding:
    - At the technical level (e.g., data syntax, format, frequency, time, requirements, constraints);
    - At the conceptual level (e.g., semantics, taxonomies, ontologies, policies, operational);

    - At the organisational level (e.g., hierarchies, strategies, quality, business, motivations and intentions).
- The computerised intention to mimic human communications or other interactions found between real entities in nature.

IEEE defines Interoperability as “The capability of two or more networks, systems, devices, applications, or components to externally exchange and readily use information securely and effectively” [34].

Enterprise Interoperability (EI) concerns the seamless exchange of information that allows an enterprise to perform globally, whether that exchange of information is internal (between the various departments and parties that compose the enterprise), external (between the enterprise or part of it and an external party), or both. Additionally, EI denotes that the exchanged information is understood in the same way by all of them [35].

While this objective may be achieved by some large enterprises by setting the market standards and compelling the surrounding environment to comply with these standards, SMEs are much more sensible to the oscillations on the environment that involves them. They are the ones complying with the standards set by others, and living in an environment of changes to interoperate with them. Therefore, achieving a sustainable EI in this context entails adaptations in a constant and recursive effort to recheck the new interoperable status, while maintaining the existing interoperability towards the surrounding environment.

Considering that SMEs (Small, Medium and Micro Enterprises) are the backbone of the European enterprise economy tissue, accounting for more than 99% of the enterprises in the European Union [36], a major focus should be given to ensure these companies are most performant and interoperable.

EI concerns alignment or common understanding in various forms and levels, from middleware (e.g., signalling, interfaces, and platforms) to semantics (e.g., concepts, taxonomies, and relationships), workflows and methodologies, solutions and business visions. EI can be challenged either because a new enterprise which joined the ecosystem introduces new conditions, or because one of the existing enterprises has made a change in the existing interoperable environment [37].

2.2.1.Literature Review

Adaptive solutions for EI model real cases into mature reference models (e.g., the GRAI Model [31], [38]), resulting in the reference modelling of the real case. Despite most literature focuses on models and implementation of technological aspects (e.g., interfaces, adapters, translators) of the interoperation, business factors and human behaviour also have a major influence on the collaborative environment that should not be neglected.

As the number of enterprises in the ecosystem grows and interoperability becomes more complex, any change may result in the loss of the interoperable environment and therefore to a “downtime” period where there is no business. When trying to regain interoperability, e.g., due to a new concept, an enterprise may decide to establish a particular peer-to-peer interoperable hub (e.g., translator, adapter [39]) and isolate it from the others, or it may decide to adopt the new concept. The first option has the disadvantage of allowing the coexistence of different concepts in the same enterprise, which will end up in problems e.g., due to reuse and team rotation. The latter option has the danger of triggering a chain-reaction of changes in the network until all of them adopt the new concept, which may be desirable or not.

Enterprise Interoperability is acknowledged as an important factor that affects productivity, quality, time and cost of operations in either private and public sectors [40], demanding for stability and standardisation to be more focused on this area. Despite several research initiatives [41] and the fact that EI is a high-impact factor for the current and forthcoming business activities, the steps to establish a science base for EI are still starting.

Pataki et al. [42] state that the standard ISO14258 defines three forms of interoperability:

- **Integrated**, where a specific, static and restricted standard model is imposed for all interoperability issues; it has the advantage of clearly defining a model, with the cost of needing major effort in applying this model to each enterprise – it is very difficult to define a standard model that covers all interoperability needs, and even if one is defined, it is very hard to convince all players to comply to this standard;
- **Unified**, where a common meta-model is used on all players, which allows each of the players to adapt to the common model with transformations; while this is much more flexible than the integrated approach, it has the disadvantage of allowing interpretations of the common model, hence it is eventually prone to misunderstandings, so the transformations need to be careful and possibly mediated;
- **Federated**, where no player determines or imposes a model for interoperability. While this is indeed the most promising especially when dealing with SMEs, which need to be very flexible and adaptable to new concepts and realities, the somewhat anarchical approach leads each enterprise to define and make their own concepts and definitions, thus misunderstandings are frequent and may lead to rework.

2.2.2.Research Contribution

Interoperability needs to be more than exchanging business data. It must rely on knowledge and share of the involved business models and semantics. This increases the trust and confidence between the interacting parties, leading to fewer errors in the interoperable environment and promotes continuity in the current business relationship and in future projects.

The framework that is proposed on section 4 comprises a set of methods and tools for characterising and shaping enterprise businesses towards becoming interoperability-aware, for converting the business operations into a set of flexible cloud-based services, for tackling interoperability problems derived from the Technology and Business + People views of the Enterprise and for negotiating the changes that are needed to achieve and maintain interoperability with current and new players.

This research aims to achieve interoperability not by imposing a new interoperability standard model but by negotiating it with the existing interoperable partners and the new player towards reaching a better solution. This approach leads to improved solutions for interoperability, standards and best-practices that will benefit all players. While the Federated model seems the most promising to achieve this, it seems however to be too complex to be achieved and evolved, especially when dealing with a business case that requires some reliability in the obtained results (see section 4.4), which involves a large number of heterogeneous parties, and where rework is one of the major problems. So in this research, a unified interoperability architecture model is the one that best fits the intended purpose. A reference ontology shall be defined but it shall be adaptable to the needs of new partners, as a methodology for mediation and harmonisation of the incoming parties is suggested to reduce misalignments (see section 4.2).

2.3. Frameworks for Enterprise Interoperability

2.3.1. Literature Review

The following sub-sections present the literature review on some of the analysed frameworks promoting Enterprise Interoperability.

2.3.1.1. Model-Driven Interoperability

The Model-Driven Interoperability (MDI) concept derives from MDA (see section 2.1.1.1): it comprises the same abstraction layers, but in this case the target to be modelled is the interoperation between the involved parties. The idea behind MDI is to define models for each MDA level that allow the exchange of information. If the MDA can be described as a set of vertical transformations from a conceptual high-level model to a progressively detailed model, then MDI may be seen as a set of horizontal transformations to allow interoperability at each MDA level, e.g., Process, Product and Organisational models with the System Requirements at CIM level and transformations of these models into interoperability models.

Projects like the Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application (ATHENA) defined a framework that supports interoperability throughout the various abstraction levels and business aspects of enterprise software engineering [32], [43]. The ATHENA framework divides the technology view of an enterprise (section 2.1) into four main areas: Business, Knowledge, ICT Systems, and Semantics, and defines techniques to handle interoperability: Enterprise Modelling, Architectures and Platforms, and a reference Ontology.

Lemrabet et al. [44] provide simplified views over the MDI concept and the ATHENA Interoperability Framework (AIF) concepts and solutions on actions to develop each level of interoperability:

- Interviews, workshops and BPMN choreography diagrams for CIM levels;
- Diagrams, definition of business goals and BPMN collaboration diagrams for PIM levels;
- Service Oriented Architectures (SOA, see section 2.3.1.3) and BPEL implementations at PSM levels.

Chen et al. [45] define a roadmap on the possible approaches towards the development of enterprise architectures accounting interoperability.

Another important project in this area is the FUSION research project. It proposes a model-based approach with the ability of decoupling the physical representation of the business and being able to have progressive and further detailed design definitions across time [46], [47]. It has put focus on businesses, concentrating on the business model representation and not on technical implementations or platform-specific details.

The idea behind Model-Driven Interoperability is to use the MDA approach to issue problems related to interoperability, i.e., to define Process, Product and Organisational models with the System Requirements at CIM level and posterior transformations of these models into Interoperability models.

SOA4BIM is a framework that integrates MDA, SOA and Cloud technologies and an application scenario towards a Construction Business model [48]. Lopez-Sanz et al. [49] presents MIDAS, a basic SOA Model for PIM level, and details the components and interactions that need to be taken care when defining this level, and Zdun et al. [50] define a way of integrating SOA Models using Model-Driven Engineering.

2.3.1.2. Ontologies

Currently, the involved resources (e.g., documentation, data, units, methodologies, hierarchies, business flows) are often represented in heterogeneous formats. This makes the effective capture, retrieval, reuse, sharing, and exchange data and knowledge a critical issue in a networked enterprise environment.

According to Studer et al. [51] ontologies are “a formal, explicit specification of a shared conceptualization”, where:

- “conceptualization” refers to an abstract model of some part of the world which identifies the relevant concepts and relations between these concepts;
- “explicit” means that the type of concepts, the relations between the concepts, and the constraints on their usage, are explicitly defined;
- “formal” refers to the fact that the ontology should be machine readable;

- "shared" means that the ontology should reflect the understanding of a community and should not be restricted to the comprehension of some individuals.

Hence, the basic purpose of ontologies is to capture consensual knowledge. In order to properly address all issues regarding the ontological terms of the involved business and semantic interoperability, the development of one or more ontologies is essential to permit a correct and unambiguous set of business definitions and concepts.

The heterogeneity involved of the process to gather knowledge requires the adoption of ontologies to assist in the activities required to achieve and sustain the collaborative enterprise environment. In order to manage enterprise knowledge in a manner that is explicit, formal, modular, extensible, interoperable, and yet comprehensible, it is more appropriate to use an ontology-based modelling approach to collaborative design. Besides, the use of ontology modelling for EI has been proving to be a prominent approach to assure a seamless collaborative enterprise interoperability [52].

Ontologies have been realized as the key technology to shape and exploit information for the effective management of knowledge and for the evolution of the Semantic Web and its applications [53]. In such a distributed setting, ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge.

There are generally three approaches for combining such distributed heterogeneous ontologies:

- Ontology inclusion, in which the source ontology is simply included within the target ontology;
- Ontology merging, using mediators;
- Reasoning on ontologies;
- Ontology mapping, in which a part of the source ontology is related to the target ontology's entities [52].

The solution of using a common reference ontology, described in section 2.3.2, has been approached in many works, such as Torres [54]. Albeit beneficial in many ways, the use of a common ontology becomes much more complex when dealing with multiple application fields regarding creation, update and efficient structuration.

To provide supportive environments for collaboration, systems must provide facilities for sharing information, coordinating tasks, and solving conflicts towards a global interoperable collaborative enterprise environment [55]. Such environment can use Web Services (WS, see section 2.3.1.3) that are suitable to be used with ontologies. Enriched WS with expressive and computer interpretable languages will help capturing the semantics of WS, with ontologies functioning like metadata schema with a rich descriptive capacity, facilitating the exchange of syntactic and semantic information among people and computers [53]. Thus, integrating ontologies into WS not only enhances the quality and robustness of service discovery and invocation but also paves the way for automated composition and seamless interoperation.

2.3.1.3. Service-Oriented Architectures

In the last decade, Service-Oriented Architectures have contributed to an extraordinary improvement towards interoperability. Services (and their subset Web-Services in particular) have reshaped the existing concepts of solution deployment and provisioning, and paved the way for other important concepts using the same paradigm, like functional discovery and subscription in common repositories, orchestration and composition of services into more complex ones [56].

Following the most advanced technological trends, service-based SOA shall be used to implement the platforms for interoperability between the involved parties. SOA are very flexible and rich architectures that provide the needed balance between the capacities of performing very complex functionalities, to being able to perform service compositions if needed, to the wide-range implementation universe [57].

The adoption of SOA does not directly respond to any interoperability requirements, though it may be seen as the foundation backbone, for the implementation of the set of services that shall address all interoperability issues, and that can be composed and reused. The SOA4BIM [48] framework also integrates MDA, SOA and Cloud technologies in an application scenario towards a Construction Business model, while MIDAS [49] presents a basic SOA Model for PIM level, and details the components and interactions that need to be taken care when defining this level. Zdun [50] defines a way of integrating SOA Models using Model-Driven Engineering.

While SOA and Service Composition [58] are concepts that have been commonly adopted by several implementations, with Web Services (WS) still being the standard for implementing SOA [59], they still face several research challenges that need to be met, e.g., in terms of semantic service discovery, additional complementary data should be defined by the service providers on semantic annotations to describe the services and its Quality Of Service (QoS) characteristics [60], and service discovery on the client side should use these, with the help of common ontologies, to reduce the involvement of users in service discovery and selection.

Dynamic service connectivity and dynamic configuration of the infrastructures are also aspects that still have a large margin for improvement. Services should also be able to adapt and optimise in response to external environment changes (e.g., demands, QoS). The Globus toolkit [61] is useful a set of computer tools and applications for building distributed SOA applications, while Tsai et al. [62] propose a Service-Oriented SaaS Cloud Computing infrastructure, and Menascé [63] proposes a framework for selecting services optimally in SOA.

2.3.1.4. Cloud Computing

Web Services have the ability to be reused and composed into more complex services, but they must proceed towards distributed computing to handle scalability issues in solutions deployment. Cloud Computing is not actually a new concept or major breakthrough in terms of technology but rather a business concept towards the idea of distributed computing.

Cloud-based solutions or Cloud Computing are not actually a new concept or major breakthrough in terms of technology, but rather a business concept towards the idea of distributed computing existing for quite some time. Cloud-based solutions may be roughly split into the concepts of on-demand storage and server availability (Infrastructure as a Service or IaaS), on-demand platform integration (Platform as a Service or PaaS) and on-demand processing availability (Software as a Service or SaaS) [64].

This concept also boosted interoperability, with a special favouring for SMEs, whereas solutions can now be developed in small, inexpensive proof-of-concepts, and if proven correct, rapidly be scaled into large solutions, reducing dramatically time-to-market and allowing companies to be able to plan peak workloads without the burden of keeping infrastructures on lower workloads thus enhancing agility and flexibility for businesses. Sharma & Sood [65] label SaaS as the natural evolution of grid computing and SOA. Interoperability using services in a cloud-based environment ensures flexibility towards changes due to e.g., new requirements, semantic heterogeneity, thus contributes to sustainable interoperability.

Just like SOA, Cloud Computing is a technological trend that is a breakthrough change. Though it can still be considered to be on its ground steps, the existing implementation is already a successful business concept that has already been widely spread, supported by major hosting players, e.g., Amazon, Google, Force.com, Microsoft [66], making use of the evolution of ICT to host business services that offer the promise of virtually infinite on-demand storage and processing capabilities, multi-tenancy, scalability, essentially a tremendous infrastructure cost reduction.

Actually, the concept’s early success may be considered [67] as the main cause for delay to further research on the area, where some projects are still striving and achieving success, like the OpenNebula project [68], which is trying to define a generic Cloud-supported infrastructure, and Reservoir [69] which explores the possibilities for the existence of a federation of clouds that allows the selection of the most appropriated cloud for each demand.

Cloud Computing premises also account concerns like environmental impact and energy saving (“Green Cloud Computing”) [70]. Subashini [71] presents a survey on the main problems and issues regarding the concern for security over Clouds. Cloud Computing are especially important towards a proper interoperation between parties due to its ability to handle scalability of the solutions.

Moreno-Vozmediano et al. [72] highlight that despite the various efforts to advance the cloud computing research in the recent years, numerous problems and concerns are still without a solution and being analysed, e.g., the dynamic provision of the cloud services, the inclusion of negotiations for Quality of Service (QoS) and Service Level Agreements (SLA), an automatic scalability in service provisioning, context-aware applications, multi-provisioning and portability, and of course, the ongoing concerns about security and privacy of the information stored in the cloud, as with legal issues.

2.3.2.Research Contribution

The framework that is proposed for improving the sustainability of the interoperability (see section 3.7) includes the definition of a model for interoperability (MDI), but the horizontal transformations that are performed for each MDA level include an additional step which is the negotiation of the interoperability solutions.

It is important to ensure the correctness and precision of the contents, so that different models can be integrated. Hence, the first steps to be taken in the proposed methodology (see section 4.2) are the formalization and the harmonisation processes for semantics. In order to support the integration of different kinds of knowledge representation into the proposed architecture, the harmonisation method is proposed. The final goal of this harmonisation is to provide a reference ontology to be used by all enterprise systems and applications. The harmonisation method is a stage undertaken before and after defining the generic ontology, with the purpose of combining the knowledge of different formalisms to improve the representational adequacy and deductive power. Again, it is important that the harmonisation of ontologies include an additional step which concerns the negotiation of what is to be harmonised and how, in order to achieve stability and consistency.

Then the proposed framework also determines that the modelling of the knowledge is determined by MDA, MDI, B-MDA and B-MDI towards a service-oriented architecture (SOA) which should be deployed as SaaS cloud services. One contribution on this aspect is performed on the provision of the dynamic services that shall perform the interoperability activities, where negotiations are proposed to perform this provision, hence determining the best combination services to be selected according to the set of needs, parameters and requirements. This includes the determination of the context of the needed solutions, hence providing means for context-awareness of the deployed solutions.

This negotiation should also be included on the determination of the cloud provider or the combination of providers that are available to supply their services, and agents are proposed on the set of cloud management services to analyse and detect changes and early signs of an increase (or decrease) on the service and resources demands, hence providing a way to request automatically the upscaling or downscaling of the provided cloud base.

2.4. Sustainable Interoperability

If interoperability as defined on section 2.2 is a tongue-twister, then the hard-to-say “interoperability sustainability” is a double tongue-twister. While interoperability may be considered a technical aspect, sustainability is a quality, an adjective that is frequently associated with resources, ecologic and economic aspects, and that is defined as the capacity to endure [73] by means of a responsible use of resources.

A business is said to be sustainable when the resources it needs to operate are and will be available in the needed quantity (even considering the business evolution) throughout the course of time, or a business may be considered “more sustainable” than another if it has a more responsible and frugal resource consumption.

As an example, it can be said that the renewable energy generation (e.g., solar, wind, wave, geothermal, tidal) can be considered sustainable, whereas the oil or gas prospection businesses are not sustainable, as the time it takes to replenish the needed resources is longer than its consumption. Sustainable then means that the needed resources are unlimited (which are the renewable energy examples) or that the needed resources are produced in a faster pace than their consumption throughout time. Time, future and business evolutions are necessary environment variables.

While recognising this close connection nature and ecosystems, sustainable interoperability has been defined as “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” [74], which means that a sustainable interoperability is one that is open to evolve while being able to adapt to present and future changes. The concept of a sustainable interoperability is closely linked to the actors who implement that interoperability, in this case the actors for SEI are Complex Adaptive Systems (CAS), which have a set of properties which makes them prone to be used for this purpose: They are autonomous, self-organised, they evolve, interact and are heterogeneous. Hence, they are flexible, adaptable and dynamically configurable and especially have context-awareness (they are conscious of their surrounding environment and are able to detect changes and react accordingly), which are considered core characteristics for interoperability.

Accordingly, a Sustainable Interoperability Framework should have as its grounds four major layer components:

- A monitoring system which is able to detect changes in the interoperable ecosystem;
- An Integration and Interoperability Intelligence component, targeted to perform the learning and adaptation to execute maintenance activities towards restoring the interoperable environment, eventually using technologies like MDA and MDI;
- A decision support system to support the decision to enrol or not into a harmonisation and reintegration process;
- A communication layer with the purpose to ensure the proper harmonisation and cooperation between the various nodes of the network.

Ducq et al. affirm that a sustainable interoperability can only be achieved through the continuous management of enterprise interoperability [75]. They defend the existence of the concept “sustainable interoperability” due to the fact that if it is logic to state that some systems have a higher potential in terms of interoperability than others, then it must be possible to define a set of structural characteristics or resources for comparing interoperability, and hence, the possibility of having sustainability is a reality.

Khouri et al. [76] defend that if the definition of requirements, together with their connection to semantics is represented using formal expressions and is stored into dedicated knowledge bases, then this promotes the creation of systems capable to inferring and deducing changes and readjustments due to requirement changes, thus contributing to facilitate a sustainable enterprise interoperability.

Jardim-Goncalves et al. [77] define Sustainable Interoperability as being related to the aim of improving the quality of service, by contributing to a more robust interoperability. In this sense, they tackle the problem regarding the resource consumption by stating the resources in this context have to deal with concepts like man-power or time, and with the valid concept that the purpose is to maximise efficiency by maintaining itself interoperable throughout time, according to its natural evolution and changes required by the surrounding environment.

2.4.1.Literature Review

Jardim-Goncalves et al. [78] define a set of characteristics for sustainable interoperability frameworks, which are:

- The capability of determining when the interoperable environment has changed, provoking a break in the interoperable harmonisation;
- Adaptability to learn from previous harmonisation breaks towards triggering automatic maintenance processes which help solving the problem;
- Transient analysis, i.e., the capability to capture other environment indicators and extrapolate how the system shall behave during the period of no interoperability and how to compensate it;
- Notifications along the network, specifying the expected behaviour of each interoperable node in a way to be aware of adaptability needs.

Literature refers several examples of methodologies to enhance interoperability, most focusing on the development of adapters[17], translators, even also using MDI [79]. While this approach is valid as it pertains acting solely in translating the interaction between two different entities, there are times the changes are too many and too profound, leading to the inability of maintaining interoperability . If the changes due to the establishment of a new partnership or due to internal improvement have a very high impact, it is likely that model transformations will not be enough to maintain interoperability, leading to the need for interoperability negotiations.

Ducq et al. [75] identify a set of system properties that should provide the basis for interoperability, which are openness, adaptability, flexibility, re-configurability and modularity. Based on these premises, a method of evolution management is proposed towards reaching sustainable interoperability, which consists in the continuous development of small tasks “projects” to maintain interoperability, concluding that sustainable interoperability can be reached by managing the system evolution using modelling, identification of objectives and performance evaluation throughout the collaborative life cycle of the systems.

Khouri et al. [76] targets sustainability from the point of view of requirement changes. They state that changes may occur at two levels:

- Within content changes, i.e., changes in the recorded requirements themselves, including the creation, update and deletion of requirement records and/or data;
- Within the schema of the data, i.e., the modification of the data structures and their attributes.

They propose the evolution and versioning of the database schemas, and the use of OntoDB exchange format to facilitate a sustainable interoperability upon changes and evolutions in the data model.

An Enterprise Interoperability Science Base was proposed by [77], which includes multiple references and artefacts focused specifically on the improvement of the sustainability of interoperability:

- Reference ICT infrastructures and architectures, methodologies and tools;
- EI dynamics and transient analysis, networked enterprise systems management, behaviour monitoring and adaptability strategies, collaboration and coordination facilities, impact of changing requirements and information models in interoperable environments, and interoperability of digital ecosystems as ‘‘complex systems of systems’’;
- Case studies concerning applicability of the sustainable interoperability in enterprises and applications.

2.4.2.Research Contribution

Sustainability has a number of associated concepts, which may be elicited:

- Evolution, or the aptitude to embrace new challenges;
- Robustness, i.e., the ability to resist to small changes;
- Resilience or stability;
- Communication in terms of reporting and interoperability;
- Innovation i.e., the improvement of processes;
- Adaptability or Flexibility, the capacity to cope with the surrounding environment;
- Generosity, a very important and natural capacity, which means the willingness to advance, to move, to change.

Thus, it may sound strange to address the common buzzword “sustainable interoperability”; however, this is not so, as one can consider interoperability as a business itself which has resources consumption. While it is not easy to determine the whole set of resources needed for a set of entities to interoperate (as these may vary immensely), there are some that are inherent to the act of interoperating and specific to this business. Here are some measurable examples:

- **Indicator#1**: In a new interoperability link, the amount of time between the decision to interoperate and the establishment of the new interoperable connection;
- **Indicator#2**: In a new interoperability link, the cost, effort, and other related measures which are necessary to establish the interoperability link;
- **Indicator#3**: The complexity of the interoperability in terms of the level of understanding that is needed to be able to interoperate (see section 5.1.1);
- **Indicator#4**: In an already established interoperation, the cost, effort, and other related measures to maintain the interoperable environment;
- **Indicator#5**: The number of changes per unit of time that, if adopted by one of the participants, affect the established interoperable environment;
- **Indicator#6**: The complexity of a change that if adopted by one of the participants, affect the established interoperable environment (see section 5.1.1);
- **Indicator#7**: The ratio between the number of adopted changes and the number of opportunities to change that are discarded due to the fright of losing interoperability;
- **Indicator#8**: The period of time between the establishment of an interoperation environment and the loss of that interoperable environment, which may be called “Uptime”;
- **Indicator#9**: The period of time between the loss of the interoperable environment and the reestablishment of an interoperable environment, which may be called “Downtime”;

Hence, the sustainability of the interoperability may be defined as the ability of an interoperable ecosystem to handle these resources, in terms of maximising the Uptime and reducing the other referred periods of time, and reducing the costs and effort. There is currently no unique measurement unit to define it, thus this sustainability should be faced as a qualitative measure.

Although all of the above indicators are important to determine the level of sustainability of an interoperable ecosystem, there is one that needs to be highlighted, which is Indicator#3. The definition of sustainability determines it as endurance, i.e., the stability through the passage of time.

Section 1.1 indicates that the current interoperability environments are still simple and most focused in connectivity, data formats and syntax problems. Hence, interoperability is relatively simple and rapid to be obtained, and in the case of the loss of interoperability, its simplicity makes the reestablishment very quick. Even in a world facing numerous and rapid changes that affect interoperability (Indicator#5), and numerous partners in supply chains and Virtual Organisations, the rapid recovery from changes in the interoperability makes the Uptime significantly higher than the Downtime.

The focus on Indicator#3 concerns the shift in the interoperability paradigm, from interfacing and syntax to semantics, processes, workflows and knowledge. This shift will inevitably increase the complexity it the relationships between enterprises (and consequently, Indicator#3). Figure 5.4 shows the results of the observation of a proof-of-concepts which revealed that a linear growth of the interoperability complexity corresponds to an exponential growth of the reestablishment period.

Thus, a linear increase of Indicator#3 is expected to provoke Indicators #1, #2, and especially #9 to grow geometrically. Therefore, the uptime is continuously challenged by Indicator #5, and if the Downtime grows geometrically, it is expected that in the future, the Downtime will overrun the Uptime, thus enterprises seeking viability will need to reduce Indicator #7, i.e., they will stop adopting new changes and complying to regulations, due to the fear of losing interoperability and consequently, of running out of business.

The achievement of a sustainable interoperability, the goal of any interoperable system can then be defined as the ability of an interoperable ecosystem of adapting and coping to the coming changes in such a way that throughout the evolution of time, the benefits of the Uptime are and will be enough to compensate the losses due to Downtime and still comply with the expectations of evolution of business goals and performance.

Mimicking the industry processes of sustainability, a sustainable interoperability can only be pursued if the enterprises that are comprised in the interoperable ecosystem perform a consistent, planned and very well-orchestrated operation of finding ways to reduce Downtime, and especially reducing its dependency from complexity.

This research aims to prove that:

- A consistent perception of the environment knowledge will lead to a better understanding between all the participants, and help the early detection of changes in the environment (or even of small pre-change indicators), favouring the adoption of proactive early solutions and the timely planning of contingencies, hence mitigating/reducing the probability and impact of interoperability changes, and thus, reducing or avoiding Downtime;
- The establishment of a flexible and modular set of reusable tools, added to the body of knowledge, shall permit the development of standard and predictable solutions and behaviours;
- A decision support system based on negotiations will promote the development of mature and standard solutions, which will be able to endure. The analysis of previous negotiations and corresponding decisions (of the same business or market analysis) will enable the determination of faster and better supported new decisions;

- An adaptable solution platform shall permit the rapid reconfiguration of the business in order to promptly respond to new requirements.

Achieving a sustainable Enterprise Interoperability, more than defining the MDI and B-MDI, is about defining and communicating for each MDA level what are the strategy drivers that lead to the development of technologies and the results from the technologies that drive business decisions and new strategies.

This is especially important when dealing with the exposed business case of the ESA-CDF space missions (detailed on section 4.4), as the business at stake is intimately related to people interaction and integration. The design of space missions in this facility is what can be called a major, moderated and controlled set of brainstorming sessions where human interaction is of the utmost importance. Moreover, the needs and problems reported by ESA are not solely accounted to infrastructure and ICT interoperability problems, it has specific concerns about human interoperability as well, as the study domains need to interact with each other personally, and there are several reported problems of loss of interoperability in this matter, which leads to rework and lack of reuse of methodologies and tools defined in previous studies.

The current challenge to achieving a sustainable EI (SEI) is therefore to build, on top of the currently established services, other services concerning technology, business and people, which provide the ability to rapidly adapt to innovation and imminent changes in the surrounding environment and yet maintain the seamless interoperability towards its ad-hoc network of partners, providers, subcontractors and customers. This pushes EI a step ahead towards its complete life cycle.

Achieving a sustainable EI is then [78] an integrated and interactive process of adaptation in a constant and iterative effort to recheck the existing interoperable status, while maintaining the existing interoperability towards the surrounding environment. Knowledge, adaptation and flexibility are therefore the pillars for undertaking a Sustainable EI, as described on Figure 2.8.

Figure 2.8: The pillars for a Sustainable Enterprise Interoperability (SEI)

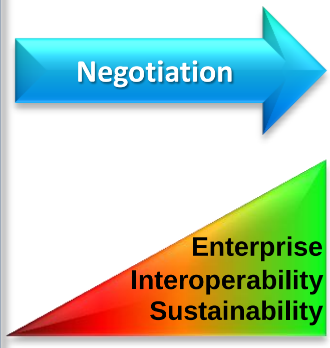
This research extends the state of the art in the EI area, by proposing a collaborative framework to identify critical aspects of the enterprises' interoperability and their impact towards the modelling and servitisation of technology, business and human behaviour (section 3.7).

This framework includes mechanisms to identify the aspects related to the interoperability problems that are sensible to changes in the networked environment, including services to support the negotiating of solutions that act on these changes, enabling the sustainability of the enterprise-networked interoperable environment along its life cycle with less transient downtime. It also includes a methodology for implementing the items depicted in Figure 2.8, which includes the establishment of an adaptive platform based on services and proposes negotiations to be present in all the steps concerning the definition and development of interoperability assets (section 4.2).

Table 2.1 shows the major issues observed in the current enterprise implementations and the proposed changes resulting from this research.

Table 2.1: Current business vs Proposed business

Current	Proposed
Empiric knowledge of possible solutions	Formal and justified solutions
Previous mistakes are often repeated	Lessons-learned avoid previous mistakes
Quick but poor solutions, leading to frequent rework	Mature solutions tend to form standards
Frequent patches to cover unforeseen scenarios	Negotiations are time consuming, but lead to complete solutions



2.5. Negotiation

As Negotiation is one of the core subjects of this investigation, this section shall only contain the literature review related to negotiation, as the research contributions shall be specified on the separate section 3.

Oliveira and Camarinha-Matos [80] defend the creation of Virtual Enterprises (VE) or Network of Enterprises from the need of enterprises to collaborate towards increasing their economic efficiency, leading to the creation of Virtual Organizations (VO). They require an electronic negotiation support environment that increases the agility to create successful dynamic VOs in the heterogeneous context of the VO breeding environment (VBE). In order to achieve their goal, they adopt several hypotheses to support the negotiation process of dynamic VOs with "smart" characteristics such as: collaboration risk reduction, management of participants' expectations, traceability, etc.

The creation of a distributed multi-party multi-issue negotiation platform is proposed in the Open Negotiation Environment (ONE) project [81]. Such platform leverages the Digital Ecosystem (DE) paradigm, where an essential element is represented by the negotiation of alliances among entities, with complementary or even competitive competences. Within the same subject, de la Rosa et al. [82] describe a recommender system in project ONE where agents negotiate services on behalf of small companies.

The solution of using a common ontology for negotiations has been approached in many works, such as Torres and Wijnands [54]. Albeit beneficial in many ways, the use of a common ontology becomes much more complex when dealing with multiple application fields regarding creation, update and efficient structuration. Gruber also outlined the use of formal ontologies to support the negotiation and exchange knowledge perpetrated by intelligent agents [83].

The final goal of the negotiation process consists in reaching a common agreement among parties in order to support possible collaborations. The SANA (Supporting Artefacts for Negotiation with Argumentation) framework [84] incorporates intelligent components able to mediate the agents participating in negotiation to reach an agreement by inferring mutually-acceptable proposals. This solution of using an artificial intelligent mediator can be found in other researches on argumentation based-negotiation, particularly in systems designed for public deliberation [85].

In the distributed environments of multi-agent systems, interaction protocols represent a promising approach for coordination in multi-agent systems. Miller and McBurney [86] introduce the term “first-class protocol” referring to a protocol which allows agents to select, reference, share, compose, invoke and inspect protocols at runtime. Parandoosh performed an analysis [87] on the some of the most common methodologies used to develop agent-based systems:

- **Multi-agent Systems Engineering (MaSE)**, which consists in two major phases, similar to object-oriented engineering: Analysis (definition of objectives, use-cases, tests and roles), and Design (Creation and Assembly of Agent classes, Conversations and System design);
- **Gaia**, which provides a modelling framework for the development of agent systems, also defining two steps for development: Analysis and Design;
- **Prometheus**, which is an agent-oriented software engineering (AOSE) supported by the JACK development environment (see section 2.5.2.2) which provides a modelling graphic design tool which converts graphic specifications into agent code;
- **Tropos**, which is a methodology somewhat linked to Agile development methodologies in the sense that it comprises requirements analysis in all the phases of the software development, having a strong focus on early requirements, goals and objectives.

In the same field, the Interactive Recommendation and Automated Negotiation Systems (IRANS) [88] uses multiple criteria decision-making and automated negotiation. The bilateral and multilateral negotiation in IRANS is made through mediator agents that perform the negotiation process, managing communication among IRANS entities and filtering the offers from buyers/seller agents on behalf of owners. In a subsequent work [89] the e-negotiation system proposed solves the problem of multiple-issue negotiations. In addition, the system is based on the multi- agent system approach in which agents can make autonomous negotiation decisions.

The Multi-Agent Supply Chain Simulation (MACSIMA) framework [90] allows the design of large-scale supply network topologies. These are formed by a multitude of autonomous agents that act on behalf of the enterprises they represent in the supply network. The result of the proposed solution shows that the use of negotiating agents for coordinating a supply network is conditional on the fine tuning of the parameters of the learning mechanism of the agents in order to reach an optimal coordination outcome. The setting of the parameters of the learning mechanism leads to a significant variation of the overall profit and turnover of a supply network. MACSIMA provides all agents with negotiation and learning capabilities that allow the evaluation and simulation of the joint evolution of the price negotiation strategies of the business agents that exchange goods over an electronic business-to-business (B2B) market.

A study was presented by Duan et al. [91] about applying automated negotiation to self-interested agents that have complex utility functions based on local combinatorial optimization problems. Their objective is to apply automated negotiation to achieve better overall efficiency of the supply chain. Their agreement approach is based on the fact that each agent has a combinatorial optimization problem and the set of common values grows exponentially, making it impossible to propose every solution in a particular order based on learning. In addition, they propose a communication protocol and two negotiation strategies for making concessions in a joint search space of agreements. In a subsequent work [92] they introduce a Pareto improvement phase at the end of a negotiation which brings the final agreement closer to the Pareto frontier.

Luo et al. [93] developed a template-based knowledge engineering methodology for building automated negotiation system, which consists essentially on a generic knowledge model of a main task and a library of modular and reusable templates of subtasks. The different combination of templates may constitute different automated negotiation models. Jazayeriy et al. [94] also provide a review on the progress of soft-computing (SC) techniques used in e-negotiation. Their approach is based on the idea that using a combination of soft computing techniques, e.g., Fuzzy Logic (FL), Neural networks (NN), Genetic algorithms (GA), and Probabilistic reasoning (PR) can decrease the complexity of negotiation making it closer to real world negotiation.

Relating to the same idea of constructing a more human-like agent model, Cao [95] describes a generic decision-making model based on the Belief-Desire-Intention (BDI) agent architecture, which can support dynamic selection of the negotiation strategies during the negotiation process, filling the gap between the theory development and the implementation of the negotiation support model. A multi-layered, multi-agent architecture is also described by Cretan to support negotiations in inter-organisational alliances (e-Alliances) [96].

One problem that can be incurring by the constant use of negotiations throughout interoperability is the existence of deadlocks. Several approaches have been proposed in literature addressing this problem. Guo Qing and Chen Chun [97] stated that negotiation deadlock on single issue may occur in multi-issue negotiation when agents cannot make a compromise on one of issues.

Their strategy for avoiding the deadlock is based on an optimization algorithm for reservation value of multiple issues in bilateral negotiations by using a Reinforcement Learning algorithm which broadens the reserved value of the negotiation and narrows other irrelevant issues.

Other approaches tackle the issue of negotiation deadlock by introducing a mediator in the negotiation process such as the proposal of Lin and Chou [98]. They introduce a mediation mechanism to break the deadlock by supervision from the third party. On recent works, Yang et al. [99] solve this problem occurred in negotiation without time limit by using an optimization strategy based on a learning method to estimate opponent's weight of their negotiation issue in order to give them full consideration to the issue dependencies.

Still related to same area of collaboration processes, conflict prevention and deadlock-free, Fang Huan and Wang Lili [100] provide a model of task dynamic allocation among multiple agents based on Contractor Network Protocol (CNP) and Coloured Petri Nets while Quan Liu et al. [101] present a prototype system of conflict prevention and resolution based on a video conference negotiation strategy.

Recent papers describe the advantages of using negotiation in the cloud computing. Li Pan [102] states that the main issue for cloud computing in order to support negotiation is to establish the Service Level Agreement (SLA) between service providers and consumers and proposes a software agent based automated service negotiation framework for on-demand cloud service provision.

MinChao Wang et al. [103] provide a SLA platform in order to help cloud consumers to select most reliable cloud providers and make negotiation process more equitable and convenient. Groléat and Pouyllau [104] tackled the end-to-end SLA negotiation problem by optimizing long-term revenues of domains while processing dynamically customers' requests.

Finally, in the Internet of Things Strategic Research Roadmap, Vermesan et al. elicit the importance of event-driven agents in the establishment of an intelligent and self-aware behaviour of the networked elements [24]. Ferreira et al. describe an implementation of the MIRAI Multi-Agent System (MAS) in JADE, and enumerate the agents' main characteristics: autonomous, social, reactive, and pro-active [17].

2.5.1.Agent-Based Negotiation

Negotiation is the communication process used to obtain agreement over some matter of common interest. Negotiations may involve two parties or multiple participants. The negotiation subject may concern a single issue, in which case there is a conflict of interests between the two parties, i.e., the benefit of a participant results in the detriment of the others, therefore means a win-lose relationship.

On the other hand, negotiations may concern multiple issues, and in this case it is possible for all the involved participants to benefit, hence resulting in a win-win settlement. However, the problem regarding multiple issues concerns the complexity of the negotiation, resulting in an increase of the time and effort to reach an agreement.

The term "agent", philosophically meaning an entity capable of performing actions, emerged from the evolution and conjunction of artificial intelligence and object-oriented concepts, is defined as a computer system, situated in some environment, capable of flexible and autonomous actions towards a defined objective [105].

It is a term commonly used in many non-computer areas, particularly in economics, but also in other, e.g. sociology, chemistry, philosophy.

Three main characteristics for an agent are enumerated:

- **Situated**, meaning that the agent is embedded in an environment from which it receives inputs and whose outputs can change that environment;
- **Autonomous**, which considers that though an agent is defined and designed with a specific purpose, it is able to act without the direct intervention of humans, and controls its own actions and states;
- **Flexible**, in terms of its responsiveness (ability to detect changes in its environment and respond to them timely), in terms of pro-activeness (targeted, initiative, and focused), and in terms of sociability (ability to interact with humans and with other agents).

Although an agent-based system may be composed of a single agent, representing an entity that performs some processing, Multi-agent Systems (MaS) empower the concepts of distributed computing and are able to describe and handle complex systems and problems. In many ways, a Multi-agent System can be seen as network of multiple small-sized elements that collaborate towards a determined purpose, where the value of the collective is usually greater than the sum of the individual contributions.

The use of intelligent agents to perform the negotiation tasks on behalf of their owners is one of the most effective tactics towards the automation of negotiation processes between business-to-business parties [94]. The use of automated Negotiating Software Agents (NSAs) has proven to be able to reduce significantly the negotiation time and thus provide an efficient way to deal with the limitations of the human relation aspects.

The main characteristics of a MaS are:

- The information in each agent may not be complete, its capability for solving the problem may be limited, or they have a limited view of the business, thus favouring the concept of distributed processing, collaboration and group conscience;
- The absence of a global system control, nor centralisation of data, which enforces the idea that each agent is autonomous and a small but essential piece of the business;
- Asynchronous computing, which is the most effective and transparent way to mimic, support and synthesize the reality.

2.5.2. Agent-Based Software Development

An analysis over a set of well-known agent-based development platforms was performed to determine the development platform that would be best-fitted to be used for the proof-of-concept described on section 4.5. This analysis started by enumerating the criteria that would be used for the selection of a target platform, and then considered five referenced platforms: JACK, OAA, JADE, Jason, and AnyLogic. The analysis concluded with the justified selection of a development platform.

2.5.2.1. Selection Criteria

The criteria for selecting a platform for development, considering the research constraints, were:

- The costs to obtain the full version of platform, i.e., free to use versions have a higher weight in the selection;
- Documentation available to have enough support on installation and developing;
- Preference on the use of Java language in order to avoid the learning curve of a new programming language;
- The need for negotiation attribute, in order to provide a simple but effective negotiation between various agents;
- Easiness on developing and understanding the programming code.

2.5.2.2. JACK Agent-based development platform

JACK (analysed version: 5.6, 2012) is an Agent Oriented development environment from AOS (Autonomous Decision-Maker Software, [106]) that consists in a collection of autonomous agents allowing them to communicate.

JACK's main features are listed below:

- Lightweight and efficient cross-platform;
- Low resource requirement, which allows the environment to manage hundreds of agents at the same time;
- Have its own programming language, the JACK Agent Language;
- Transparent inter-agent communication;
- Graphical agent development tools;
- Very well documented.

A great advantage of this technology is that it is light-weighted and doesn't need much computer resources in order to run hundreds of agents at the same time.

On the other hand, JACK has a determined specific language, the JACK Agent Language, hence it is necessary for the developer to learn a new programming language, despite its similarity with of Java.

JACK offers an Evaluation license for 60 days. This is a downside as there are other studied platforms that are open-source projects with no costs associated.

Figure 2.9 illustrates the JACK IDE for programming agents.

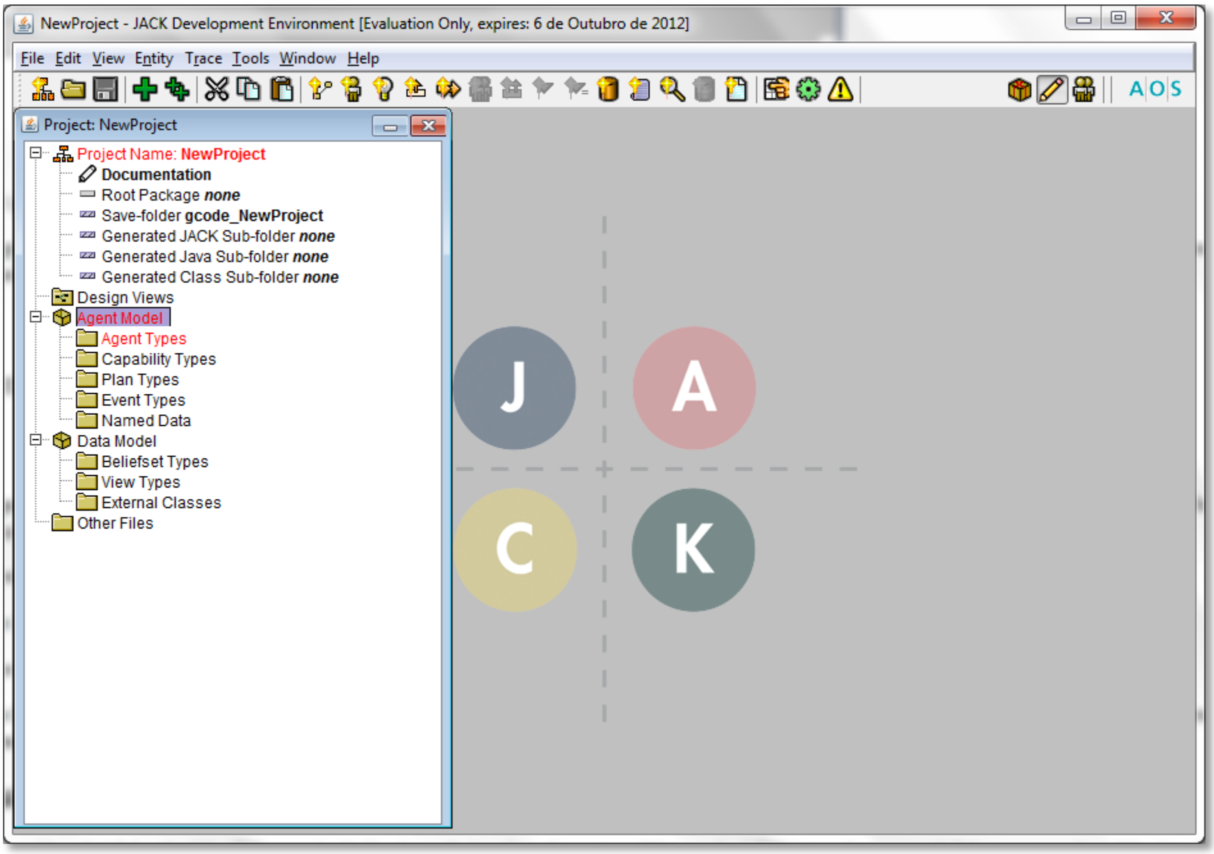


Figure 2.9: The JACK agent development platform

2.5.2.3. JADE Agent-based development platform

The Java Agent Development framework (JADE, analysed version: 4.2, 2012) is a software framework that simplifies the implementation of multi-agent system and is distributed by Telecom Italia [107].

JADE principal features are listed below:

- It allows a simple implementation of a multi-agent system (MaS);
- It uses the FIPA (Foundation for Intelligent Physical Agents) specification for communication inter-agents;

- The agent platform can be distributed across machines with different Operating Systems;
- It provides a Graphical User Interface to control the agents. This control can be performed remotely;
- It is open-source software with no costs associated;
- It is very well documented.

Creating an agent environment in Java with JADE is an easy task since we can get a lot of documentation and a lot of tutorials in the JADE website. On the other hand, installing JADE on computer is a somewhat complicated task, despite the availability of a lot of documentation, some of which seems to be a bit outdated.

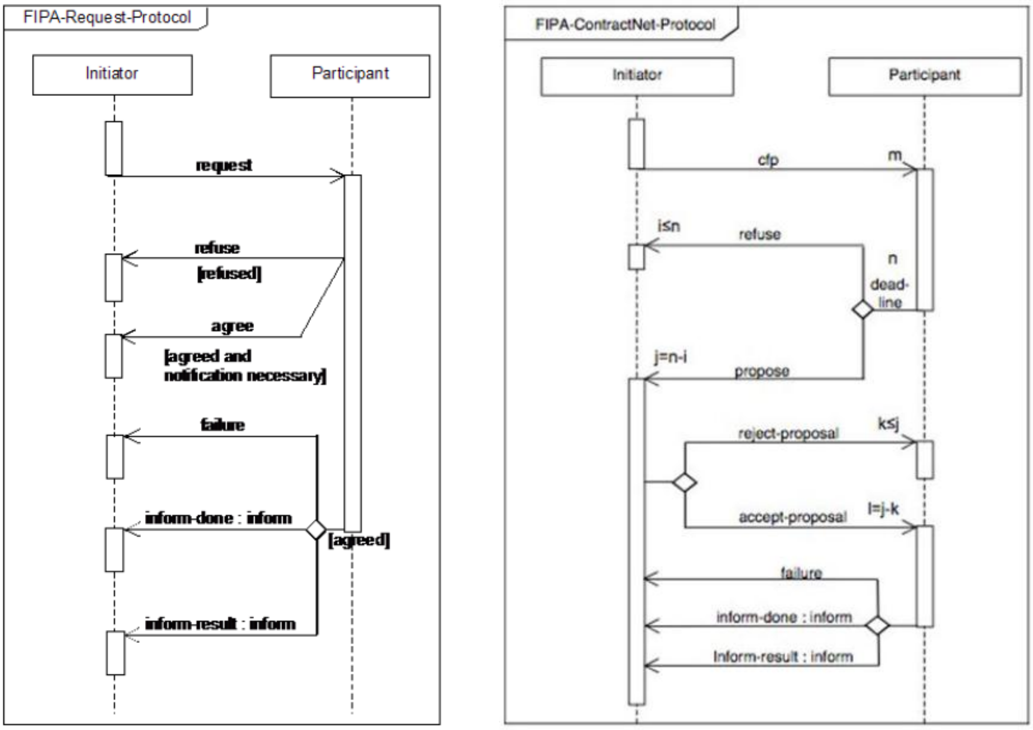


Figure 2.10: The FIPA protocol for JADE agent communication

JADE agents need to obey to the FIPA protocol, as can be seen on Figure 2.10, in terms of inter-agent communication; this means that every communication made by an agent to another need to respect the protocol. This obligation can be a problem, as it can complicate the development of a simple application with simple messages to be changed between agents.

On other hand, JADE is open source with no costs associated and uses Java for programming the agents, which is an advantage, because the developer doesn't need to learn one more language.

From the analysis that was performed, the FIPA protocols are used to establish the communications between agents and this is performed using some sort of negotiation itself.

The first one, the FIPA Request, is normally used when an agent needs to make a request to another agent. The second, called FIPA Contract Net, is used when an agent needs to make a proposal to all agents in the network.

2.5.2.4. Jason Agent-based development platform

Jason (analysed version: 1.3.7, 2012) provides a platform for the development of multi-agent systems, using an extended version of AgentSpeak, which is a programming language based on BDI (Beliefs-Desires-Intentions) architecture.

The main features of Jason are:

- Support for developing environments, i.e., can be programmed in AgentSpeak language or in Java language for example;
- The possibility to run a distributed multi-agent system over a network using JADE;
- It has its own IDE (jEdit) and provides an Eclipse plugin;
- It is well documented, and the Jason website provides very good tutorials;
- It is an open source project free to use.

Figure 2.11 illustrates a simple code in Jason for a "hello world" agent.

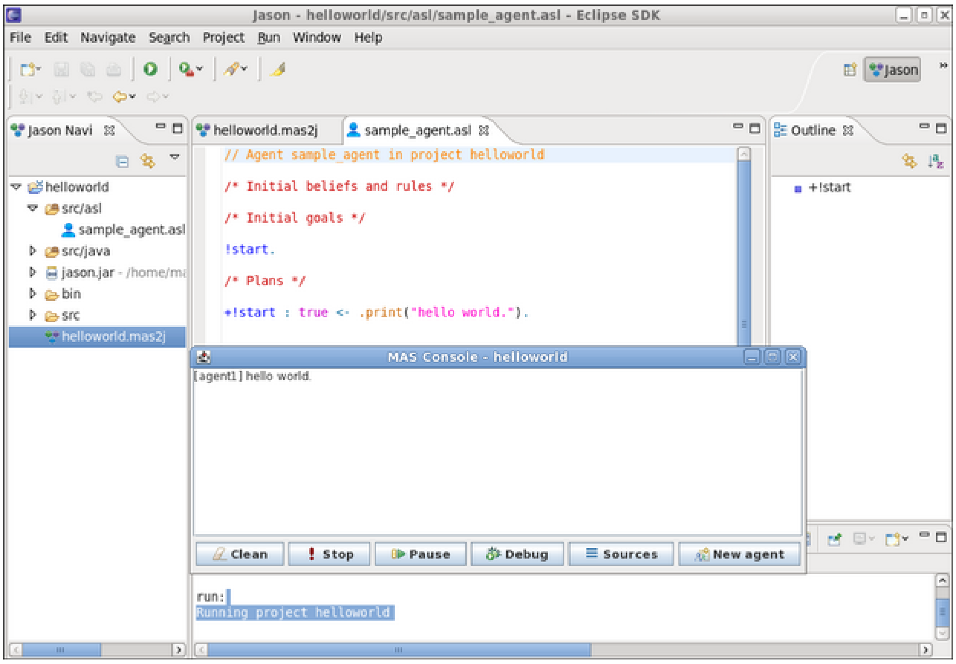


Figure 2.11: The Jason agent development platform

With the eclipse IDE (it needs version 3.7 or higher) plugin, programming Jason agents can be easy, but the fact that it uses a specific language to do so (AgentSpeak) can be a negative point, since the developer needs to learn a new programming language.

The installation is very simple and the tutorials provided in the web site are very well explained; contrariwise, the plugin for eclipse IDE has a lot of instability, which is not good for developers.

Jason is an open source project, which is an asset, comparing with other agent technologies.

2.5.2.5. AnyLogic Agent-based development platform

AnyLogic (analysed version: 6, 2012) provides an agent-based modelling language based on a simulation methodology. The main feature of AnyLogic is that it provides a visual language which simplifies development of agent based models. Other features of this technology are listed below:

- It uses UML state charts to define agent behaviours and UML actions charts to define algorithms;
- It's a simulation-oriented software;
- It supports limitless extensibility, including custom Java code;
- It's not a free application, although there are a lot of editions, including evaluation and educational editions;

Figure 2.12 shows a screenshot of the AnyLogic IDE with an example of a state chart, representing the agent behaviour.

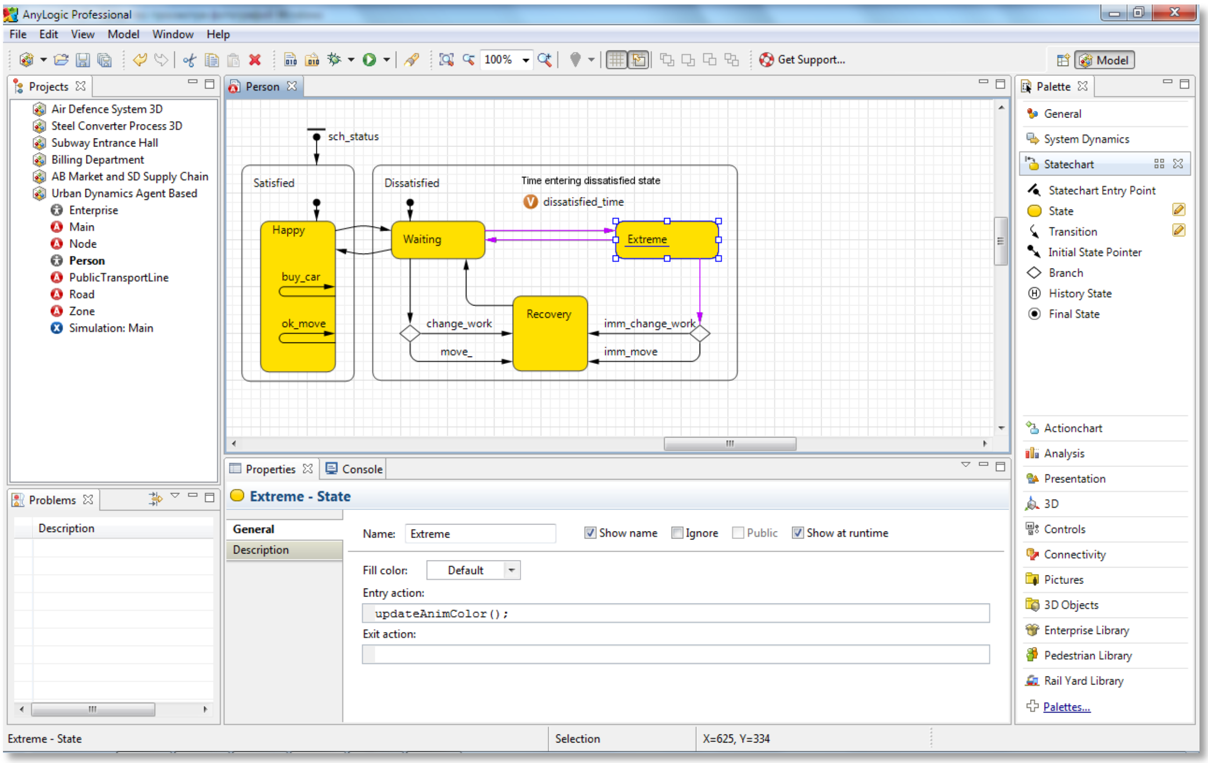


Figure 2.12: The AnyLogic agent development platform

The main focus of this application is the development of models using agent-based, system dynamics, discrete-event, continuous and dynamic system models. To do this, AnyLogic provides its own IDE with lots of tools.

AnyLogic has a lot of documentation which makes this application very strong in the market.

2.5.2.6. OAA Agent-based development platform

The Open Agent Architecture (OAA) (analysed version: 2.3.2, 2007) is a framework for integrating agents in a distributed environment and its main features are shown in the list below:

- It uses the ICL (Inter-agent Communication Language) for the communications;
- It supports many programming languages and systems;
- It is an open source project with no costs associated.

This technology is not as well documented as JACK or JADE, but nevertheless there are some documents on the subject. It uses a different type of communication between agents than JADE but in general the communication method is similar to JADE.

The last version of the software was released 5 years ago, which may indicate that the developers abandoned the project or it can be seen in a positive way, that the project might be in a mature state.

2.5.2.7. Selected Candidate

To perform the selection of the best tool for Agent-Based development considering the determined criteria, the author started by excluding the technologies that don’t offer a free product, which was only coped by the following technologies:

- JADE;
- OAA;
- Jason;

From this list, OAA was excluded due to the absence of documentation. So, the decision came between the following technologies: JADE and Jason.

Since Jason uses its own programming language and as the AgentSpeak language is somewhat difficult, the final decision was favourable to JADE.

Therefore, the selected candidate for development was the JADE agent platform because from all the technologies presented in this document, JADE is the one that, in some part, can perform all selection criteria points.

2.5.3. Rules Engine Development

An analysis over a set of well-known rule engine development platforms was performed to determine the development platform that would be best-fitted to be used for the proof-of-concept described on section 4.5. This analysis started by enumerating the criteria that would be used for the selection of a target platform, and then considered five referenced platforms: JESS, Drools, OpenRules, Zilonis and Hammura. The analysis concluded with the justified selection of a development platform.

2.5.3.1. Selection Criteria

The criteria for selecting a platform for development, considering the research constraints, were:

- The costs to obtain the full version of the engine, i.e., free to use versions have a higher weight in the selection;
- Documentation available to have enough support on installation and developing;
- Easy to understand rules language in order to does not spend much time learning a new language;
- Easiness to integrate with Java code.

2.5.3.2. JESS Rules Engine development platform

The Java Expert System Shell (JESS, analysed version: 7.1, 2008) most important features are:

- It's small, light and fast;
- It uses the enhanced version of Rete algorithm to process rules;
- It manipulate and reasons about Java objects;
- It supports a native XML rule language called "JessML";
- JESS is also a Java scripting environment (Java Interface);
- It has a fully-featured development environment based on Eclipse.

JESS applications can be written from pure JESS language scripts (Figure 2.13) to pure Java code (using the Java API provided by JESS). The JESS scripts use a language which is similar to LISP, which is not so easy to understand, and this is a constraint. However, there's some documentation available on the JESS website, which allows supporting programming with JESS.

JESS needs to be used with Eclipse IDE and that it's a possible constraint. On the other hand, once JESS uses Java, the best IDE to programming in Java is probably Eclipse.

```
(defrule done
  ?problem <- (problem (answer-fractions $? ?unreduced-answer $?))
       ?unreduced-answer <- (fraction (numerator ?num) (denominator ?denom))
  ?num <- (textField (value ?n&:(neq ?n nil)))
  ?denom <- (textField (value ?d&:(neq ?d nil)))
  (test (eq (gcd ?d ?n) 1))
=>
(predict done ButtonPressed -1)
       (construct-message
              [ Is there anything else to do?]
              [ If the greatest common divisor of a numerator and a denominator is 1,
           then the fraction cannot be further reduced.]
              "[The greatest common divisor of " ?d " and " ?n " is 1.]"
              [ You are done. Press the done button.]
        )
)
```

Figure 2.13: The JESS rules language

JESS is simple and easy to install, and easy to integrate with the Eclipse IDE. A great feature is that JESS is available at no cost for academic use. However, the latest version of JESS was released in 2008, and there are no evidences of activity from its developers.

In terms of performance, according to the JESS web site, JESS is small, light and one of the fastest rule engines available.

2.5.3.3. Drools Rules Engine development platform

Drools (analysed version: 5.4.0, 2012) is a bit more complex than JESS in terms of architecture, as it consists in five modules: The Drools Expert (rule engine), the Drools Guvnor (Business Rules Manager), the jBPM 5 (Process / workflow), the Drools Fusion (Event processing/temporal reasoning) and the Drools Planner (Automated planning).

Drools' main features are:

- It uses an enhanced implementation of the Rete algorithm;
- It supports Forward and Backward chaining;
- It has temporal reasoning (temporal relationships between events);
- It provides an Eclipse-based IDE;
- It requires Java 1.5 (Java SE).

```
package com.sample

import com.sample.DroolsTest.Person;

rule "Say hello to a Guy"
    when
        p : Person(gender == Person.MALE, myAge : age, myName : name)
    then
        p.setTitle("Mr”.);
        System.out.println("From Drools: Hello Mr. " + myName + " you have " + myAge + " years old!");
end

rule "Say hello to a Girl"
    when
        p : Person(gender == Person.FEMALE, myAge : age, myName : name)
    then
        p.setTitle("Ms”.);
        System.out.println("From Drools: Hello Ms. " + myName + " you have " + myAge + " years old!");
end
```

Figure 2.14: The Drools rules language

Drools is very user friendly, it even uses some interfaces integrated with the Eclipse IDE.

It is very well documented, very easy to understand and start programming.

Drools is an open source project, so no costs are involved.

Drools is more complex than other Rule Engines technologies, due to having various modules to install and to understand; however, if the rule engine module is sufficient for the required needs, then there is no complexity.

2.5.3.4. OpenRules Rules Engine development platform

The main features of OpenRules (analysed version: 6.2.1, 2012) are (Figure 2.15):

- It uses Excel as a rule editor;
- It provides many predefined decision tables;
- It provides a plugin to Eclipse IDE;
- It has a high performance and efficient use of memory.

Figure 2.15: The OpenRules suite of tools

OpenRules, combined with Excel and Eclipse (Java), offers a simple way to store and execute rules, since the rules are stored in Excel files, so it's very easy to create/update them.

Normally, a OpenRules project needs four Excel documents to run: One file for the decision tables, another file to represent the rules, another file for the glossary and another for the test data. This makes a little complex to execute a simple application, because we need always more than one excel file.

The use of Excel files can be a constraint once the rules can only be declared in the "xls" files. In other hand, if that is no problem to use/create this type of files, this feature can be very useful.

The use of Excel files to declare the rules is an advantage when comparing with other rule engines where it is necessary to know some programming.

Because this engine is well documented ally with the rules defined in Excel, becomes very easy to understand the rules.

Some cases where is needed only one Excel file, inside of this file, exists always more than one worksheet. Figure 2.16 illustrates an example that contains two rules tables in Excel.

Rules void **defineGreeting**(int hour, Response response)		
Hour From	**Hour To**	**Set Greeting**
0	11	Good Morning
12	17	Good Afternoon
18	22	Good Evening
23	24	Good Night

Rules void **defineSalutation**(Customer customer, Response response)		
Gender	**Marital Status**	**Set Salutation**
Male		Mr.
Female	Married	Mrs.
Female	Single	Ms.

Figure 2.16: OpenRules rules in Excel

The first table defines the rules for greeting people, depending on the hours of the day and the second table defines the rules form saluting people, depending on the gender and the marital status.

Figure 2.17 represents an example of an OpenRules decision table in Excel.

Decision DetermineCustomerGreeting	
Decisions	**Execute Decision Tables**
	:= System.out.println(customers[0])
Define Current Time	:= DefineCurrentHour()
Define Greeting Word	:= DefineGreeting()
Define Salutation Word	:= DefineSalutation()
Define Customer Greetin	:= decision().put("result", response.greeting + ", " + response.salutation + " " + customers[0].name + "!")

Figure 2.17: OpenRules decision table in Excel

There is a free Evaluation version of OpenRules, however it limits the total number of OpenRules tables used in the application which is a limitation considering that the excel tables represent the rules.

2.5.3.5. Zilonis Rules Engine development platform

The features of Zilonis (analysed version: 0.97b) are:

- It is based on a variation of the forward chaining Rete algorithm;
- It supports multithreading;
- It provides an interface that allows to see the internals of how the rules are getting compiled;
- It requires Java RE 1.6.

The multithreading capabilities are a nice feature that can be an asset to the application development.

Zilonis is an open source project and is free to use, so no costs are associated with this option, but there is not much information/documentation about this rule engine.

2.5.3.6. Hammurapi Rules Engine development platform

The main features of Hammurapi Rules (analysed version: 3.0.0.15, 2009) are:

- It supports Forward and Backward chaining;
- The rules are written in Java;
- The rule sets are assembled and configured in XML;
- Hammurapi supports multithreading.

This engine does not require additional knowledge, since the rules are written in pure Java language.

Executing rules in parallel threads can be an asset to improve performance, although that can bring some complexity to the application. This feature can be analysed and should be applied only if is necessary.

Hammurapi is an open source project and free to use.

Hammarupi Rules has been replaced by Hammarupi Event Bus and becomes more complex with less documentation.

The last version of this engine was released on 2009 and seems that there’s no activity on the project anymore.

2.5.3.7. Selected Candidate

To perform the selection of the best tool for Rules Engine development considering the determined criteria, the author started by excluding the technologies that don’t offer a free product. In this case, the only tool that was excluded was OpenRules

The next technologies to be excluded are Zilonis and Hammurapi, because there is not much information/documentation about these engines.

The remaining technologies are JESS and Drools.

Both JESS and Drools are very easy to integrate with Java, thus the selection had to deal with the complexity of the programming language of each one of them. Since JESS uses a programming language which was considered a little bit more complicated, comparing with the one used by Drools, the selected candidate is the Drools rule engine technology.

3. NEGOTIATION

"

You got to know when to hold 'em, know when to fold 'em,
Know when to walk away, and know when to run.
You never count your money when you're sittin' at the table.
There'll be time enough for countin' when the dealin's done.

"

The Gambler – Don Schlitz

Businesses evolve defining their own concepts, interfaces and methods. Achieving and maintaining interoperability between players that decide to collaborate then becomes an act of understanding each other player's business and adapt to eventual changes driven by it.

As enterprises raise their levels of business complexity and maturity, and as the whole surrounding environment (ICT, services, providers, partners) is evolving to concern less about syntax and connectivity and more about semantics, contents, and knowledge, the number of options and decisions towards interoperability that need to be taken is escalating in number and importance, and these decisions have and increasing influence on the evolution of the enterprise business.

Kaddouci et al. [108] state that in any case of divergence, negotiation is the most appropriate method to solve conflicts. Usually, interoperability conflicts are resolved via one party forcing the other to change, or by reaching consensus towards a midway solution. This main driver of this research is the need for interoperating enterprises to formally negotiate the interoperability solutions towards reaching the most appropriate decision, one that minimises the effort and time spent regaining interoperability.

Managing and moderating adaptation through negotiations promotes reaching the best decisions about interoperability changes, allowing interoperability parameters (e.g., time, effort, cost) to be optimised. It is the proper way to resolve discrepancies, allowing a correct formalisation of the SWOT (strengths, weaknesses, opportunities and threats), while presenting the various alternative solutions. It also enables a clear definition of criteria (e.g., impact, downtime, cost, alignment with enterprise strategy, new markets) for the selection of a particular solution, providing a strong justification for reaching mature choices.

Negotiations are sets of complex actions, some of which may occur in parallel, where multiple participants exchange and take decisions in multiple phases over a set of multiple attributes [109]. The participants to a negotiation may define proposals, and each participant can decide autonomously to end a negotiation, either by accepting or by rejecting the received proposal. Depending on its role in a negotiation, a participant may invite new participants to the negotiation. The negotiation services make use of negotiation techniques and negotiation model to determine the best alternatives for the negotiation.

On each negotiation, in any of these layers, the problems are exposed and formalised, the benefits are presented and the interoperability solution alternatives are enumerated. The negotiations can then reach several conclusions:

- One of the partners agrees in changing to the other's definition;
- Both partners agree in a compromise solution;
- The partners agree in a solution imposed by the supervisor;
- The partners agree that interoperability can only be achieved via the use of translators and mediators;
- The partners agree that interoperability can only be achieved partially and thus need to redefine the scope;
- The partners agree that interoperability is not possible, not desired or not worth the changes/effort/time needed to achieve it.

3.1. Negotiation Model

Each negotiation is organized in three main steps:

- **Initialization**: This step allows the definition of the Negotiation Object (i.e., what is to be negotiated) and how (Negotiation Framework) [92];
- **Refinement of the job under negotiation**: In this step, the participants exchange proposals on the Negotiation Object trying to satisfy their constraints [110], [111];
- **Closure**: Here, the Negotiation Object is reflected into decisions to be implemented by the framework services towards interoperability [112], and the negotiation is concluded.

The different negotiation scenarios that can be found in the lifecycles of the business to business negotiations are able to be modelled. Some of these scenarios can range from simple cases of selection of possible partners and a direct outsourcing of a job, to more complex scenarios of concurrent negotiations with multiple partners to outsource a non-divided job or concurrent negotiations with the possibility to dynamically split the job during the negotiation.

This section proposes a formal model to settle and manage the coordination rules of one or more negotiations which can take place in parallel, by describing the basic concepts underlying the model, and the negotiation model using the metaphor of Interaction Abstract Machines (IAMs, section 3.1.2). The Program Formula is described to define the methods used to manage the parallel evolution of multiple negotiations.

3.1.1.Basic Concepts

At a local level, the model requires a formal description of the coordination rules that manage the behaviour of all agents in a negotiation. At a global level, the model must provide a global coordination of all negotiations of an agent.

A Negotiation Model is defined by a quintuplet

$$M = < T, P, N, R, O >$$

where:

- **T** denotes the timestamp of the system, assumed discrete, linear, and uniform [113];
- **P** denotes the set of participants in the negotiation framework. The participants may be involved in one or many negotiations;
- **N** denotes the set of negotiations that take place within the negotiation framework;
- **R** denotes the set of coordination rules among negotiations that take place within the negotiation framework;
- **O** denotes the common ontology for the set of definitions of the attributes that are used in a negotiation.

A negotiation is consequently described at a determined time instance through a set of negotiation sequences.

Let $Sq = \{si \mid i \in \mathbb{N}\}$ denote a set of negotiation sequences, such that $\forall si, sj \in Sq, \ i \neq j \Rightarrow si \neq sj$. A negotiation sequence $si \in Sq$ such that $si \in N(t)$ is a succession of negotiation graphs that describe the negotiation N from the moment of its initiation and up to the time instance t.

The negotiation graph created at a given time instance is an oriented graph in which the nodes describe the negotiation phases that are present at that time instance (i.e., the negotiation proposals sent up to that moment in terms of status and of attributes negotiated) and the edges express the precedence relationship between the negotiation phases.

This model covers formal and non-formal aspects of the interoperability, as they may be qualified and hence modelled in the framework in order to be able to be included in the negotiation.

The *negotiation phase* (*ph*) indicates a particular stage of the negotiation under consideration.

The *Status* is the possible state of a negotiation, such that:

Status* $\in$ *{initiated, undefined, success, failure}

- **Initiated**: the negotiation, described in a sequence, has just been initiated;
- **Undefined**: the negotiation process for the sequence under consideration is ongoing;
- **Success**: in the negotiation process, modelled through the considered sequence, an agreement has been reached;

- **Failure**: the negotiation process, modelled through the sequence under consideration, resulted in a denial by the other participants.

Issues corresponds to the set of attributes with associated values that describe the proposals made in a negotiation phase.

Snapshot is the set of combinations between a negotiation aspect (*Status*) and the information that is negotiated (*Issues*).

3.1.2.Interaction Abstract Machines (IAMs)

Negotiation processes are complex asynchronous interactions between multiple players and it is difficult to describe and model these interactions. One way to perform this modelling is by using standard observation techniques. Berry and Boudol developed a Chemical Abstraction Machine to model the interactions and reactions between molecules [114]. Several implementations have been developed following this concept like Pinto's Sequential and Concurrent Abstract Machine [115] and the Interaction Abstract Machines (IAMs) by Andreoli et al. [116]. The latter was selected to formalise and model the proposed negotiation processes.

Using a metaphor of Interaction Abstract Machines (IAMs) facilitates the modelling of the evolution of multi-attribute, multi-participant, multi-phase negotiations. In IAMs, a system consists of different entities, each characterised by a state that is represented as a set of resources. Each entity may change its state independently of others, using its own resources and the methods of its computational space.

This approach allows the modelling of the evolution of multiple negotiation phases in parallel. It may evolve according to different laws of the following form, also called "*methods*":

A1 @ … @ An <>– B1 @ … @ Bm

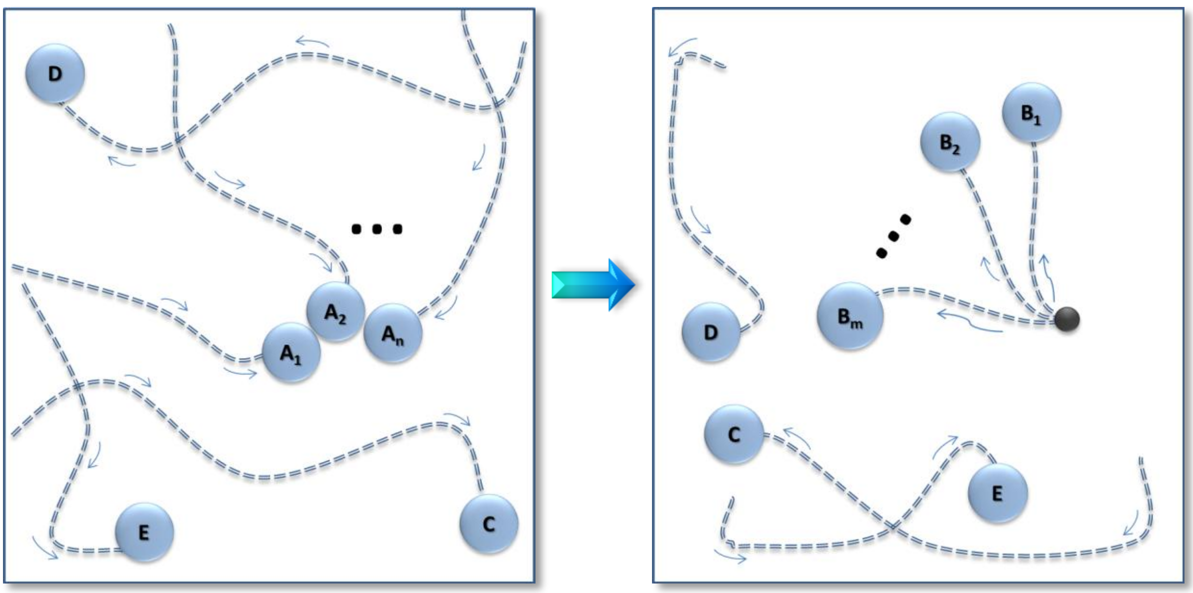


Figure 3.1: The evolution of agents in the metaphor of IAMs

Figure 3.1 shows the evolution of a system of multiple particles. A method is executed if the state of the entity contains all resources from the left side (called the “head”) where the definition of the above rule states that when the particles A_1, A_2, …, A_n are present in the same situation. In this case, the entity may perform a transition to a new state, where the old resources (A1, …, An) are replaced by the resources (B1, …, Bm) on the right side (called the “body”).

All other resources of the entity that do not participate in the execution of the method, i.e., the particles C, D and E existing in the environment maintain their existence unaffected, and hence are present in the new state.

The operators used in a method are:

- The operator “@” assembles together resources that are present in the same state of an entity;
- The operator “**<>–**” indicates the transition to a new state of an entity;
- The operator “**&**” is used in the body of a method to connect several sets of resources;
- The symbol “**T**” is used to indicate an empty body.

In IAMs, an entity has the following characteristics:

- If there are two methods whose heads have two sets of distinct resources, then they may be executed in parallel;
- If two methods share common resources, then a single method may be executed and the selection procedure is made in a non-deterministic manner.

In IAMs, the methods may model four types of transition that may occur to an entity:

- **Transformation**, where the state of an entity is simply transformed in a new state;
- **Cloning**, where an entity is cloned into a finite number of entities which have the same state;
- **Destruction**, in which case the entity disappears;
- **Communication**, which is of type broadcasting, represented by the symbol “**^**”. This mechanism of communication thus executes two synchronous operations:
    - Transformation: if all resources that are not pre-defined at the head of the method enter in collision, then the predefined resources are inserted in the entity and are immediately consumed by the application of the method;
    - Communication: insertion of the copies of the predefined resources in all entities present in the system at that time instance.

3.2. Program Formula

In a multi-entity system, the metaphor IAMs allow the modelling and control of the autonomous evolution process for each entity in the system. Each entity may change its state independently of others, using its own resources and the methods of its computational space. This approach allows the modelling in parallel of the evolution of multiple negotiation phases.

Using the metaphor IAMs, the evolution of the negotiation phases associated to the nodes of a negotiation sequence will be managed through different methods that are put together in a *Program Formula (PF)*. Program Formula of a negotiation sequence *s* – *PF(s)* – represents the set of methods used to manage the evolution of the sequence *s*.

In the proposed negotiation model, a negotiation phase is connected to the set of snapshots of the negotiation status and of the instants of the attributes negotiated that are present in a node of the negotiation graph. In this way, the specification of the information regarding the negotiation state and the attributes values, but also of the actions that will contribute to the evolution of the negotiation, is modelled in the nodes of a graph of the negotiation sequence as sets of particles, called *negotiation atoms*. Thus, a negotiation atom, denoted *atom(s, ph)*, is a set of resources called *particles* that describe the negotiation state in terms of the negotiation sequence *s* for the negotiation stage *ph*. Four types of particles were defined in this way: *representation*, *event*, *message*, and *control particles*.

Representation Particles

Each negotiation sequence keeps sets of snapshots in the nodes of the graphs, i.e., images that each participant has about the negotiation status and about the attributes that are negotiated in the current sequence as well as in all other sequences for which there is a distribution of information.

This information is modelled within the negotiation process as *representation* particles that are described by three parameters: *Name, Status,* and *Issues* of the negotiated attributes with the associated values:

- **Name** is defined by concatenation of the identifiers of the participants with the sequence under consideration (e.g., pjsj);
- **Status** takes values in the set defined in section 3.1.1. This value corresponds to the value returned by the function *status()*;
- **Issues** takes values in the set issues of the negotiated attributes with the associated values. This value corresponds to the value returned by the function *issues()*.

Hence, a *representation* particle of an atom, associated to a sequence *s* for a phase *ph*, is a snapshot of the sequence *s* for the phase *ph*. To provide a detailed description of the negotiation sequences involved in a negotiation phase, the following *representation* particles are defined:

- **localr(Name, Status, Issues)**, local representation particle. This particle holds the local snapshot of the current sequence;
- **extr(Name, Status, Issues)**, external representation particle. This particle holds the external snapshot that describes the modality in which another sequence perceives the same negotiation phase;
- **firstr(Name, Status, Issues)**, external negotiation. This particle holds the external snapshot associated to the sequence that generated the current sequence.

In this way, a new node of a negotiation sequence may be described through a set of representation particles that are part of the same atom.

Event Particles

The *event* particles specify the types of transitions used by IAMs in terms of the message types that are exchanged within a negotiation. An *event* particle is described by three parameters:

- **Id**: identifies the atom to be cloned;
- **New_id**: identifies the newly created atom;
- **Msg:** contains the negotiation message with data that will contribute to the evolution of the negotiation in the newly created atom.

To facilitate the identification of both the cloning operation and the direction in which the new negotiation atom will evolve, four *event* particles are proposed:

- **clone_propose(Id, New_id, Msg)** models an event that signals the existence of a new negotiation message of type *propose*;
- **clone_accept(Id, New_id, Msg)** models an event that signals the existence of a new negotiation message of type *accept*;
- **clone_reject(Id, New_id, Msg)** models an event that signals the existence of a new negotiation message of type *reject*;
- **clone_create(Id, New_id, Msg)** models an event that signals the existence of a new negotiation message that announces creation of a new sequence for the current negotiation.

Message Particles

The *message* particles model the messages sent to allow their processing in terms of their interpretations in a typical negotiation process. These particles have the following parameters:

- **Rname** and **New_r_name** are identifiers of the sequence that generates the message and of a new sequence that is invited to negotiation, respectively;
- **Content** represents the content of the message which is a proposal regarding the negotiation task;
- **Type** represents an identifier of the new coordination policy that satisfies a certain pattern and that must be managed by the sequence invited to negotiation.

Four types of *message* particles are defined:

- **propose(Rname, Content)**, signals the existence of a new proposal with the content *Content* in the negotiation process through the sequence *Rname*;
- ***accept(Rname)***, signals the existence of an acceptance of a proposal that was sent by a participant in the negotiation through the sequence *Rname*;
- **reject(Rname)**, signals the existence of a denial of a proposal that was sent by a participant in the negotiation through the sequence *Rname*;
- **create(New_r_name, Type)** signals the existence of a new sequence that is part of the current negotiation phase, identified by *New_r_name*, where *Type* represents an identifier of the new coordination policy that satisfies a certain pattern and that must be managed by the sequence invited to negotiation.

Control Particles

To properly formulate a coherent execution of a negotiation process, we introduce the *control* particles. These particles have several functions in the computation space of a negotiation sequence:

- An identification function (e.g., *name(Id)*) that identifies the negotiation atoms by specifying an unique value to the parameter *Id* for each atom.
- A limitation function (e.g., *start(), enable(), freeze(), waiting()*) that introduces the concept of control over the methods which may induce errors in negotiation.
- A notification function e.g., *stop(Accord), ready(Accord)*.

3.3. Negotiation Infrastructure

The proposed collaborative framework offers mechanisms to support negotiations in a distributed environment. Conceptually, these mechanisms consist in a set of three hierarchically distributed negotiation levels, represented by components which are implemented as SaaS services that together implement the rules of the negotiation and handle the intrinsic interoperability aspects related to it, as described on Figure 3.2.

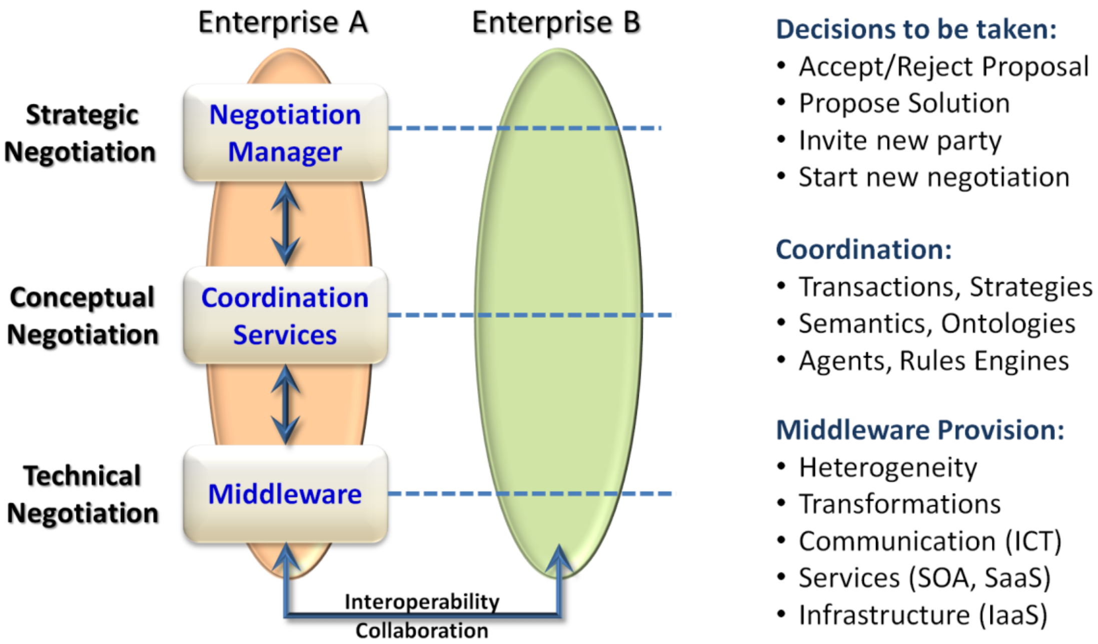


Figure 3.2: NEGOSEIO's negotiation levels

The services on the top negotiation level perform the task of Negotiation Manager close to the enterprises' business decision centres, implementing the strategic high-level business decisions [117] that need to be taken for the negotiation, e.g., starting a new negotiation, inviting other parties to join the negotiation, making proposals to the negotiation, and accepting or rejecting proposals; it also manages the negotiated parameters, and communicates with the lower layers using web-services [118].

A second intermediate negotiation level below it implements the Coordination Services (CS), a module which has the purpose to assist the negotiations at a global level (i.e., negotiations with different participants on different jobs) and at a specific level (i.e., negotiation on the same job with different participants). It handles all issues regarding communication at this negotiation level (synchronisation between the CSs of the several parties that are taking place in the negotiation). It also manages the ongoing transactions and negotiation data persistence, and handles the semantic discrepancies between the negotiating parties. It includes services that perform management of data transactions, semantic interpretation, dynamic discovery of services, and implementation of the business model rules, persisting its data (e.g., business-specific data, semantic ontologies, rules) on IaaS infrastructure and PaaS platforms modelled using standard reference models and data access [119]. It may also include an agent-based architecture [93] to support the complexities of the negotiation operations through the Middleware layer [120].

Finally, a lower negotiation level for Middleware implements communication services issues at technical level, dealing with middleware provision, supporting the aspects related with the basic infrastructures, handling the heterogeneity related with multiple negotiation players, which shall interact using an Enterprise Service Bus (ESB) for dealing with technical interoperability issues. It includes services that handle heterogeneity issues on the basic interoperability level (e.g., authentication, permissions, communications, syntax, session, and data), infrastructure and ICT.

Different negotiation scenarios can be modelled following the life cycles of the business-to-business negotiations. Some of these scenarios can range from simple cases of selection of possible partners and a direct outsourcing of a job, to more complex scenarios of concurrent negotiations with multiple partners to outsource a non-divided job, or even concurrent negotiations with the possibility to dynamically split the job during the negotiation.

To handle the complex types of negotiation scenarios, five different services are proposed as examples of coordination component services:

- **Outsrc** for *outsourcing* jobs by exchanging proposals among participants known from the beginning. This service is always present in a negotiation; it is the head of the negotiation, the one which starts the negotiation from the point of view of the initiator participant. It has the possibility to see the whole sequence graph and use all the verbs and attributes of the *Negotiation Object*;
- **Insrc** for *insourcing* jobs by exchanging proposals among participants known from the beginning. This service is always present in a negotiation; it is the one which starts the negotiation from the point of view of the invited participant;
- **Block** service for assuring that a task is entirely subcontracted by the single partner;
- **Split** service manages the propagation of constraints among several slots, negotiated in parallel and issued from the split of a single job;
- **Broker**, a service automating the process of selection of possible partners to start the negotiation.

The Coordination Services level (CS) has two main functions:

- Mediation of the transition between the negotiation image at the Negotiation Manager level and the image at the Middleware level;
- Allowing the implementation of various types of appropriate behaviour in particular cases of negotiation. Consequently, each service is considered to correspond to a particular negotiation type.

On this level, interoperability is provided by offering a full-generic coordination framework for the participants to negotiate. To develop this structure, a succession of phases that are specific to different stages of negotiation (initialization, negotiation, contract adoption) is proposed.

The advantage of this negotiation process structure is that it allows a proper identification of the elements forming the object of coordination, of the dependencies that are possible among the existing negotiations within the Cloud-based environment, and the modality to manage these negotiations at the Coordination Services level.

Depending on the behaviour they implement, each of the services has its own features and own way of reading the image of its graph and perform on it. These services are able to evaluate the received proposals, and (if these are valid) able to reply with new proposals constructed based on their particular coordination constraints [79]. The coordination problems managing constraints between negotiations can be divided into two distinct classes of services: Coordination services in a closed environment and Coordination services in an open environment.

3.3.1. Coordination Services in a Closed Environment

These services build their images on the negotiation in progress and manage the coordination constraints according to information extracted only from their current negotiation description.

These coordination services have the ability to connect other services in the ongoing negotiation, and manage the coordination constraints of a single negotiation at a time. They manage the flow of information among the enterprises participating on the same negotiations.

Four services were defined to have this capacity: **Outsrc** (section 3.4.1), **Insrc** (section 3.4.2), **Block** (section 3.4.3) and **Split** (section 3.4.4).

3.3.2. Coordination Services in an Open Environment

These services also build their images on the negotiation in progress, but manage the coordination constraints according to the available information in data structures representing certain characteristics of other negotiations currently ongoing into the system.

The main functionality of the services is to manage the flow of information coming from outside of the current negotiation, in particular, information provided by other negotiations where the considered enterprise is simultaneously involved. These coordination services manage the flow of information within the same enterprise but for multiple negotiations.

As the middleware does not provide communication mechanisms among the instances of the components involved in different negotiations, and also as the coordination services in open environment cannot invite other components in the negotiation, instead these components shall have the possibility to connect to private data structures.

This way of communication is known as *blackboard*, and consists in a data structure for each type of component in the open environment. With the support of these data structures, components are able to read the information from the exterior, in particular the information about other existing negotiations.

The services are also able to write in these data structures in order to update the information on the negotiation in which they are involved. The information stored in these structures is necessary to establish dependency relationships among multiple negotiations.

An example of information needed is represented by the identifiers of the participants involved in negotiations among which there is an exchange of tasks. This information allows the management of the exchange procedure among the considered negotiations, a sort of directory service.

A key aspect of this approach is the opportunity to specialize the services able to coordinate this type of dependencies. For this purpose, the **Broker** service (section 3.4.5) is proposed to allow the coordination of constraints among several different negotiations in parallel.

3.4. Negotiation Component Services

The formal model to settle and to manage the coordination rules of one or more negotiations that may take place in parallel includes the basic concepts of the model (section 3.1.1), as well as the usage of the metaphor of Interaction Abstract Machines (IAMs, section 3.1.2), the Program Formula to define the methods to manage the parallel evolution of multiple negotiations (section 3.2), and the description of the Negotiation Process itself (section 3.6).

The proposed negotiation model can be used for describing a particular coordination service. To represent a negotiation, a graph structure was chosen where the nodes describe the negotiation phases that are present at that time instance (i.e., the negotiation proposals sent up to that moment in terms of status and attributes negotiated) and the edges express the precedence relationship between negotiation phases.

It is assumed that each of the atoms implicitly contains in its state particles for processing different proposals in terms of mathematical operations or string manipulations.

3.4.1. Outsrc Negotiation Service

The *Outsrc* service can be seen as the main component of a negotiation. Generally, the automatic negotiation process is initiated by creating an instance of this component starting from the initial negotiation object.

Such a component must build the negotiation graph by following the high-level negotiation features (assessment and creation of proposals and rules of coordination). Beyond these functionalities, the *Outsrc* service has to interpret and check the constraints on the negotiation, which are set up in the *Negotiation Object* and *Negotiation Framework* data structures.

The information provided by the *Negotiation Object* structure regarding the possible values of the attributes to be negotiated allows the *Outsrc* service to easily check whether the received proposals concern the attributes negotiated in the current negotiation and if they are associated to the values of the intervals specified.

As an example, if the *Negotiation Object* requires that the price should be "cost <= 10", the *Outsrc* service can stop the continuation of the negotiation in the phases associated to the nodes where the proposals are outside the interval that was determined as constraint (e.g., for nodes containing "`assert(12, Job, cost, =11)`").

The *Outsrc* service can provide information to the other components about the limitations imposed by the *Negotiation Object*, as well as check if the associated value confirms the constraints imposed by the Negotiation Manager.

Actually, the *Outsrc* service also works at middleware level, the has also the function of administrating the transactional aspects of the negotiation. This service is seen like a coordinator with the task of concluding the agreement among the component instances participating in the same negotiation.

Another functionality of the *Outsrc* service is to interpret and execute the tactics specified in the *Negotiation Framework* structure by connecting a combination of different instances of the other services.

Therefore, the *Outsrc* service concerns the aspects imposed by the strategic coordination, specified at the level of the *Negotiation Agent*, and their implementation at the level of the Coordination Services.

3.4.2.Insrc Negotiation Service

The *Insrc* service administrates the negotiation from the invited participant's side, deciding whether to accept a proposal; some of its features are similar to those of the *Outsrc* service. The differences come from the fact that this component does not have a complete picture of the negotiation; its vision of the negotiation graph is limited to the data referring only to its direct negotiation with the *Outsrc* service. Another difference is that the *Insrc* service has no information about what is negotiated nor about the constraints of its Negotiation Manager at the beginning of the negotiation; instead, despite of its limited view of the negotiation graph, it has a rich interaction on the new aspects required in the negotiation.

Hence, depending on the attributes required by the negotiation initiator, the *Insrc* service is able to progressively build the data structures describing the Negotiation Manager's preferences on the *Negotiation Object* and on the negotiation process.

3.4.3.Block Negotiation Service

The *Block* service is used in the negotiations where the task must be executed in its totality by a single participant of the negotiation process. Its main role is to mediate the negotiation between the enterprise that initiated the negotiation and all other enterprises that are invited to the current negotiation.

The mediation is performed with the goal of establishing a contract regarding the execution of the whole task by a single participant. In this way, a service called "Block" is set to manage the constraint of not splitting the subcontracted task in different slots, but to maintain the unity of a single task block.

An example of the interactions on the *Block* service may be provided by considering a negotiation among an initiator enterprise A and other two guest enterprises B and C acting in the same industrial area.

In this case, the negotiation begins in A by the initialisation of an *Outsrc* service, which invites the *Block* service to the negotiation. Subsequently, *Block* connects to the *Insrc* services of each partner B and C, and will coordinate bilateral negotiations with them simultaneously. As soon as all services are connected, the interaction process between the participants may begin. During this process, the Negotiation Manager of each company involved in the negotiations starts generating and exchanging proposals and counter-proposals for the task at hand.

The negotiation ends when the enterprise A reaches an agreement with one of the partners (e.g., enterprise B) regarding the set of attributes that describe the task being negotiated. At the same time, enterprise A ends the negotiation with enterprise C, this coordination being provided by the service *Block*. It should be noticed that the negotiation may also end without reaching an agreement (e.g., a time limit set for the negotiation has passed or the two partners B and C are no longer interested in the negotiation).

Hence, the sequence of actions to be modelled is:

- Negotiation initialization: Enterprise A invites enterprises B and C to the negotiation;
- Processing of the negotiation proposals;
- Implementation of the constraint to accept the task as a whole (i.e., the constraint regarding the size of the task contracted);
- Processing of the messages to accept a proposal;
- Processing of the messages to reject a proposal;
- End of the negotiation.

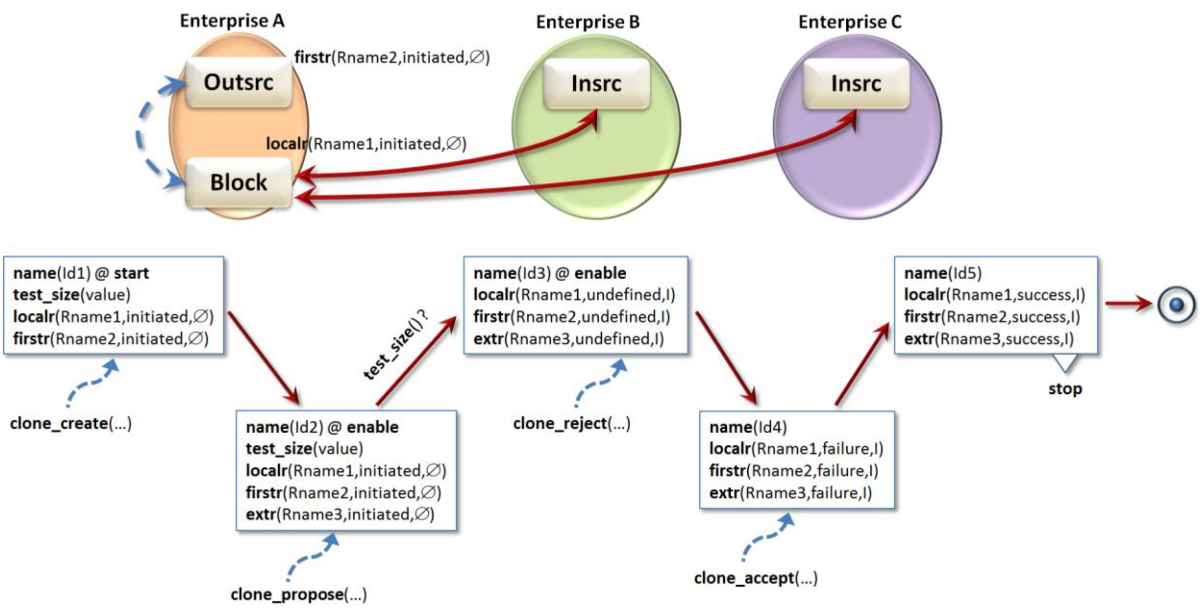


Figure 3.3: The Block negotiation service

It is assumed that for each partner involved in a negotiation there is a negotiation sequence. By using a *Program Formula* (section 3.2), all methods that model the entire negotiation process that must be managed by a certain sequence are defined. This behaviour was identified using a particular negotiation sequence named s_{Block}.

Figure 3.3 depicts the main transitions of the IAM for describing the *Block* negotiation service. This picture models the actions enumerated above, assuming that the first atom of the negotiation sequence s_{Block} initially contains the following particles:

name(Id);

start;

localr(Rname1,initiated,∅) : the representation particle of the sequence *Block*;

firstr(Rname2,initiated, ∅) : the representation particle of the sequence *Outsrc*;

test_size(value) : the particle that contains the size of the task fixed by enterprise A.

Assuming that the sequence s_{Block} sees all the bilateral negotiations, this sequence will participate in the negotiation from the beginning, i.e., from the moment when invitations are transmitted to the potential participants.

Method #1 starts the sequence by stating that for each event **clone_create**(Id, New_Id, Msg), a new atom is generated:

name(Id) @ **start** @ **clone_create**(Id, New_Id, Msg) @ {**string_create**(Msg, Rname, Type)} <>- (**start** @ **name**(Id)) & (**freeze** @ **name**(New_Id) @ **create**(Rname, Type))

The particle {**string_create**(Msg, Rname, Type)} will call a local computational particle to determine in the variable Msg the values of the parameters of the particle **create**(Rname, Type).

Method #2 generates a new representation particle **extr**(Rname,initiated,∅) that represents the image of the new sequence over this new negotiation phase:

freeze @ **create**(Rname, type) <>- **extr**(Rname, initiated, ∅) @ **enable**

The control particle **start**, that is only present in the first atom, limits the introduction of events **clone_create** to the first atom. This way, all of the cloned atoms starting from the first atom will be negotiation phases distributed between enterprise A and only one other participant.

Methods #3 and #4 manage the dependence of the size of the task negotiated, according to the constraint stated by the *Block* service.

Starting from the newly created atoms, the Method #3 states that the negotiation will be continued with the newly received proposals:

name(Id) @ **enable** @ **clone_propose**(Id, New_ Id, Msg) @ {**string_propose**(Msg, Rname, Content)} <>- (**enable** @ **name**(Id)) & (**freeze** @ **name**(New_ Id) @ **propose**(Rname, Content))

The proposal will result in the evolution of the new negotiation phase, but only if the constraint imposed on the task size is satisfied – the computational particle {**substring**(value,Content,true)} verifies this constraint on the Method #4:

freeze @ **test_size**(value) @ **localr**(Rname1, S1, I1) @ **firstr**(Rname2, S2, I2) @ **extr**(Rname3, S3, I3) @ **propose**(Rname, Content) @ {**substring**(value, Content, true)} @ {**construct**(I1, Content, I)} <>- **enable** @ **localr**(Rname1, undefined, I) @ **firstr**(Rname2, undefined, I) @ **extr**(Rname3, undefined, I)

Methods #5 to #9 model the dependencies regarding the status.

Method #5 states that in all valid negotiation phases (i.e., those having the control particle **enable**) the negotiation partners may accept the current proposal:

name(Id) @ **enable** @ **clone_accept**(Id, New_ Id, Msg) @ {**string_accept**(Msg, Rname)} <>- (**enable** @ **name**(Id)) & (**freeze** @ **name**(New_ Id) @ **accept**(Rname))

In the event that the two partners reach an agreement, in the newly created atom the representation particles for a single partner will have the status *success* – Methods #6 and #7:

freeze @ **localr**(Rname1, S1, I) @ **firstr**(Rname2, S2, I) @ **accept**(Rname2)<>- **localr**(Rname1, success, I) @ **firstr**(Rname2, success, I) @ **waiting**

freeze @ **extr**(Rname3, S3, I) @ **accept**(Rname3)<>- **extr**(Rname3, success, I) @ **waiting**

A new control particle (**waiting**) was introduced to preserve the negotiation atom only for events of type **clone_accept** or **clone_reject** on Methods #8 (and #10):

name(Id) @ **waiting** @ **clone_accept**(Id, New_ Id, Msg) @ {**string_accept**(Msg, Rname)} <>- **name**(Id) @ **freeze** @ **accept**(Rname)

Method #9 states that an agreement is reached only if all three representation particles have the status *success*:

localr(Rname1, success, I) @ **firstr**(Rname2, success, I) @ **extr**(Rname3, success, I) @ ^**stop** <>- **ready**(I)

In this situation, all other negotiation atoms are instructed to stop the negotiation (the particle **stop** is introduced through the broadcasting mechanism in all atoms of the negotiation sequence s_{Block}).

Methods #10 and #11 regard the event of type **clone_reject**, and Method #12 concerns the message of type **reject**.

Method #10 models the transformation of the event **clone_reject** in a negotiation atom where one of the participants has made a proposal of type *accept* in the current negotiation phase (the particle **waiting** was created in the method processing a message of type *accept* – Method #7):

name(Id) @ **waiting** @ **clone_reject**(Id, New_ Id, Msg) @ {**string_reject**(Msg, Rname)} <>- **freeze** @ **name**(New_ Id) @ **reject**(Rname)

Method #11 models the transformation of the event **clone_reject** into a negotiation atom where temporarily the proposal made is neither accepted nor rejected:

name(Id) @ **enable** @ **clone_reject**(Id, New_ Id, Msg) @ {**string_reject**(Msg, Rname)} <>- **freeze** @ **name**(New_ Id) @ **reject**(Rname)

When processing the message **reject**, the author chose to preserve the negotiation phase associated to the current atom by modifying all statuses to *failure* and preventing all possible evolutions of the atom through the use of all control particles – Method #12:

freeze @ **localr**(Rname1, S1, I) @ **firstr**(Rname2, S2, I) @ **extr**(Rname3, S3, I) @ **reject**(Rname) <>- **localr**(Rname1, failure, I) @ **firstr**(Rname2, failure, I) @ **extr**(Rname3, failure, I)

Method #13 is a rule that completely destroys all the negotiation atoms visible through the sequence s_{Block}, except the negotiation atom that contains the agreement regarding the negotiation (the particle **stop** has been created through this atom on Method #9):

stop <>- **#t**

This example presents how the proposed model can describe a particular coordination negotiations service.

3.4.4. Split Negotiation Service

The *Split* service is used in negotiations where there is the possibility of fragmentation of the task in two parts for two different contractors. Its main function is to coordinate the connections between the complete descriptions of the task accomplished by an instance of an *Outsrc* service in enterprise A and the proposals concerning both sides of the task accomplished in two instances of the *Insrc* service, one for each interacting enterprise B and C.

When the *Split* service is connected, it becomes an intermediary between the *Outsrc* service in enterprise A and the *Insrc* services of the enterprises B and C.

The *Split* service raises some constraints. The constraints on the negotiated task are established by the *Outsrc* service. When one of the *Insrc* services makes a proposal, the *Split* service is able to build the proposal for the other *Insrc* service, so that the task proposed by the *Outsrc* service is entirely subcontracted.

The *Split* service is invited in the negotiation by the *Outsrc* service. In Figure 3.4, the implementation of the negotiation for the initiator enterprise A is achieved through *Outsrc* and *Split* services. For each of the invited enterprises, the implementation of the negotiation is realized through their *Insrc* services.

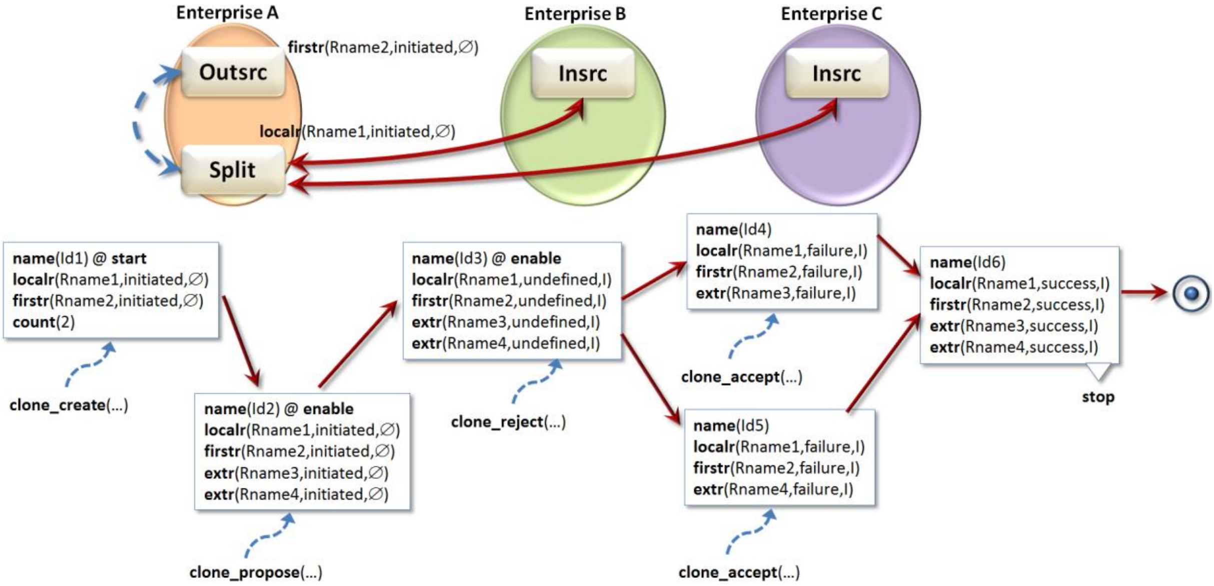


Figure 3.4: The *Split* negotiation service

Furthermore, the two *Insrc* services coordinate only the interactions with the *Outsrc* service of enterprise A. Thus, enterprises B and C are not necessarily aware of being in competition and the synchronization between their proposals is made by the *Split* service.

The management of the constraints resolution is thus delegated to the *Split* service. This service implements generic mechanisms for handling constraints and ensure the propagation of these constraints.

The negotiation ends the moment when enterprise A is satisfied with the proposals made by enterprises B and C. The two negotiations end simultaneously by a single agreement, which will result in two contracts.

Hence, the sequence of actions to model the behaviour of the s_{Split} negotiation sequence with $view(s_{Split}) = (p_1, N, R_{Split})$ is:

- Initialization of the negotiation: Enterprise A invites enterprises B and C to the negotiation;
- Processing of the negotiation proposals;
- Implementation of the constraint to accept the task as a whole (i.e., the constraint regarding the size of the task contracted);

- Processing of the messages to accept a proposal;
- Processing of the messages to reject a proposal;
- End of the negotiation.

It is assumed that for each partner involved in a negotiation there is a negotiation sequence. By using a *Program Formula* (section 3.2), all methods that model the entire negotiation process that must be managed by a certain sequence are defined.

Figure 3.4 depicts the main transitions of the IAM for describing the *Split* negotiation service. This picture models the actions enumerated above, assuming that the first atom of the negotiation sequence s_{Split} initially contains the following particles:

name(Id);

start;

localr(Rname1,initiated,∅)   : the representation particle of the sequence *Split*;

firstr(Rname2,initiated, ∅)   : the representation particle of the sequence *Outsrc*;

count(X) : the particle used to split the task. In this example the split of the task is done into two parts, so X=2.

Method #1 starts the sequence by stating that for each event **clone_create**(Id, New_Id, Msg), a new atom is generated:

count(X) @ **name**(Id) @ **start** @ **clone_create**(Id, New_Id, Msg) @ {X != 0} @ {**string_create**(Msg, Rname, Type)} <>- (**start** @ **name**(Id) @ **count**(X)) & (**start** @ **freeze** @ **name**(New_Id) @ **create**(Rname, Type) @ **count**(X))

The particle {**string_create**(Msg, Rname, Type)} will call a local computational particle to determine in the variable Msg the values of the parameters of the particle **create**(Rname, Type).

Method #2 generates a new representation particle **extr**(Rname, initiated, ∅) that represents the image of the new sequence over this new negotiation phase:

count(X) @ **freeze** @ **create**(Rname, type) <>- **extr**(Rname, initiated, ∅) @ **count**(--X)

Considering that our aim is to split the task in two parts, the atoms which can start the negotiation are those containing the **count**(X) particle. Thus, we can obtain two particles of external representation for two new partners with Method #3:

count(0) @ **start** <>- **enable**

Methods #4 to #6 deal with the exchange of proposals and ensure dependencies among the proposals made on the execution of the two parts of the *split_competition rule* task.

Method #4 introduces a new proposal within the system through the **clone_propose** event**.** The negotiation will develop by taking into account this proposal only on the atoms containing this **enable** control particle:

name(Id) @ **enable** @ **clone_propose**(Id, New_ Id, Msg) @ {**string_propose**(Msg, Rname, Content)} <>- (**enable** @ **name**(Id)) & (**freeze** @ **name**(New_ Id) @ **propose**(Rname, Content))

The shaping of the **propose** particle is different depending on the participant which made the proposal. If the initiator participant (enterprise A) was the one which made the proposal, this will refer to the whole task and will be split by the **construct_split** computational particle in order to build proposals for each invited participant, modelled in Method #5:

freeze @ **localr**(Rname1, S1, I1) @ **firstr**(Rname2, S2, I1) @ **extr**(Rname3, S3, I3) @ **extr**(Rname4, S4, I4) @ **propose**(Rname2, Content) @ {**construct_split**(I1, I3, I4, I5, I6, I7)} <>- **enable** @ **localr**(Rname1, undefined, I) @ **firstr**(Rname2, undefined, I) @ **extr**(Rname3, undefined, I)) @ **extr**(Rname4, undefined, I)

In case a participant invited in the negotiation (enterprises B or C) makes the proposal, this will be handled through the **solve_split** computational particle in order to build a proposal for the other participant concerning the execution of the remaining part of the task, as defined in Method #6:

freeze @ **localr**(Rname1, undefined, I1) @ **firstr**(Rname2, undefined, I1) @ **extr**(Rname3, undefined, I3) @ **extr**(Rname4, undefined, I4) @ **propose**(Rname2, Content) @ {**solve_split**(I1, I3, I4, I5, I6, I7)} <>- **enable** @ **localr**(Rname1, undefined, I) @ **firstr**(Rname2, undefined, I) @ **extr**(Rname3, undefined, I)) @ **extr**(Rname4, undefined, I)

The modelling of a proposal acceptance is the same as in the *Block* example.

Method #7 states that in all valid negotiation phases (i.e., those having the control particle **enable**) the negotiation partners may accept the current proposal:

name(Id) @ **enable** @ **clone_accept**(Id, New_ Id, Msg) @ {**string_accept**(Msg, Rname)} <>- (**enable** @ **name**(Id)) & (**freeze** @ **name**(New_ Id) @ **accept**(Rname))

In the event that the two partners reach an agreement, in the newly created atom the representation particles for a single partner will have the status *success* – Methods #8 and #9:

freeze @ **localr**(Rname1, S1, I) @ **firstr**(Rname2, S2, I) @ **accept**(Rname2) <>- **localr**(Rname1, success, I) @ **firstr**(Rname2, success, I) @ **waiting**

freeze @ **extr**(Rname3, S3, I) @ **accept**(Rname3) <>- **extr**(Rname3, success, I) @ **waiting**

A new control particle (**waiting**) was introduced to preserve the negotiation atom only for events of type **clone_accept** or **clone_reject** on Methods #10 (and #13):

name(Id) @ **waiting** @ **clone_accept**(Id, New_ Id, Msg) @ {**string_accept**(Msg, Rname)} <>- **name**(Id) @ **freeze** @ **accept**(Rname)

The difference from the *Block* service is that in the *Split* service there are two ways of concluding an agreement. In the first case, all four particles of representation must be in a *success* status – under the *split_success* rule (Method #11).

localr(Rname1, success, I) @ **firstr**(Rname2, success, I) @ **extr**(Rname3, success, I) @ **extr**(Rname4, success, I) @ ^**stop** <>- **ready**(I)

In the second case, an agreement can conclude if one of the external representation particles finds itself in a *success* status, but the concluded agreement refers to the whole task – in accordance with the *competition* rule (Method #12):

localr(Rname1, success, I) @ **firstr**(Rname2, success, I) @ **extr**(Rname3, success, I) @ ^**stop** <>- **ready**(I)

In both cases, when an agreement is concluded, all other atoms are notified to stop the negotiation (the **stop** particle is introduced by the mechanism of broadcasting through all atoms composing the s_{Split} negotiation sequence).

Methods #13 and #14 regard the event of type **clone_reject**. In this case, two cases are shaped: firstly, an atom is created following the acceptance of a proposal (atom created by Method #9) and secondly, an atom is created following a new proposal (atom created by Method #4).

Method #13 models the transformation of the event **clone_reject** in a negotiation atom where one of the participants has made a proposal of type *accept* in the current negotiation phase (the particle **waiting** was created in the method processing a message of type *accept* – Method #9):

name(Id) @ **waiting** @ **clone_reject**(Id, New_ Id, Msg) @ {**string_reject**(Msg, Rname)} <>- **freeze** @ **name**(New_ Id) @ **reject**(Rname)

Method #14 models the transformation of the event **clone_reject** into a negotiation atom where temporarily the proposal made is neither accepted nor rejected:

name(Id) @ **enable** @ **clone_reject**(Id, New_ Id, Msg) @ {**string_reject**(Msg, Rname)} <>- **freeze** @ **name**(New_ Id) @ **reject**(Rname)

The processing of the **reject** type message is done according to the partner who decides to decline a proposal. If one of the invited participants decides to stop the negotiation in the current atom, this will be visible in the atom managed by the s_{Split} sequence only at the level of the representation particle of the participant who sent the rejection (Method #15):

freeze @ **extr**(Rname3, S3, I) @ **reject**(Rname3) <>- **extr**(Rname3, failure, I)

If the initiator participant sent the refusal, the negotiation atom is completely dissolved (Method #16):

freeze @ **localr**(Rname1, S1, I) @ **firstr**(Rname2, S2, I) @ **reject**(Rname2) <>- **#t**

Similarly, the negotiation stops in the atoms where the two participants are invited in a *failure* status (Method #17):

extr(Rname3, failure, I) @ **extr**(Rname4, failure, I) <>- **#t**

The negotiation also stops in the atoms containing the **stop** particle (Method #18):

stop <>- **#t**

3.4.5.Broker Negotiation Service

The Broker component provides the mechanism for selection of potential partners. The Broker component functionality resides in seeking partners who can perform a particular kind of task. Its image on the graph is limited to data contained in the node to which it was connected, but, in addition, Broker component has access to the database of the alliance partners (including name, address and description of the type of job that can be done). Starting from the graph and more specifically from the node to which it was connected to, the Broker component is able to extract attributes that characterize the task and further validate the existence of these attributes in the proposals made by potential partners invited in the negotiation.

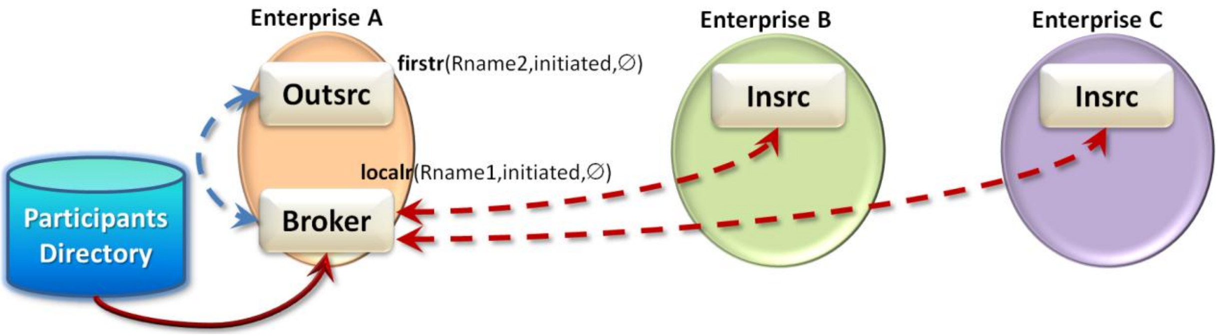


Figure 3.5: The *Broker* negotiation service

Figure 3.5 presents a possible case scenario. As in previous scenarios, the considered negotiation is a multi-bilateral and multi-attribute negotiation. The only difference is that the Manager does not specify the potential partners and delegates this task to the infrastructure. Thus, partners will be known only during negotiation and not at the initiation of it.

This selection process of partners is delegated to a component, called *Broker*. The selection process is based on the description of the negotiated task and the observation of the alliance partner database. So, in Figure 3.5, the *Outsrc* component initiates the negotiation by demanding the *Broker* component to find workshops able to fulfil a particular task. Once the *Broker* component finds potential partners to meet the task characteristics, the negotiation starts from the initial point of the previous scenarios.

3.5. Negotiation Process Structure

The Coordination Services provide the main features of this negotiation process: distribution and parallelism. Distributed coordination schemes, implemented through the services, can be executed in parallel by the proposed infrastructure.

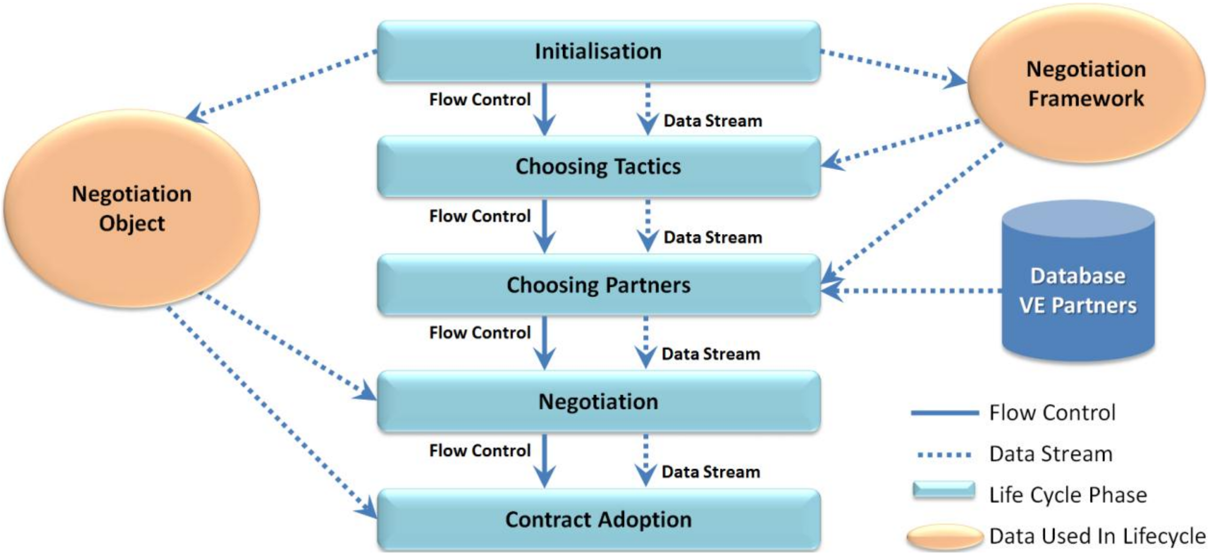


Figure 3.6: The structure of the Negotiation process

The Negotiation process in Figure 3.6 is divided into five parts:

- **Initialization**: The Negotiation Manager initiates the outsourcing of a task, defining and communicating through the infrastructure the properties and the constraints of the negotiation object and the negotiation framework. The negotiation process begins by creating an instance of the *Outsrc* service. This instance will initiate other stages of negotiation, based on the constraints provided by the Negotiation Manager (i.e., the invitation of other coordination services, e.g., *Insrc*, *Broker*). Moreover, this instance will conduct negotiations in terms of construction and evaluation of proposals for outsourcing proposed task;
- **Choosing tactics**: Using the negotiation tactics specified in the framework, the coordination is decomposed into several coordination schemes. Three tactics correspond to three coordination schemes: Block, Split and Transport;
- **Choosing partners**: The possible choices of partners are:
    - <u>Among known partners</u>: The Manager initiating the outsourcing can specify any constraints on the set of possible contractors. To do this, the manager uses the description of the job to be outsourced and also the database partners within

the Cloud-based environment and/or the different adhesion contracts they signed;

  - Among unknown partners: in this case, the entire research activity of the potential partners is managed by the infrastructure through the *Broker* service.

- **Negotiation**: At this stage, during the exchange of proposals, the negotiation object evolves according to the constraints imposed by the manager on the negotiated attributes of the outsourcing task. The objective of the negotiation stage is to build an Instantiated Negotiation Object (e.g., a negotiation object whose attributes have been accepted by all partners) from the initial specification of the negotiation object. After that, this object will be used to establish a contract;

- **Contract Adoption**: In this final stage, the negotiation object has fixed values. The Manager validates the result of negotiation and makes contact with other partners within the Negotiation Environment. Thus, the Manager may decide:

  - To restart or to suspend negotiations;
  - To enable the contracting process that will state an agreement.

The negotiation process involves several parties (for several bilateral negotiations), each having different criteria, constraints and preferences that determine their individual areas of interest. Criteria, constraints and preferences of a participant are partially or totally unknown to the other participants. The job under negotiation is described as a multi-attribute object. Each attribute is related to local constraints and evaluation criteria, but also to global constraints drawing dependencies with other attributes.

In conclusion, the proposed architecture manages in a decentralized manner the coordination of multi-phase negotiations on a multi-attribute object and among several participants, featuring:

The definition of the negotiation process structure: participants, interaction protocol, negotiation protocol, tactics and coordination services, the negotiation object and the negotiation strategies;

Modelling of all negotiations in which a participant may be involved in the form of a set of bilateral negotiations;

Modelling of the coordination among the negotiations based on a set of coordination services and on the synchronization mechanisms at the middleware level.

3.6. Description of the Negotiation Process

According to the proposed approach regarding the negotiation, the participants to a negotiation may *propose* offers and each participant may decide in an autonomous manner to stop a negotiation either by *accepting* or by *rejecting* the offer received.

Also, depending on its role in a negotiation, a participant may *invite* new participants to the negotiation. To model this type of negotiation, we will make use of the previously defined particles and we will propose the methods to manage the evolution of these particles.

Through the use of the metaphor IAMs, the evolutions of the negotiation phases correspond to the evolutions at the atoms level. The evolution may be regarded as a process consisting of two stages: a *cloning* operation of the atom existent in the initial stage and a *transformation* operation within the cloned atom to allow for the new negotiation phase. The cloning operation is expressed by a set of methods involving the *event* particles and these methods are used to facilitate the evolution of the negotiation.

Four cloning methods are proposed, associated to the *event* particles to model the cloning of an atom where new *message* particles are introduced:

- The method **Propose** is associated to the *event* particle *clone_propose(Id, New_id, Msg)* and models the introduction of a new proposal *clone_propose* by one of the participants. This method is expressed as:

name(Id) @ enable @ clone_propose(Id, New_ id, Msg)<>- (enable @ name(Id)) & (freeze @ name(New_ id) @ propose(Rname, Content))

  As result, the atom identified by the particle *name(Id)* is cloned. The new proposal, contained in the particle *propose(Rname, Content),* will be introduced in the new atom *name(New_id)*.

- The method **Accept** is associated to the event particle *clone_accept(Id, New_id, Msg)* and models the case when one of the participants sent a message indicating acceptance (*clone_accept*) of a proposal. This method is expressed as:

name(Id) @ enable @ clone_accept(Id, New_ Id, Msg) <>- (enable @ name(Id)) & (freeze @ name(New_ Id) @ accept(Rname))

  As result, the atom identified by the particle *name(Id)* is cloned. The message to accept that proposal is contained in the particle *accept(Rname),* which will be introduced in the new atom *name(New_id)*.

- The method **Reject** is associated to the event particle *clone_reject(Id, New_id, Msg)* and models the denial of an older proposal (*clone_reject*) made by one of the participants. This method is expressed as:

name(Id) @ enable @ clone_reject(Id, New_ Id, Msg) <>- (enable @ name(Id)) & (freeze @ name(New_ Id) @ reject(Rname))

  As result, the atom identified by the particle *name(Id)* is cloned. The message of denial contained in the particle *reject(Rname)* will be introduced in the new atom *name(New_id)*.

- The method **Create** is associated to the event particle *clone_create(Id, New_id, Msg).* This method models the invitation of a new sequence (*clone_create*) made by one of the participants toward the distribution of the newly created negotiation phase. This method is expressed as:

  name(Id) @ enable @ clone_create(Id, New_Id, Msg) <>- (enable @ name(Id)) & (freeze @ name(New_Id) @ create(Rname, Type))

  As result, the atom *name(Id)* is cloned and a particle *create(Rname, Type)* is introduced in the new atom *name(New_Id)* that will subsequently generate a new representation particle for the new sequence that is participating to the negotiation.

The *message* particles participate in transformation methods that change the negotiation phase of an atom by replacing the representation particles of the negotiation sequences involved in the generation or in the receiving of the messages that are exchanged.

Next, four transformation methods are proposed:

- The transformation method associated to the particle *propose(Rname, Content)* contributes to the local evolution of a negotiation phase regarding the status and the attributes negotiated. This evolution takes place by replacing in the existing atom all representation particles that are involved (depending on the method) with the new particles that have the status changed to *undefined*, and the set of the negotiated attributes (*Issues*) contains the new proposal expressed in the *Content* of the message particle:

  freeze @ localr(Rname1, S1, I1) @ extr(Rname, S2, I1) @ propose(Rname, Content) <>- enable @ localr(Rname1, undefined, I) @ extr(Rname, undefined, I)

- The transformation method associated to the particle *accept(Rname)* leads to the local evolution of a negotiation phase regarding the status. The evolution is made by replacing in the existing atom the representation particles involved with the new particles whose status has been changed from *initiated* or *undefined* to *success*:

  freeze @ localr(Rname1, S1, I1) @ extr(Rname, S2, I1) @ accept(Rname) <>- localr(Rname1, success, I1) @ extr(Rname, success, I1)

- The transformation method associated to a particle *reject(Rname)* is similar to *accept(Rname)*, except that the evolution is made replacing in the existing atom the particles involved with particles whose status has been changed from *initiated* or *undefined* to *failure*:

  freeze @ localr(Rname1, S1, I1) @ extr(Rname, S2, I1) @ reject(Rname) <>- localr(Rname1, fail, I1) @ extr(Rname, fail, I1)

- The transformation method associated to the particle *create(New_r_name, Type)* contributes to the evolution of a negotiation phase regarding the number of the sequences that participate to this negotiation phase. This evolution is made by introducing in the corresponding atom a new representation particle:

```
freeze @ create(Rname, type) <>- extr(Rname, init, ∅) @ enable
```

  As soon as this sequence is invited to the negotiation, its status is *initiated* and its set of the negotiated attributes is the empty set.

The evolution of all negotiation atoms and phases take place in parallel. By modelling this parallel evolution in a dynamic environment, the proposed model allows us to describe and manage negotiations on multi-attribute negotiation objects and involving multiple participants. The methods proposed above allow us to model different negotiation scenarios from single linear negotiation to concurrent and dependent multi-proposal negotiations.

3.7. Agent-based Negotiation Rules Engine

The negotiation model and its transitions shall be stored in the framework in the shape of business rules with the support of a rule engine. This was the most appropriate and flexible solution to store information and build a decision-support system that learns from previous negotiations.

Rules are pieces of information that are interpreted and their usage and storage constitutes a knowledge base for supporting new negotiations, in a way that static programming would have many difficulties.

The negotiation mechanism (Negotiation Services) that was built to implement this negotiation model shall comprise a set of rules defined in Drools (see section 2.5.3.3).

The architecture of the proposed framework will include an agent-based platform which will detect the changes in the interoperability environment and trigger the negotiation rules engines towards determining if the proposed change should be adopted by the other parties as well. The intelligent agents shall use the knowledge of the rules engine to determine suggested decisions.

4. FRAMEWORK FOR SUSTAINABILITY OF EI

“

We climb the highest mountains,
We'll make the desert bloom,
We're so ingenious we can walk on the moon.
But when I hear of how the forests have died,
Saltwater wells in my eyes.

”

Saltwater – Julian Lennon

This research contributes to the improvement of the sustainability of Enterprise Interoperability by proposing a framework called NEGOSEIO, which stands for NEGOtiations for Sustainable Enterprise InterOperability, and aims to tackle the gaps identified on section 1, and improve the sustainability of the interoperability in business-to-business interoperations, focusing particularly on the ESA-CDF environment [121] as a validation business case.

This solution combines a methodology and framework that have at its core the need for negotiations in the development of interoperability solutions. These negotiations can have multiple participants (e.g., the engineering domains, or a domain and its supply chain).

The proposed framework towards a SEI encompasses (Figure 2.8):

- A set of tools (e.g., questionnaires, surveys, data and documentation analysis) which describe and model the enterprise, capturing and publishing their interoperability details and requirements;
- The modelling of technology, business flows and human interactions, together with a set of services able to support enterprise interoperability operations;
- The detection of modifications in the advent of changes or arrival of new elements in the collaboration environment (e.g., partner, tool, supported platform, method);
- The availability of decision support and service-based mechanisms to assist in the negotiation of solutions for the identified interoperability changes.

The NEGOSEIO framework provides a set of services to handle the collaboration and to negotiate the interoperability solutions. These include the creation of negotiation mechanisms that analyse and reduce the impact of interoperability changes, not by avoiding them but by compromising these changes with the need of maintaining interoperability and with the need of reducing effort and downtime.

From this compromise, the resulting decisions may vary from implementing the changes, rejecting the changes, performing partial changes, delaying the changes, selecting new strategies or in a radical move, even determine that interoperability is not possible or viable.

The NEGOSEIO framework is composed by a multi-level set of services defined over a Cloud SaaS platform adopting the MDA, MDI, B-MDA and B-MDI paradigms (see section 2.1.2). These include middleware services that handle heterogeneity issues on the basic interoperability level (e.g., authentication, permissions, communications, syntax, session, data), and on top of these middleware services, a set of services developed to perform activities like management of data transactions, semantic interpretation, dynamic discovery of services, and implementation of the business model rules. This level stores its data (e.g., business-specific data, semantic ontologies, rules) on a cloud-based (IaaS) infrastructure modelled using standard reference models and data access [122].

It is based on a negotiation mechanism and a decision-support system for the negotiation of interoperability changes, which allows enterprises to take decisions over the option that will best suit their needs, resulting from the analysis of the required changes and motivations, benefits, opportunities and threats, and the resulting impact of proposed changes in terms of resources, effort, cost and time, with the purpose of reducing downtime and effort towards (re-)achieving interoperability. Adaptation, flexibility and scalability are core concepts for the methodology towards sustainable interoperability.

The proposed approach aims to work right from an earlier stage, determining the best solution to handle the existing problems, which means analysing and formalising the required changes, determining the pros and cons of each solution and then maturely and factually selecting the solution that best suits the purpose. Negotiations favour the analysis of alternative solutions, the adoption of new methodologies, models, semantics, or instead the creation of adaptors and translators, but especially, they motivate decisions supported by consensus of the involved parties.

The framework bases its interaction in the definition of a set of negotiation rules, supported by a negotiation model to formalise and register the negotiation interactions in a set of declarative rules, implemented in Drools (section 3.7).

Each enterprise works on its own premises, developing its own business concepts, trades, skills, functionalities and methods. Although the establishment and embracement of standards is becoming more frequent, most of the business knowledge is based on past experiences, other partners' knowledge and new ideas. When two or more enterprises settle in the establishment of a partnership, several of these ideas may collide, which means they need time to harmonise. The relation between the complexities in the interoperation between them and the time and effort that is necessary to solve them is not linear, which means they need to develop a way to optimise this time and effort, thus reducing the time where there is no operation.

The proposed solution combines a methodology and framework that have at its core the need for negotiations in the development of interoperability solutions. These negotiations can have multiple participants (e.g., the partners and the supervision of their prime contractor or customer).

4.1. Requirements

After determining the scope and problems detected, this research proceeds to identify the needs, gaps and difficulties, in order to improve the sustainability of the interoperability between the various engineering domains in a CDF study and overall between the CDF disciplines.

The adoption of the MDA, MDI, B-MDA and B-MDI principles shall represent a major advance in the interactions in the ESA-CDF, as some of the main problems regarding interoperability regard the lack of knowledge of the business itself [123].

While this is very important, one should not neglect another important matter regarding interoperability – Change management. The enormous investment (manpower and resources) that was (still is, and will be) necessary to define at full-scale the business models and to keep them updated may lead to another pitfall, which is the fear to change, to avoid knowledge updates.

Competitiveness is pressuring enterprises to build better solutions with fewer resources, following new trends and supporting new platforms and methodologies. On the other hand, broader knowledge dissemination with the rise of the Internet and new technologies are leading to the adoption of best-practices, which means legislation and regulations are updating more frequently and deeply, and demanding rapid compliance from enterprises. These frequent business changes shake all the interoperability links between the enterprises, leading to periods of adaptation and readjustment where business operation is not possible.

This macro scenario is also applicable on the ESA-CDF operation, where new technologies, materials, trends and policies drive the alternatives to perform the mission study. New tools, development frameworks, calculation methods, security measures and quality requirements inevitably affect the studies evolution. Even new engineering disciplines may themselves be created and defined to accommodate regulation and policies changes.

Hence, change management is itself a major issue regarding interoperability, as it affects greatly the other two pillars in the business continuity: efficiency/availability and model/knowledge. The urge to rapidly regain interoperability often leads to unfounded, poorly-chosen solutions, which lead to inefficiency and rework.

The pursuit for a sustainable Enterprise Interoperability must then comprise a dynamic, recurring and adaptable effort for tackling changes, supported by strong business knowledge (model). However, flexibility to submit to all changes is not always desirable. Complying with a new concept may alter the delicate balance of the whole interoperating network.

Therefore, it is essential to define formal negotiation mechanisms to deal with interoperability changes and factually support decisions on the best solution for compliance, to state that the benefit/cost ratio needs to support higher investment on a stronger interoperability, or even that interoperability in this case is not feasible or worthy.

Driven by the outcomes of this research, the requirements considered essential for improving the interoperability between partners in a business-to-business relationship are [124]:

- **Req#1:** Interoperability should be taken into consideration from the very foundations of any business application;
- **Req#2:** Interoperability regards three logical layers (middleware, coordination and business logic), and all these should be covered;
- **Req#3:** Each party should clearly define and model its core business so that other interoperating parties can understand it;
- **Req#4:** The behaviours and interactions between the interoperating parties should be clearly defined and modelled;
- **Req#5:** Data models for each party should be clearly defined and available to the interoperating parties. These models should be standardised as well as the procedures to access data;
- **Req#6:** Each party should model its definitions and knowledge into one or more business-related ontologies and share it with the other operating parties;
- **Req#7:** Systems and applications should be adaptable to accommodate interoperability changes in businesses.

These requirements reflect an absolute need to determine knowledge, to extrapolate this knowledge to the interactions between the business players, and to adapt to changes in this knowledge.

Adapting to changes is a task whose associated effort grows with the complexity of the change but also with the complexity of the already existing interoperability ecosystem (e.g., the number of interoperable players, the depth of the interdependent knowledge). While some of the referred changes are mandatory (by legislation or other regulation, or for compliance with a supplier or customer), there are others that are discretionary, or that include alternatives for their compliance.

While not being a requirement on its own for the sustainability of the interoperability, it is essential to include mechanisms that formalise the alternative ways to deal and achieve interoperability, to determine what is the best alternative and why. These are negotiation mechanisms.

Negotiation mechanisms allow the formalisation of new proposals and ways to achieve interoperability, determining what were the drivers that led to each decision; they also provide means to validate in the future if the decision was correct (meaning it can be reused in the future) or not (meaning this is an alternative to avoid in the future), hence supplying a decision support system which shall be most useful in a near future of complex interoperability definitions.

4.2. Methodology

One of the basic problems that affect interoperability is the detection of the problem itself. For that purpose, the first step of this methodology is to gather knowledge from the interacting parties.

4.2.1.Knowledge Gathering

To do so, the proposed NEGOSEIO methodology starts by submitting a set of assessments (e.g., workshops, interviews, questionnaires and focus groups) to the involved enterprise stakeholders in order to properly capture the enterprise's knowledge and determine the its requirements towards interoperability, in which can be called the cognitive model.

While the interoperating parties need to share information in order to interoperate, competitiveness means they need to maintain their property rights on their knowledge, meaning that a solution must convey the sharing of a "public" part of the knowledge while protecting the "private" part of the knowledge (see Figure 4.1).

This is an important concern, especially when dealing with businesses where secrecy **is** the core asset, like ESA-CDF. In such cases, there is the common belief that the safest place to store information is in human resources' minds. This is what can be called "unknowledge", as this information is rapidly forgotten. The author states that the best procedure is in fact a proper formalisation of the knowledge, accompanied by an Information System which deals with security issues and levels, and a set of policies that is able to determine promptly which information can be disclosed and which need to be protected.

Figure 4.1: The NEGOSEIO knowledge capture applied to ESA-CDF

Traditionally, this capture focused only the technological issues regarding interfaces and data formats, tools and ontologies. The author proposes that besides the capture of technology assets, it is as well important to capture the intangible business needs, e.g., the human interactions and behaviours, and the operational use of the technology – "business and people" (section 2.1).

A problem regarding knowledge capture that exists within SMEs but especially in very scientific and specialised businesses like the ESA-CDF regards the human factor. The fact that CDF is a typically human environment, where most activities regard creativity and brainstorming, may pose common fears that people may find that the knowledge capture may put their own position at stake in the company. Besides that, another factor may also occur which regards the trust in the captured knowledge. As an example, it is a known fact that some ESA-CDF engineers only trust the numbers, calculations and models they made themselves. Actually these are not new subjects, they were known concerns in the industrialisation era regarding operational labour, and this is mostly a psychological factor that will need to be resolved in the future.

4.2.2.Models Building

The second step of the proposed methodology is to model the captured business knowledge into MDA, MDI, B-MDA and B-MDI models. The MDA/B-MDA will define the solutions and architecture foundations, while the MDI/B-MDI will allow a flexible horizontal transformation of data towards interoperability on all MDA layers. The objective is to target the problem related to information loss. These models shall be developed in the SysML, UML, UEML and OCL language specifications, to describe the foundations for the business + people (B-MDA), design (MDA), and interoperability (MDI) [125]. This satisfies requirements Req#1, Req#3 and Req#4.

This modelling starts right from a conceptual abstract level (MDA Computation-Independent Model – CIM), where the functionalities, visions, business premises, methodologies and standards are defined. The B-MDA shall also model a Business-CIM (B-CIM) concerning the organisational process assets that shape the enterprise knowledge base (e.g., strategies, organisation responsibilities, balance scorecards, organograms, HR hierarchies, roles, authorisations, objectives, behaviours and targets for each role, rules, policies and processes).

The CIM is then vertically transformed into an intermediate layer (MDA Platform-Independent Model – PIM) which concerns terminologies, ontologies, algorithms, pseudo-code and architectures, and all structures that concern the enterprise, while still working independently of the underlying platform. The B-CIM shall as well be transformed into a Business-PIM (B-PIM) which handles the operational workflows, management plans, and how each HR role shall fit and contribute in the system. These models shall be specified in the shape of ontologies, knowledge bases and rules.

The final vertical transformation is from the MDA PIM to concrete computerised solutions (MDA Platform-Specific Model – PSM), which are the practical implementations of the enterprise models, and which may reach the level of writing application code. Likewise, the B-MDA B-PIM is transformed into a Business-PSM (B-PSM), resulting in e.g., rule and workflow engines, operational and deployment manuals, work permits, training and schedules.

The MDI allows the definition and understanding of how the interoperability between the enterprises may be defined on the different MDA/B-MDA abstraction levels, as seen on Figure 4.3.

4.2.3. Creation of Services and Rules

The next step of the methodology is then to implement the PSM in the shape of flexible services organised in Service Oriented Architectures (SOA). The reason for selecting this solution is due to its flexibility, adaptability and reusability, where services in a SOA may be reshaped, versioned, improved, updated, adapted and combined (choreographed and composed) to build more complex services [126]. Besides, SOA is by definition independent of the underlying technology, which makes it suitable to work closely with the MDA/B-MDA and MDI/B-MDI paradigms, and SOA may be used to encapsulate the information, thus enforcing the concepts for the definition of "public" knowledge and "private" knowledge.

Other technology that is growing in popularity and functionality is the definition of business rules that are afterwards interpreted, modified and generated accordingly to the business flows and decisions, and which afterwards serve as knowledge base for the information system (see section 4.5.5).

4.2.4. Deployment and Integration

Finally, distributed computing is a complement to the service flexibility in the implemented solutions. The successful business concept of cloud-computing is a very flexible way to deal with scalability, redundancy and security, both in terms of service deployment (the definition of the SOA in the shape of a Software as a Service – SaaS) and of the entire architecture deployment (the virtualisation of the infrastructures in the shape of Infrastructure as a Service – IaaS, based on a Platform as a Service – PaaS). The decision to develop the infrastructure and services over clouds was made to grant the solution with the benefits associated to this business model, providing cheap, fast, and scalable infrastructure and services.

This information from the SOA is then used to shape a set of cloud-based services that handle the foundations for interoperability of the environment. These services tackle several issues, e.g., moderation, collaboration, coordination, negotiation, and implement other necessary infrastructures, e.g., the business knowledge base, monitoring and control tools, negotiation support tools, and decision-support systems.

The framework is complete with the introduction of a negotiation mechanism. Negotiations start from the early MDA/MDI definitions, where the involved parties propose their interoperability, exposing their "public" Strengths, Weaknesses, Opportunities and Threats (SWOT). This first negotiation shall determine if the parties are suited to collaborate at all. Negotiations between already interoperating parties regard interoperability changes that are proposed by one of the enterprises. In this case, the outcome of the negotiations may vary from e.g., accepting the change, rejecting it, adopting new consensus solutions or even ending interoperability, as can be seen on Figure 4.2.

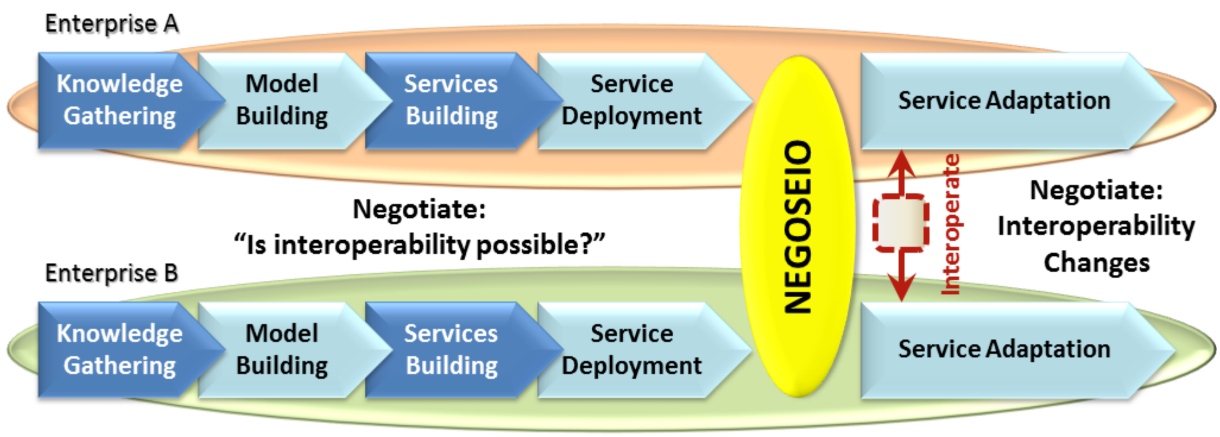


Figure 4.2: The NEGOSEIO methodology

Achieving a sustainable EI is then an integrated and interactive process of adaptation in a constant and iterative effort to validate the interoperable status, while maintaining the existing interoperability towards the surrounding environment.

The NEGOSEIO approach is to include negotiations in the definition of the interoperable services, included on the various MDA/MDI levels, as can be seen on Figure 4.3. The advocated best approach is to use negotiations as early in the process as possible, in order to achieve the best results in terms of best integration between enterprises.

This means that interoperability negotiations may start from the very enterprises' foundations, where it is easier to discuss business-related concepts and ideas, and then the progressive steps of transformation into lower-level models may also be synchronised to refine this interoperability, so that the overhead of transforming the concepts into code is performed by automation tools. A concrete example on how this is performed is specified on section 4.5.

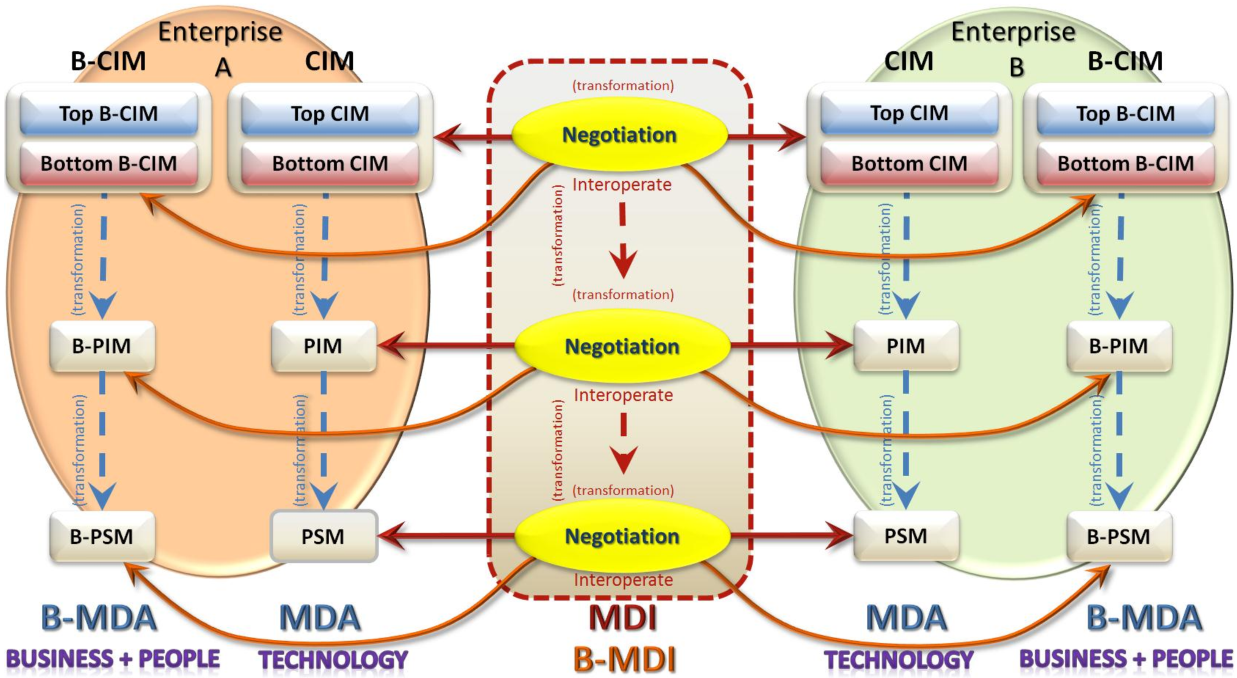


Figure 4.3: NEGOSEIO proposes multi-levelled Negotiations

4.3. Architecture

The proposed framework is composed by a multi-level set of services defined over a Cloud SaaS platform adopting a holistic view of the enterprise model using the MDA, MDI, B-MDA and B-MDI paradigms. These include middleware services that handle heterogeneity issues on the basic interoperability level (e.g., authentication, permissions, communications, syntax, session, data), and on top of these middleware services, a set of services developed to perform activities like management of data transactions, semantic interpretation, dynamic discovery of services, and implementation of the negotiation business model rules.

This level stores its data (e.g., business-specific data, semantic ontologies, rules) on a cloud-based (IaaS) infrastructure and PaaS platform, modelled using standard reference models and data access [21]. This satisfies requirements Req#5 and Req#6.

The NEGOSEIO architecture may be depicted on Figure 4.4 where it can be seen that the cloud-based services that handle the interoperable environment communicate with the enterprises (Negotiation Managers) via web-services deployed and accessed via an Enterprise Service Bus, allowing the separation and disclosure of public information and the protection of private knowledge.

The full architecture was designed to be flexible, hence the use of a rule engine to implement the dynamic negotiation flows and rules, a Service Oriented Architecture (SOA) for implementing the services that support the interoperability, and cloud management to support flexibility in the deployment and scalability. Moreover, data (e.g., business-specific data, semantic ontologies, rules) is persisted on cloud-based (IaaS) infrastructure and PaaS platforms, modelled using reference models.

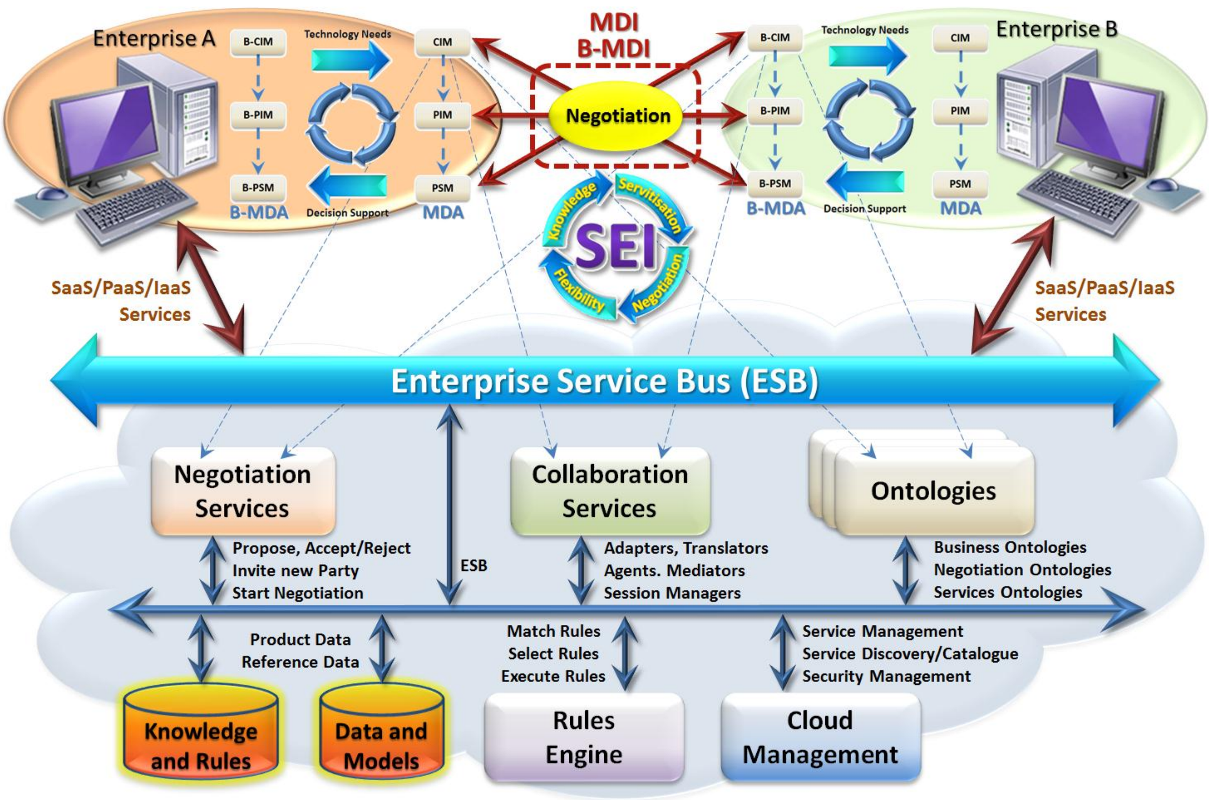


Figure 4.4: The NEGOSEIO conceptual architecture

Actually, data access, its models and data exchange can also be a concern for interoperability. Negotiation parameters, Ontologies and other entities rely on data modelling, specification and consistency and therefore the best way to define the data models and the data exchange is to use a standard. In this case the selected solution is to model the data for databases and data access using the standard ISO10303 STEP (STandard for the Exchange of Product model data) and EXPRESS language specification [119], [79]. The database infrastructure itself, as well as the whole model-driven framework infrastructure [65] shall also be implemented over Cloud, using an IaaS platform [67].

The framework bases its interaction in the definition of a set of negotiation rules, supported by a negotiation model to formalise and register the negotiation interactions in a set of declarative rules, implemented in the Drools rule engine, a business integration platform that runs on top of the Eclipse IDE (see Figure 4.5).

Negotiations allow a proper formalisation of the SWOT, while presenting the various alternative solutions. It also enables a clear definition of criteria (e.g., impact, downtime, cost, alignment with enterprise strategy, new markets) for selection of a particular solution, providing a strong justification for reaching mature choices on all three negotiation levels defined on section 3.3. This satisfies requirement Req#2.

```
rule "Calculate Result (NOTENOUGHDATA)"
    when
        result : com.negotiation.Result(resultPercentage < 50)
    then
        result.setFinalResult(com.negotiation.Result.NOTENOUGHDATA);
end

rule "Calculate Result (ENOUGHDATAREJECT)"
    when
        result : com.negotiation.Result(resultPercentage >= 50, numAnsweredPositives < numAnswered - 1 )
    then
        result.setFinalResult(com.negotiation.Result.ENOUGHDATAREJECT);
end

rule "Calculate Result (ENOUGHDATAACCEPT)"
    when
        result : com.negotiation.Result(resultPercentage >= 50, numAnsweredPositives >= numAnswered - 1)
    then
        result.setFinalResult(com.negotiation.Result.ENOUGHDATAACCEPT);
end
```

Figure 4.5: The NEGOSEIO negotiation rules in Drools

The framework is completed with the inclusion of a set of intelligent agents, implemented in the Java Agent DEvelopment Framework (JADE), which are responsible for receiving and learning the inputs from each of the enterprises and then detecting changes in the interoperable environment, triggering the negotiations (Figure 4.6). This satisfies requirement Req#7.

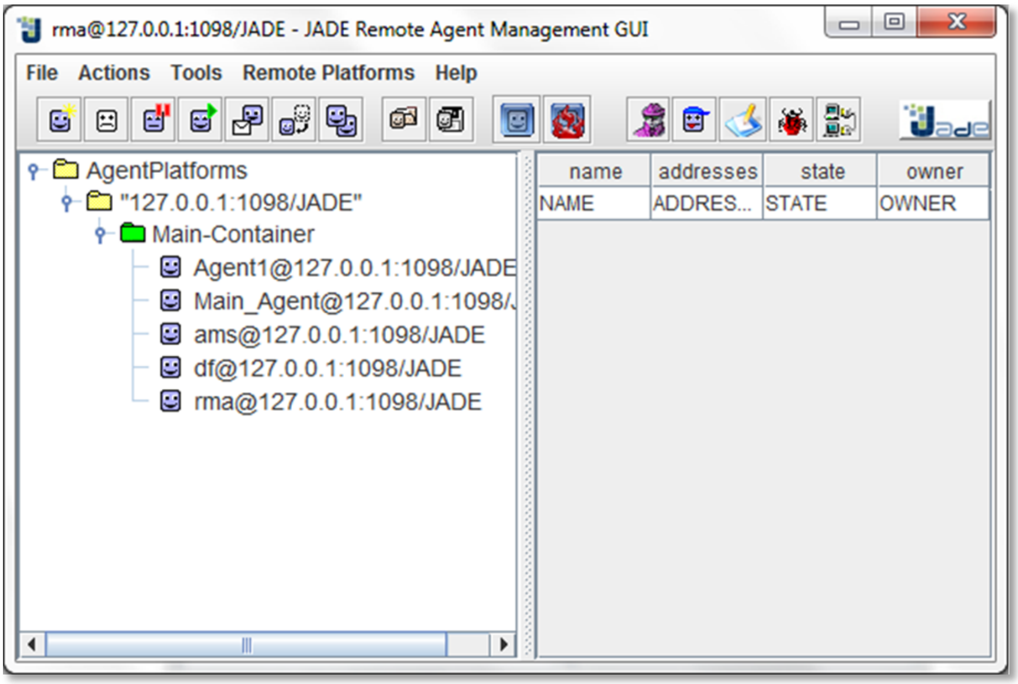


Figure 4.6: The NEGOSEIO Agent platform implemented in JADE

More than complying with the seven requirements elicited in section 4.1, this architecture provides a very flexible solution, not only because it provides the ability to capture the enterprise knowledge, but also because the architecture of services provides a good basis for delivering a modular structure, suitable for reuse and ad-hoc composition towards new solutions.

An important issue towards proper interoperation between parties concerns the ability to deal with the scalability of solutions. To handle this, the framework proposes an architecture implementation involving the Cloud Computing business model. Infrastructure as a Service (IaaS) services may be used to provide the necessary platforms and Virtual Machines to implement the storage and processing needs. Software as a Service (SaaS) services shall be used to implement the service-based environment for performing the business interoperability. A base using SaaS and MDA [127] is proposed to be used for the modelling of the business evolutions, using the RESERVOIR OpenNebula IaaS, PaaS and SaaS platforms [128], and the overall NEGOSEIO architecture is being developed according to the concepts developed by the Open Group in TOGAF [129].

A proof-of-concept for NEGOSEIO was conceived and developed (see section 4.5.1), which showed that the negotiation environment requires a growing number of resources. Numerous agents, negotiation services and collaboration services consume a growing processing capability; the reference ontologies, knowledge and rules also need growing storage capabilities. This was the main justification for the decision of providing a cloud-based solution, which allows a progressive growth on the processing and storage capabilities, according to the business needs, besides providing several other benefits, e.g., maintenance, “green” computing, disaster recovery, and backups [67].

4.3.1.Multi-Agents System

The negotiation IAM states (see section 3.1.2) change accordingly to the defined logic. To implement this logic, the author proposes an agent-based environment [130] where agents are programmed to be responsible in the monitoring and management of all changes in the environment that may lead to a state change.

The framework that is proposed includes an agent-based system developed in the Java Agents DEveloping Framework (JADE) whose task is to monitor the changes on the negotiation states and on each participant's contribution to the interoperable environment (developed MDI model). It includes different types of intelligent autonomous agents [17]:

- **Monitor Agent**, which detects changes in the enterprise's model at its various levels which have impact in the interoperable environment, and changes in the interoperable environment itself, and when these are detected triggers the Negotiation Agent;
- **Negotiation Manager Agent**, communicates with the human operator, permitting the definition of new proposals, acceptance/rejection of submitted proposals, and invite of new participants to the negotiation; it communicates the outcomes of the Negotiation Coordination Agent, i.e., suggests decisions about the received proposals and suggests counter-proposals. A trigger may be defined regarding suggestion confidence upon which the Negotiation Manager Agent has authority to decide without requesting consent to the human operator;
- **Negotiation Coordination Agent**, which performs the negotiation activities with its similar on the other players. It implements the negotiation strategies defined and conducts the negotiations according to the defined rules, and according to the data in the received proposals and gathered knowledge, suggests a decision to the Negotiation Manager Agent or suggests a counter-proposal to the Negotiation Manager Agent;
- **Communicator Agent**, which handles the communication between the agents in the system. Every time a change in the negotiation occurs, it stores the occurrence in the data model (see section 3.1) and broadcasts this to all other agents using JADE's web services;
- **Persisting Agent**, which handles the keep-alive actions regarding the other agents, i.e., monitors if the other agents are alive and if not, proceeds to restart it; this agent uses the Agent Management System mechanism provided by JADE.

4.3.2.Semantic Ontologies

The proposed NEGOSEIO approach towards the consolidation of common understanding includes the creation of a reference ontology using the Web Ontology Language (OWL [131]).

Ontologies are built for the interacting players to contemplate the same semantics and understanding of the negotiation concepts and terms, supporting each business negotiated area. More than simply acting as taxonomies or dictionaries, these ontologies define the relationships between concepts and determine knowledge and methods from these relationships.

A reference ontology supports the semantic interoperability of the framework, needed for the complete seamless understanding of the underlying business [132]. On top of it, a business high-level set of services performs the interfacing with the users and implements the business interoperation that is required by the business model (Figure 4.7).

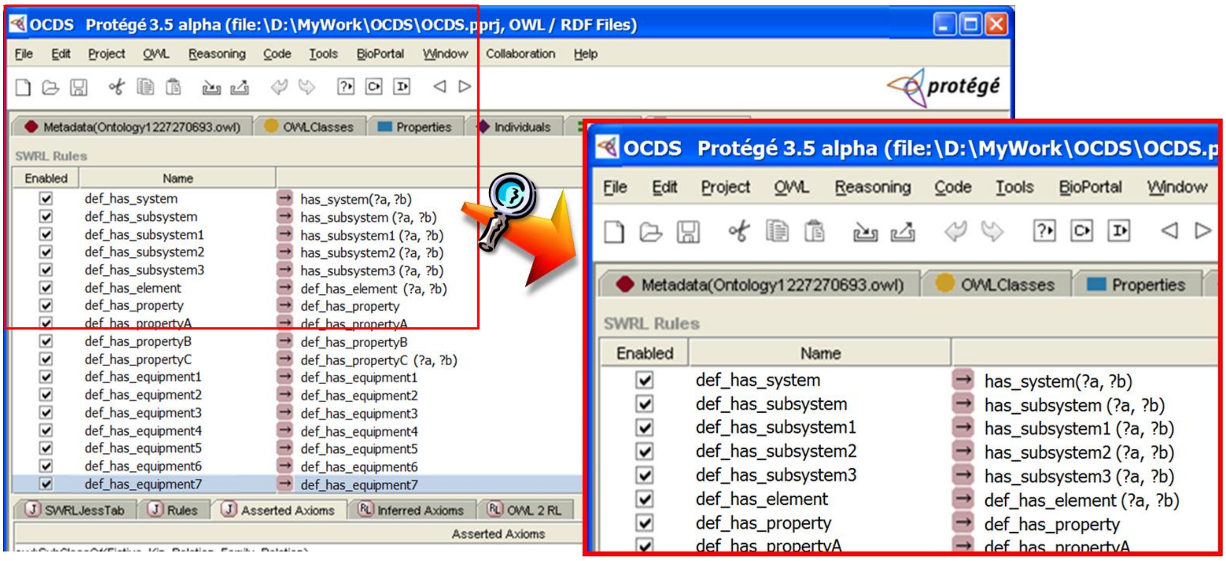


Figure 4.7: Development of a reference ontology using OWL in Protégé

Besides developing a reference ontology it is necessary to manage it adding new knowledge and relationships. This framework proposes the use of the MENTOR [52] methodology. MENTOR will allow heterogeneous business concepts to be harmonised into a common understanding in the business network, thus enhancing interoperability (Figure 4.8). Rather than being a static ontology, this reference ontology shall be adaptable to be compliant with new definitions, concepts and relationships along time, bringing updated knowledge for the users and for the management of the framework.

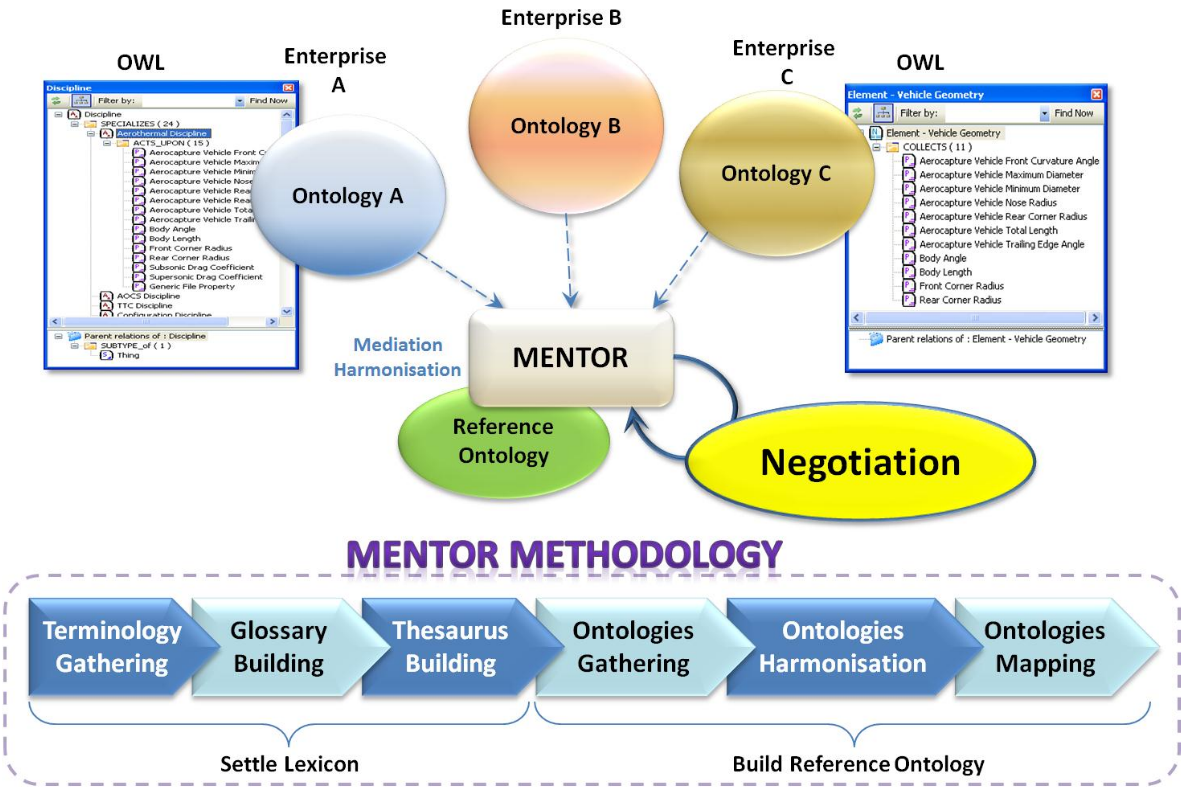


Figure 4.8: The MENTOR methodology + Negotiations

The MENTOR methodology provides a proper way to capture new knowledge (provided by new partners to the interoperable ecosystem) by combining and reasoning the different knowledge bases and formalisms targeted to interoperability. It uses the following activities to gather and enrich the knowledge from various sources and include it on the main reference ontology:

- **Mapping and matching**: for each entity (e.g. attribute, definition, concept) in the new ontology, a corresponding is created in the reference ontology, with the same or the closest meaning;
- **Alignment**: Making the subject ontologies consistent and coherent;
- **Translation**: Changing the ontology's representation (language) while keeping its semantics unaffected;
- **Transformation**: Modifying the structure and/or the properties of an ontology while keeping its semantics unaffected;
- **Checking**: A constant assessment of the ontology information consistency;
- **Evolution/versioning**: Configuration management regarding ontology domain changes and adaptations over time.

4.3.3.Parametric Models

The reasoning behind parametric models is to understand that the behaviour of a system is influenced by an infinite number of factors, e.g., a space mission behaviour and success is not only the responsibility of the conducting astronaut, it is the result of a very large team of experts that analyse and monitor every aspect of the mission, but everything, from solar winds to space dust, gravity, instruments, to even “the butterfly effect” constitutes an influence to the mission.

The target objective of parametric models is to determine the influence of each of these parameters to the behaviour of a mission by studying the historical data of past missions and the variance of each of the parameters [20].

The first step is the determination of the parameters that influence the resulting business. According to Pareto’s principle, 20% of these parameters influence 80% of the behaviour of the business, so the first idea is to determine a statistic analysis of the parameters that influence mostly the business outcome, possibly by using a Ishikawa (cause-and-effect) diagram, analysis of data anomalies, normalisation and calibration of parameters’ data [133].

Then the idea is to perform a regression analysis. Regression may be:

- **Linear**, if it can be said that the business is represented by an linear equation, where each value of y depends only of a single variable x such that it can be defined as an equation of the type:

  $$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \cdots + \beta_n \cdot x^n + \varepsilon$$

  where the value of y is a linear combination of a set of constant factors β_n

- **Non-linear**, if each y depends of more than one variable. In those cases, several algorithms can be applied, depending on the non-linearity which can be e.g., exponential, logarithmic, power, Gaussian. The most usual algorithms in this case are the Ordinary and weighted least squares, and also the Linearization of the non-linear variables.

The objective of the linear/non-linear regression algorithms is that a model can be built, based on the historical data, that is able to express with the least error a formula which models the previous behaviour of the business. The objective hence is to use this model to stimulate it with a new situation and try to determine, given a defined set of parameter conditions, which might be the future behaviour of the system.

These algorithms and systems have major appliances in weather and geological models to predict new situations based on the past experience, as can be seen on Figure 4.9:

Historical business database

Tester	Critical Factors for the Business Distance (Km)	Climate	Road	Car	Experience	Traffic	Duration (mins)
1	5	2	-1	0	0	-1	4
2	7	1	1	1	-1	1	9
3	15	2	2	1	1	1	34
4	18	0	1	-1	1	2	14
5	20	0	-1	1	-1	1	15
6	30	1	0	-1	0	-1	20
7	50	2	1	1	0	0	69
8	55	2	-1	0	1	0	49
9	60						68
10	65						42
11	66						33
12	70						67
13	70						128

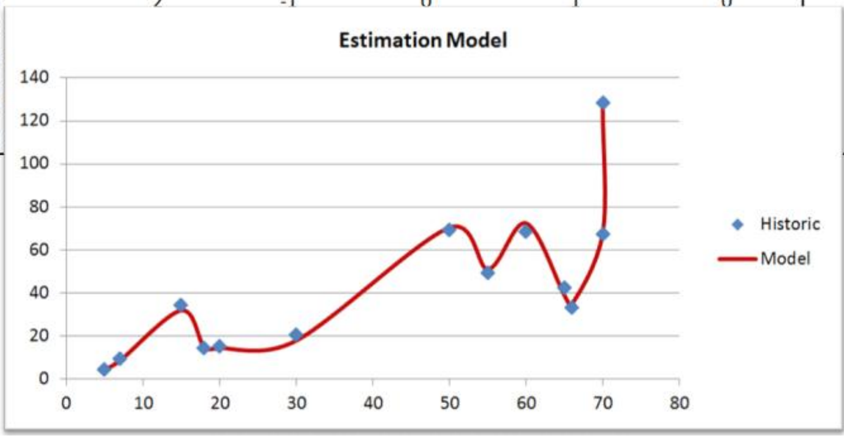


Figure 4.9: Parametric Estimation Model vs Historic data

The application of parametric models is important in the determination of the impact factor of each parameter of the negotiation, as this is important to determine the decisions that the MaS will take accordingly.

4.4. Application Scenario in a Real Business Case

The heterogeneity and interoperability situation described in the previous sections is particularly true in the aerospace segment – a competitive and demanding business supported by numerous applications, which vary from general-purposed, specific, proprietary or open-sourced, all needing to be interoperable and where the interaction between the various business elements should cause the smallest impact on the business itself.

The aerospace industry has a segment which is by nature driven by research, where new missions, challenges, strategies, and possibilities compete to be chosen for future development.

This development frequently conveys substantial resource and monetary investments; therefore it is crucial that each candidate mission is properly analysed and designed to prove its viability, opportunities, benefits and impacts.

The European Space Agency – Concurrent Design Facility (ESA-CDF) performs design studies to determine the feasibility of space missions [134]. Each mission has a defined vision and scope and is bounded by several characteristics, e.g., size, budget, lifespan, criticality, payload, the need for life-support, whether it is human-based or not, and many other.

A study encompasses a primary architectural and definition phase, where the mission is decomposed into several business-specific views which cover the different engineering aspects of the mission, e.g., power, propulsion, thermal, structure, mission analysis, avionics, known as design disciplines or engineering domains. Each domain performs develops its model view of the mission in full detail. The mission study is completed by harmonising all the engineering views towards a common vision – the mission, and the determination of its viability.

To accomplish the design of the mission domain models, the engineering experts perform several calculations and simulations, which culminate in the definition of a set of mission parameters, e.g., dry mass, structural dimensions, solar panel size, some of which are specific to be defined and used by a particular domain, and others that influence other domains, e.g., the power consumption of the instruments that are needed for the mission influences the number of batteries that are needed or the size of the solar panels, which in turn influence the structure of the spacecraft and its dimensions, which influences the total dry mass.

To develop the study, according to the above example, several scenarios may be developed considering the utilisation of the instruments to determine the needed peak power or a spacecraft's power autonomy, which in turn leads to several scenarios regarding the number of batteries, their capacity, the underlying technology, leading to several ways to dispose these batteries and all other equipment to determine the alternative structures and dimensions of the overall spacecraft and the materials that it shall be composed of, which lead to several values of the total dry mass, as can be described in Figure 4.10.

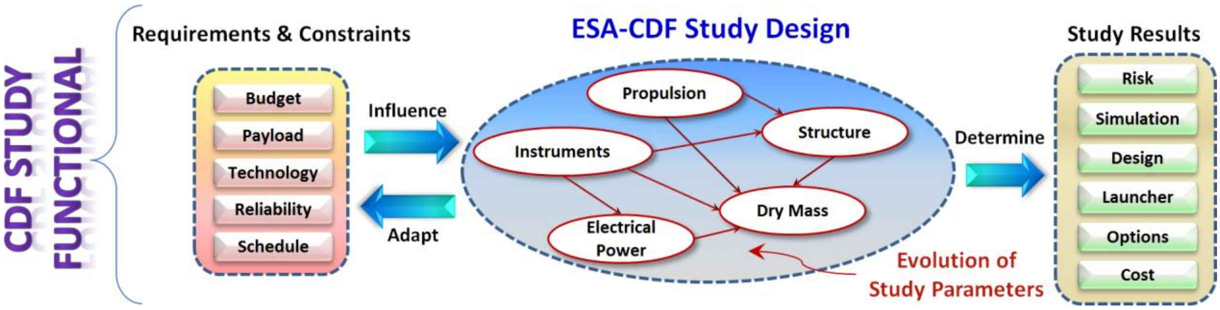


Figure 4.10: The ESA-CDF mission design functional diagram

Traditionally, the mission design used to be performed iteratively and sequentially by each domain. This meant each domain's decisions would only be disclosed at the end of its iteration, causing delays and late-adoption solutions. The need to improve the time spent performing space mission feasibility design studies has led the ESA-CDF to the adoption of Concurrent Engineering methods. Within this approach, the mission design is performed in parallel and developed in a set of design sessions carefully planned and scheduled, with durations which vary from few hours to a full working day where the whole team is present. A regular study has a medium of 6 to 10 sessions, each with the duration of half-day (one morning), twice a week. The resulting performance increased from the traditional six to nine months to just about three to six weeks.

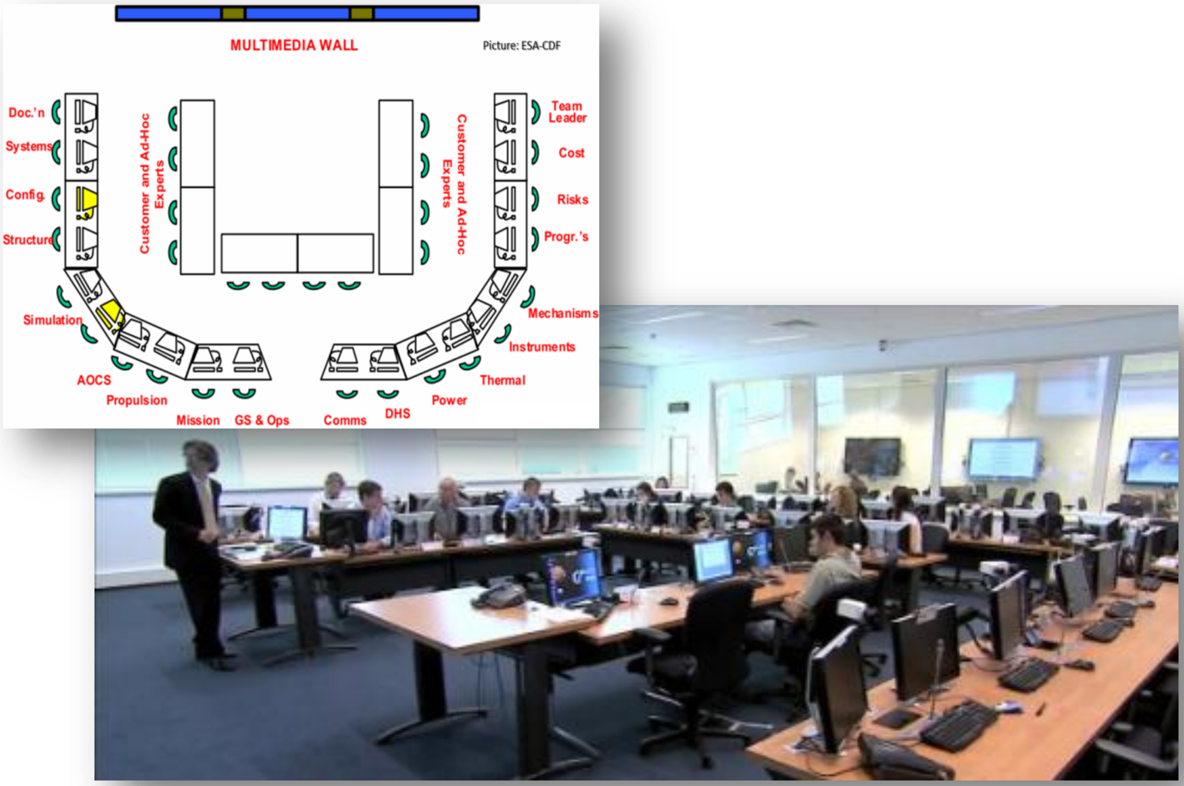


Figure 4.11: The ESA-CDF room layout (old & new)

Each mission has its own requirements, complexity and dimension, meaning that some may consider the involvement of more disciplines than others. Nevertheless, each design session involves a very expensive human-resource investment, by gathering in the same brainstorming arena – the CDF facility room [135], Figure 4.11 – dozens (normally 30 or more) of the best space engineering experts in the world towards reaching the important decisions that support the study results [136].

As can be seen from the figure, a great care was taken not to consider only the obvious technological aspects (e.g., Propulsion, Communications, Mechanisms, Simulation) but also other business-related aspects like Cost and Risk, all under the umbrella and coordination of Systems and of the Team Leader.

The "outer layer" of the room is occupied by the study domains; the centre seats are very important as they are occupied by the study customers and by invited experts in the mission area. So, all aspects of the business (customer included) are put together in a single room and all participate in the study, providing their feedback sequentially and in a moderated form.

It is thus essential that each session is conducted in order to maximise the contribution of all the involved engineers, which means interoperability in such competitive environment is priceless, between the engineers, the tools and the enterprises that are supporting their design.

The sessions are prepared, led and coordinated by a Team Leader and the Systems engineering area, which handle the interactions between the domain experts and are responsible for the budgeting, development and evolution of the main mission parameters, e.g., power, mass.

4.4.1.Problem Context

The research that conducted to this document is being developed over the real business case of the conceptual design of space missions' feasibility. This complex process, developed on ESA-CDF involves multiple domains which match the different views and interests of the mission. Each of these domains is a complex field where state-of-the-art design techniques combine with the expertise of the field engineers towards developing a model, responsible for providing a set of parameters that support mission decisions.

The difficulty about this process, besides the inherent complexity of the referred subjects, is the adoption of the concept of Concurrent Engineering, where the design is fast-tracked into a scenario where the multi-disciplinary teams (domains) perform their activities in parallel. Albeit each domain design team models its own view of the mission, the teams need to define and exchange a large set of mission parameters, required to satisfy the mission and to ensure that all views are fully integrated and fit perfectly.

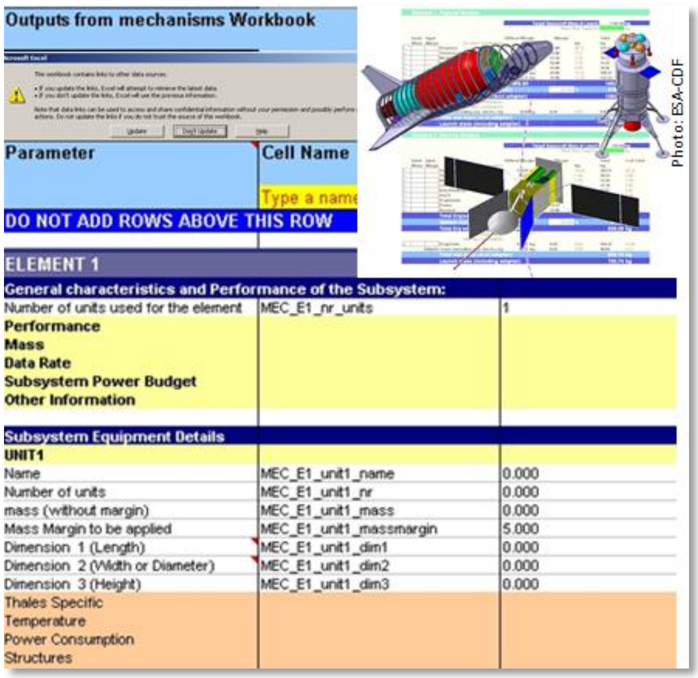


Figure 4.12: CDF domain spreadsheet

The design of each study domain is currently executed and reported in a domain specific spreadsheet (Figure 4.12), where the exchanged data which regards hundreds of mission parameters is carefully detailed in specific rows and cells. The data exchange between the domains is accomplished by linking all the domain spreadsheets in a moderated environment. Data dissemination and decisions are taken via the sharing of presentations which are planned, controlled, moderated and performed orderly by each domain (Figure 4.13).

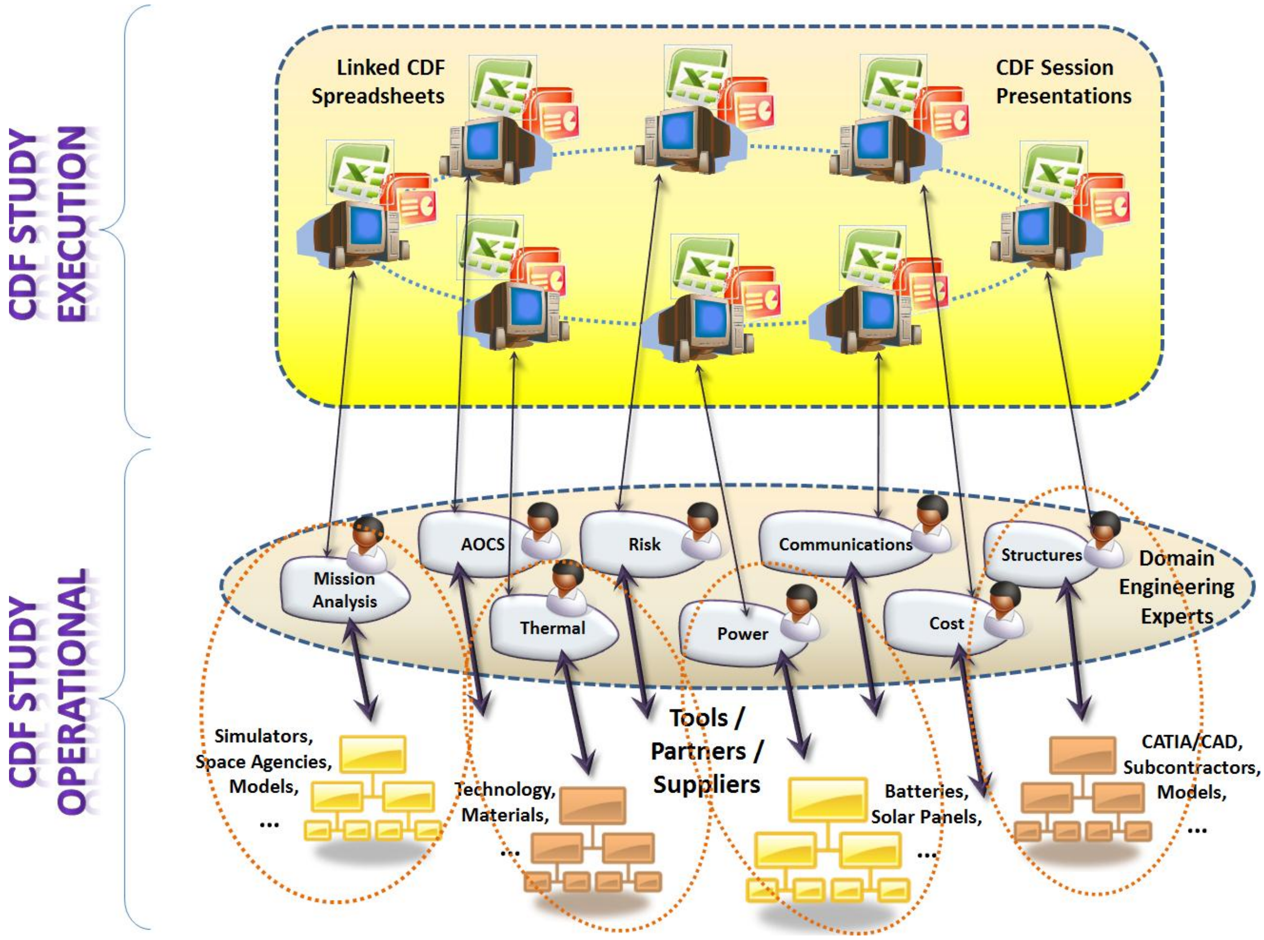


Figure 4.13: The ESA-CDF AS-IS scenario

Each domain engineering team in this AS-IS scenario performs its design using different tools (e.g., CATIA, STK, Matlab) and is provided and supported by a network of partners and suppliers – its Collaborative Environment or Virtual Organisation (VO).

Interoperability in this case is defined in two levels:

- The one between each domain and its tools, partners and suppliers in the supply chain towards the target of defining the domain design or vision of the mission;
- The one between the various domains of a mission-related study, where all the domains present their view and compete for their interests into setting the values for mission-related parameters.

With the heterogeneity related to the various systems and applications used by each design team and mission, problems of misunderstandings regarding the exchanged data and its dependences are frequent, leading to rework and interoperability problems. As studies get larger and more complex and the studied matters also leap the common understanding, interoperability weakens, and the involved parties in a study have to provide solutions to maintain and restore it.

The interoperability in this scenario is a responsibility of the study Team Leader which moderates the discussions, and of the Systems Engineer and related Assistants which provide local support to the domain engineers. The seniority of the domain experts, which are especially recruited to participate in the study, is actually a crucial factor to improve interoperability.

The interoperable environment is unstable as it depends on the synchronisation of the different methodologies used by each domain to perform the design, as well as the perfect synchronisation of the data in the linked spreadsheets. A mistake in an inserted value, a misunderstanding about a parameter's unit, a network communication failure, an error in a design macro or a spreadsheet cell name with a typo is enough to break the interoperability. This problem then has an impact which may vary from a simple communication problem between two domains, to a snow-ball effect (due to the fact that the CDF studies work in progressively detailed spiral iterations of exchanges between domains – as depicted in Figure 4.14) that may lead to the rework of the entire study.

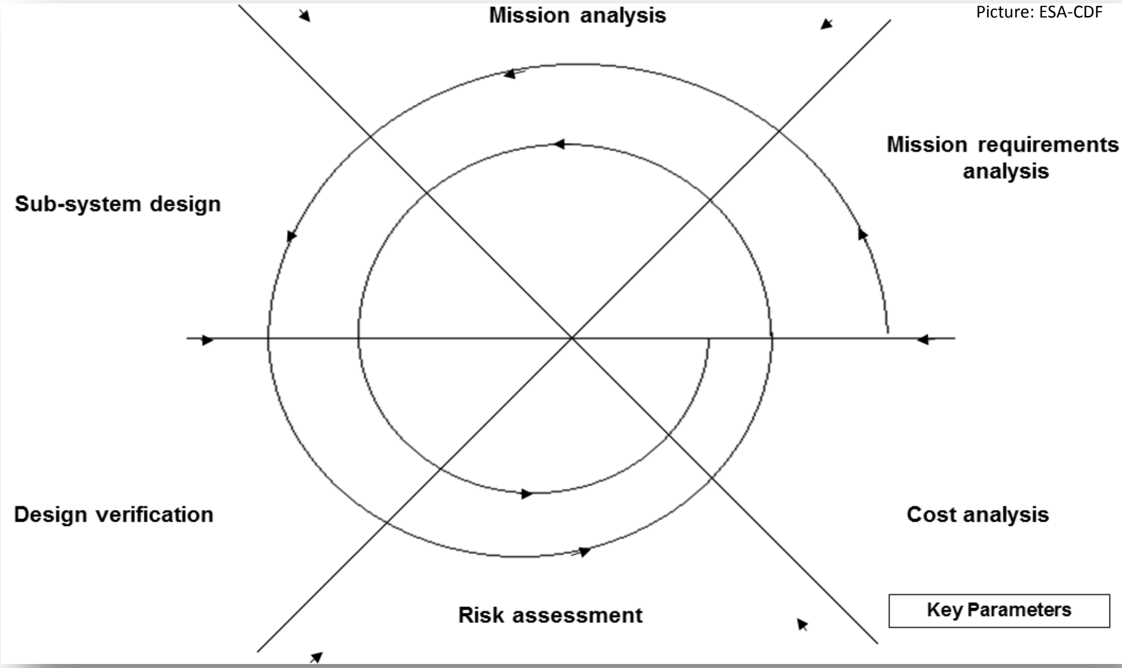


Figure 4.14: The ESA-CDF spiral model for iterative assessment

The current research developed a survey, with the collaboration of ESA-CDF, regarding the domain engineers' opinion regarding the main reasons that may lead to interoperability problems on the ESA-CDF environment, with the questions that were considered most suitable by ESA-CDF to capture their interoperability problems and needs [137].

The available results of this questionnaire stated that each domain interoperates with the others and with its supply chain (set of suppliers/partners/subcontractors/VO) using a heterogeneous set of interfaces which include shared data and tool interfaces, shared workspaces (e.g., shared databases) and shared tools. Heterogeneity in this environment grows with the proliferation of new trends, concepts, platforms, formats, technologies and development methodologies.

In addition, the survey outcomes also stated that the main difficulties regarding interoperability, not only with their supply chain but also with the other ESA-CDF domains, is related to missing deeper common knowledge of the business concepts, due to lack of available information. This reveals a problem regarding knowledge formalisation, reaching consensus and integrating design domains.

The challenge to interoperability is to comply with all already existing heterogeneous formats and interfaces and embracing new ones. If data formats and lower-level integration problems are becoming a lesser problem, as they can be solved/minimised via the use of integration tools (adapters and translators) or with the support of the Systems engineer and corresponding assistants, when it comes to semantics, terminologies, data comprehension, data dependency and methodologies, compliance cannot be solved as easily and this lack of knowledge is the most common problem in interoperability, increasing the risk of misunderstanding when integrating the business parties.

The findings also reported that the tools that are used to perform the design activities (e.g., Domain-specific tools, STK, CATIA) are standard and not customised to the specific design business needs, which means their concepts and terminology may not be aligned with the business. Standard tools are usually cheaper, more powerful and tend to be updated and improved more frequently than customised ones. Additionally, these tools have no direct interfaces towards the study domain design, which leads to the need of performing manual interactions and live data interpretation.

Despite its high-performance target, CDF environment manages to incorporate several knowledge management tools for capture of design study knowledge like decisions and lessons-learned recording. However the knowledge management on the scope of each domain is limited. The reuse of each domain’s tools and knowledge of these tools itself is scarce as each study carries its own context, dependability, together with limited available time on the domain engineers’ side. New studies regularly still lead to new design that could instead be performed reusing previous tools and knowledge if this was properly addressed.

Summarising, the ESA-CDF developed a high-performance process of designing space missions by fast-tracking the modelling of a mission into a set of different views that, when correctly integrated, should cover all the different stakeholder needs. This optimisation has the drawback of increasing risk and weakening interoperability. Conclusions of the analysis of the research survey state that the domains’ knowledge is very centred on the domain engineers expertise (core business), which deal with a large and complex set of business and data concepts, each with its meaning and dependencies. When performing the domain design, engineers need to interoperate with tools with static interfaces that do not match their business specifications and with a set of external providers/partners. When performing the overall design, the domain needs to collaborate with the other domains towards a common goal. The fragile interoperability existing in this environment is permanently endangered by any change introduced by any of these players or by the addition of new players.

The outcomes of this research are aligned with other studies and reports that were performed having the ESA-CDF as subject [138], where the analysis on the CDF IDM revealed difficulties regarding the interoperability between the engineering domains, with duplication of information, redundancy and problems regarding the sharing of data. The interaction between the domains was reported to be ensured essentially by complex and inefficient manual verification.

4.5. Validation Scenario

The NEGOSEIO framework is proposed to be applied to the ESA-CDF environment in a progressive manner. First, by submitting questionnaires that qualify the space environment in terms of requirements, needs and achievements, and major interoperability problems found [137].

Based on the ESA-CDF AS-IS scenario depicted in Figure 4.13, the application of the framework is modelling the study definitions and methodologies, concepts and practices, tools and infrastructure (high-level needs), and even the human interactions towards the study negotiations into a MDA CIM (and B-MDA B-CIM) that can define e.g., the concepts that rely behind concurrent engineering activities, the concepts and stakeholders of CDF study, the expected behaviour of the system as a whole, the organisation hierarchy, and major functionalities. This will capture methodologies, best-practices and operational knowledge, along with the needed functionality and dependencies.

Then, the B-CIM is transformed into a B-PIM that allows the CDF processes, structures and operations to be defined independently of the technology that will support them (e.g., the split of the mission into domains, what each will do, when will each interact towards the others, the definition of lessons-learned and capture of domain knowledge and practices, the relationship towards external parties, the configuration items). Nevertheless, each domain is defined together with its operational environment responsibilities and human behaviour.

Accordingly, the CIM is also converted into PIM, so that CDF functionalities, calculations and visions are defined and converted into algorithms, requirements, ontologies, data models and knowledge bases. This includes a top-down analysis performed to model the whole CDF operation.

This operation is already being performed: it has started in 2009, within the scope of project OCDS [22] which included the setting of a central data model defined by the technical memorandum ECSS-E-TM-10-23 [139] using the standard ISO10303 STEP and using EXPRESS statements, i.e., ESA-SEIM (Space Engineering Information Model [140]), a set of web-services interconnecting the various domains and related tools [118].

The data model repository holds the defined models as well as the information of the CDF study and each of its domains, including description of the interactions with each other. These models shape the set of collaboration services that handle the interaction between the domains, and from the domains to their supply chains.

An approach was also performed in the scope of the same project, to develop notations to model the CDF systems in a MDA. This modelling was performed using the Eclipse Graphical Modelling Framework (GMF), which is a part of the Eclipse Modelling Project (EMP).

The Eclipse Modelling Project focuses on the evolution and promotion of model-based development technologies within the Eclipse community by providing a unified set of modelling frameworks, tooling, and standard implementations.

This framework is based on the Eclipse Modelling Framework (EMF) which is Eclipse's attempt to implement the OMG standards on modelling notation. The EMF project is a modelling framework and code generation facility for building tools and other applications based on a structured data model.

From a model specification described in XMI, EMF provides tools and runtime support to produce a set of Java classes for the model, along with a set of adapter classes that enable viewing and command-based editing of the model [141].

On the other hand, the Eclipse GMF provides a generative component and runtime infrastructure for developing applications based on EMF.

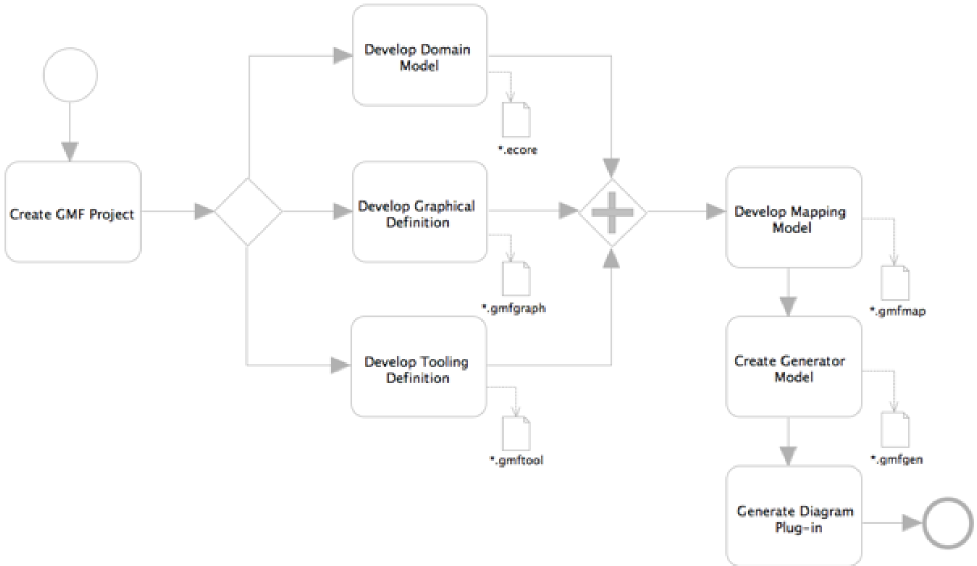


Figure 4.15: The Eclipse GMF development tool

Figure 4.15 illustrates the main components and models used during GMF-based development. Core to GMF is the concept of a graphical definition model. This model contains information related to the graphical elements that will appear in a runtime application, but have no direct connection to the domain models (here "domain models" refers to the generic business concepts' representation, using GMF specific modelling tools in this case. It is not related to the ESA-CDF Model) for which they will provide representation and editing.

It is expected that a tooling definition may work equally well for several domains. For example, the UML class diagram has many counterparts, all of which are strikingly similar in their basic appearance and structure. GMF allows the graphical definition to be reused for several domains. This is achieved by using a separate mapping model to link the tooling definitions to the selected domain model(s).

Once the appropriate mappings are defined, GMF provides a generator model to allow implementation details to be defined for the generation phase (Figure 4.16). The production of an editor plug-in based on the generator model will target a final model; that is, the diagram runtime (or "notation") model. The runtime will bridge the notation and domain model(s) when a user is working with a diagram, and also provides for the persistence and synchronization of both. An important aspect of this runtime is that it provides a services-based approach to EMF functionality and is able to be leveraged by non-generated applications.

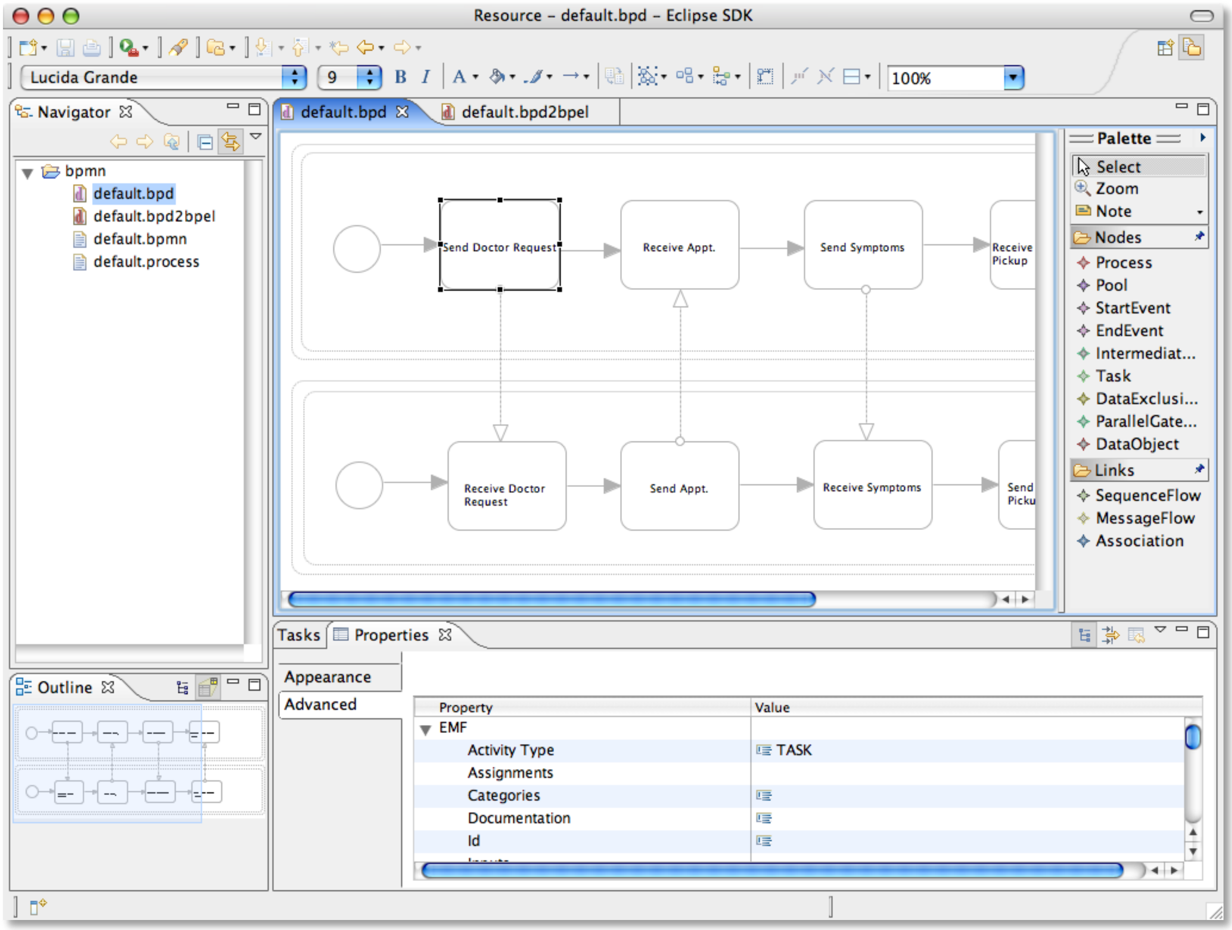


Figure 4.16: The Eclipse GMF model editor

Also, these models determined the creation of a space-related ontology, which was also defined in project OCDS, that describes, relates, and models the common space mission definitions, i.e., ESA-SERDL (Space Engineering Reference Data Library [142]), which was defined using the standard information model ISO 12006-3.

This represents a very important achievement in the CDF business as one of the major problems that was focused was the convergence in terminology and relationship between concepts and terms. The SERDL reference ontology supports the entire CDF concept, and seamlessly each domain and partners' particularities, determining the harmonisation of concepts to achieve global interoperation (Figure 4.17).

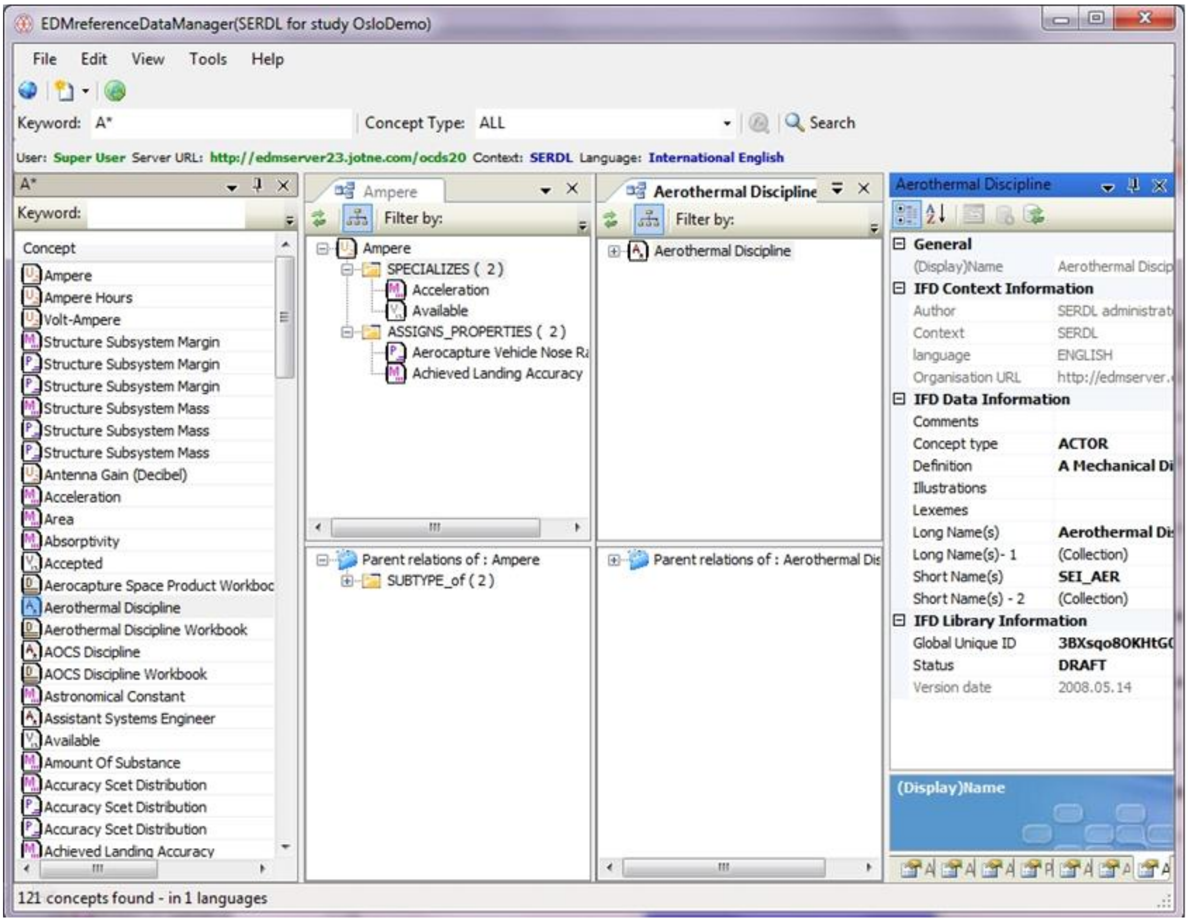


Photo: ESA-CDF - OCDS

Figure 4.17: The ESA-SERDL ontology editor

After this reference ontology is defined, the NEGOSEIO approach is to use the MENTOR methodology to enrich this reference ontology with the knowledge base of every new player that is added to the interoperable environment. Despite the techniques described in section 4.3.2, this action cannot be performed without moderation, for which NEGOSEIO proposes that its negotiation mechanisms are also used together with MENTOR (as suggested on Figure 4.8) to ensure that the reference ontology is not simply a disordered repository of knowledge but that its contents are meaningful.

Finally, transformations are performed until obtaining a PSM/TSM that implements the CDF interoperating environment embracing for example the session behaviours, including rules, manuals and wizards. The overall proposed TO-BE scenario for the ESA-CDF business case can be depicted on Figure 4.18.

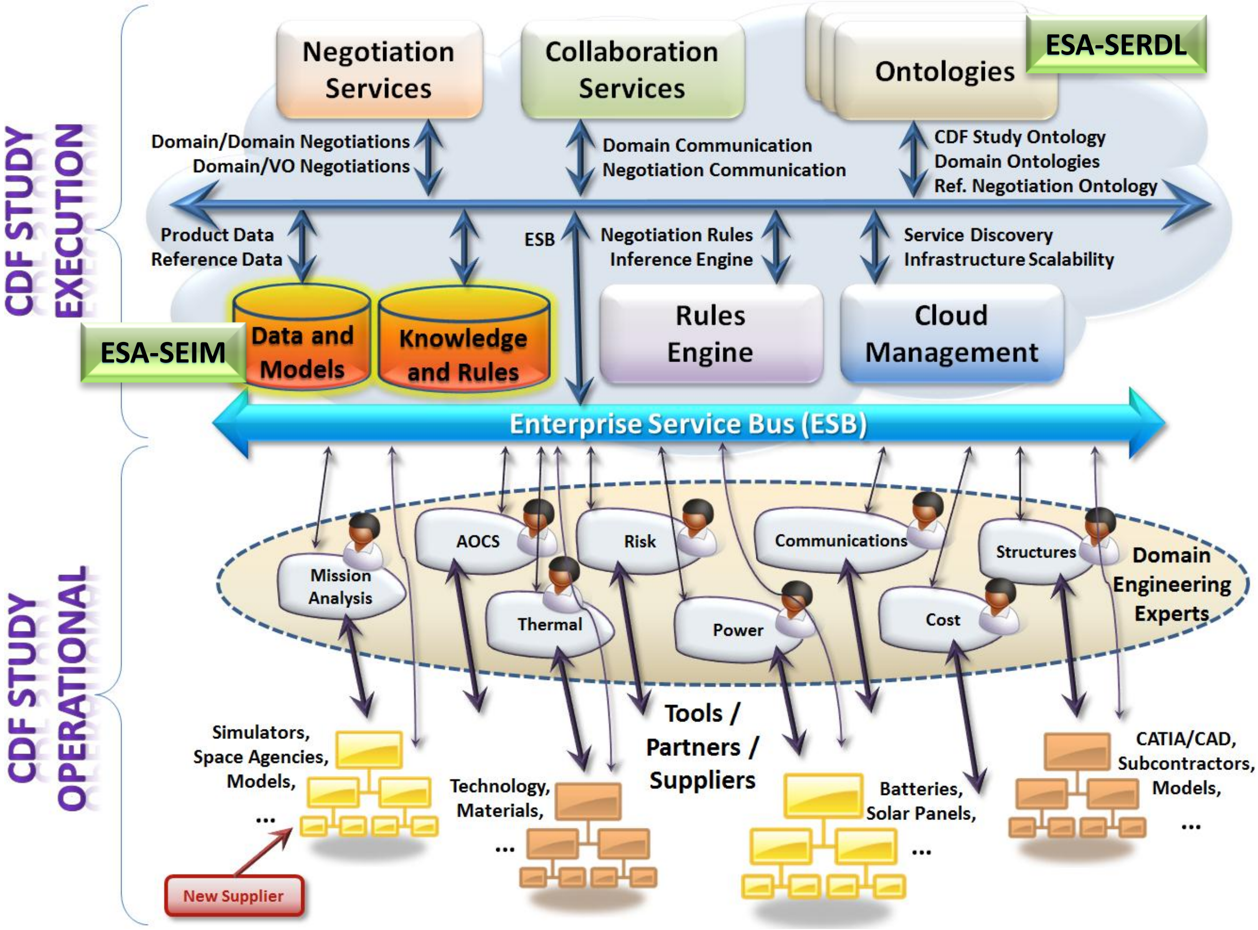


Figure 4.18: The proposed TO-BE ESA-CDF environment

In the TO-BE scenario implemented using the proposed framework, the engineering domains are interconnected by a set of cloud-based services that support the development and maintenance of the interoperable enterprise collaborative environment. These services are accessed by the engineering domains and their partners by means of a central Enterprise Service Bus.

Whenever a new supplier or partner enters or shows changes in the interoperable space (as shown in Figure 4.18), the intelligent agents developed in JADE trigger the negotiation services, powered by the Drools negotiation rules. These will then be used to inquire the parties for the motivations, strengths, benefits, threats, prejudices, opportunities and impacts (e.g., time, effort, cost, dependences) due to the changes.

Then, negotiations take place in order to reach the most suitable solution in terms of impact and stability of the networking peers, along its life cycle. Decisions derived from the negotiations include e.g., performing the changes as-is, performing adapted changes, achieving a consensus solution that best fits all parties, or decide that interoperation is not possible at all, not desirable or not worthwhile.

Regardless of the outcome of the negotiation, knowledgeable and mature decisions can be taken centred on accurate information about the other parties, based on facts and with decision support, able to be tracked and documented, hence providing lessons-learned and knowledge for future negotiations.

4.5.1. Validation Environment

Despite parts of the research are already being implemented on ESA-CDF like the SERDL, the SEIM and a backbone structure of web-services, the NEGOSEIO framework's negotiation part is still being analysed and discussed. While a full validation of the NEGOSEIO framework in the real ESA-CDF environment is still being evaluated due to the complexity of the CDF environment, a proof-of-concept has already been developed, by simulating the ESA-CDF worksheets and interactions in a nutshell environment (Figure 4.19) with sample domains, each equipped with an agent and the NEGOSEIO negotiation services implementing the negotiation levels defined in section 3.3.

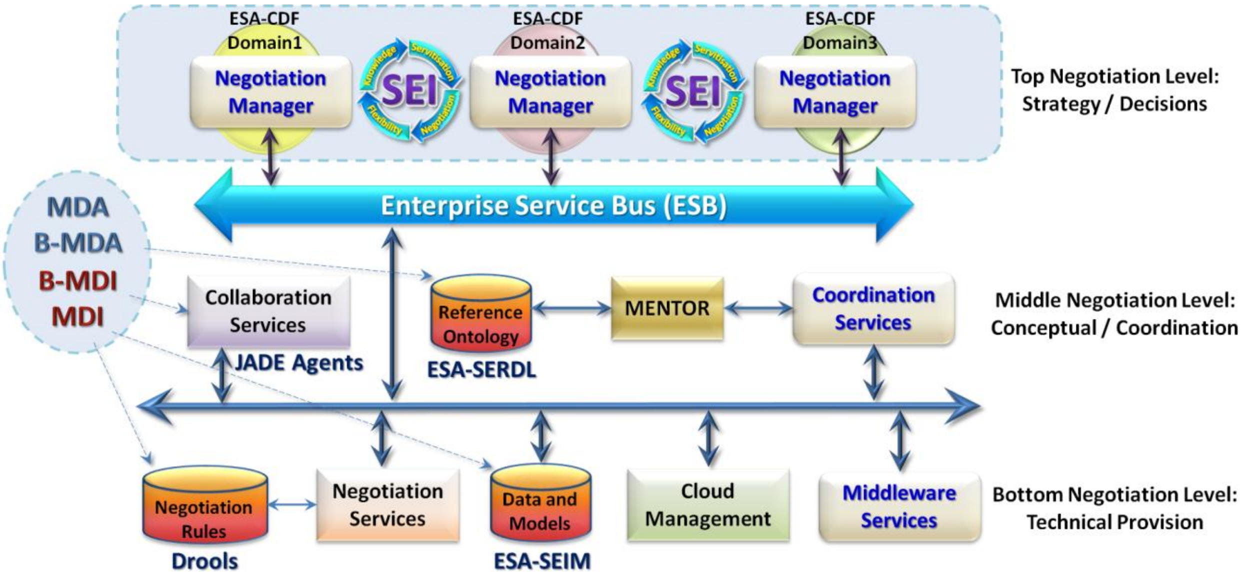


Figure 4.19: The NEGOSEIO proof-of-concept architecture

The main purpose of the proof-of-concept is to add the paradigm of negotiations to the quest for developing better solutions for interoperability. The scenario for validation is that each domain shall be assigned a software module, which shall incorporate the Negotiation Manager module (user interfacing with the framework) and a set of triggering intelligent agents.

The first steps of the methodology were taken by determining knowledge about the interacting domains. Hence, a series of questionnaires were submitted which regard the characteristics of the system and the requirements of the interoperability. These were stored in the rules knowledge base and qualified to the negotiation system as being considered positive or negative.

The proof-of-concept then was injected with interoperability changes with growing complexity, from simple interface problems to the most complex interoperability problems, and the time that it took to re-establish interoperability between the domains was measured.

4.5.2. Storyboard

When a participant wants to use the NEGOSEIO framework, it registers on the framework. Upon this registration, it receives a set of questionnaires and surveys to state if it supplies services and if so, the characteristics of these services (including the models and attributes like QoS and SLAs).

Upon the definition of MDA, and B-MDA, it subscribes to interoperate with other participants, whether by specifying clearly the unique identification of other participant or by determining via negotiations a set of potential participants (by inserting qualifying criteria).

When two or more registered participants decide to evolve to interoperability, they negotiate the interoperability terms, defining a MDI and B-MDI, and if the negotiations consider that the interoperable environment is feasible (Figure 4.2). In this negotiation, the strategies described in section 3.4 can be applied according to the participant's needs.

Then, an "interoperable node" is defined, which includes a central system controller module (4.5.6.1) and a set of cloud-based resources (infrastructures, platforms, software) is allocated to build the interoperable environment (including, e.g., a SOA, ontologies, and dedicated agents for negotiation). The supplied resources include the establishment of a Trigger Agent (section 4.5.6.2) on each participant for each interoperable node.

When one of the Trigger Agents detects a change in its interoperability environment (e.g., a change in the knowledge base rules), or traces pre-signs that the interoperability needs to be updated (e.g., a high number of data exchange errors, signs of rework in information processing), it triggers the negotiation processes which query the changing operator for the motivations for the change, and the specification of the SWOT for this change.

It then communicates with the other Trigger Agents present in the interoperability node, thus creating a negotiation proposal. Each of the Trigger Agents shall process this request, analysing its negotiation history and the set of changes that are proposed and its impact, and will produce a proposal.

This proposal is then presented to the Negotiation Managers of the other participants in the node, proposal which already includes (if possible) a suggestion on the decision to be taken, and a percentage of confidence in the suggested decision.

For the participants that accept the proposal, a new interoperability node is created with the new assumptions and resources. Likewise, the participants may reject the proposal or present counter-proposals to the other participants.

All negotiation interactions are stored in the central database according to the models defined on section 3.1.

4.5.3. Test Cases

This proof-of-concept was designed to determine a set of objectives:

- That the NEGOSEIO architecture is feasible, that with some generalisation it can be applied to the ESA-CDF business model;

- That the negotiation model is useful and that besides the static technological aspects, it captures knowledge about the business interactions, their influence in the future decisions and the decisions themselves;
- The establishment of new interoperability between new participants;
- The restore of the interoperability ecosystem after it has been lost due to some change;
- The sustainability of the interoperable environment.

A large set of test cases was then defined, which covered the definition of indicators on section 2.4.2. However, some of these are only applicable to a complete working environment and therefore are not applicable to the proof-of-concept, thus only the most relevant will be enumerated here:

TC#1: After already having an interoperable environment between two domains, simulate the induction of a change that makes that interoperability not possible anymore; determine the steps to restore it, and the overall time to complete it without NEGOSEIO (Indicator #9).

TC#2: After already having an interoperable environment between two domains, simulate the induction of a change that makes that interoperability not possible anymore; determine the steps to restore it, and the overall time to complete it with NEGOSEIO, negotiating the proposals for achieving interoperability and determine and measure the negotiation time and the overall time (Indicator #9).

TC#3: In an interoperable environment, submit changes with growing complexity and determine the time that it takes to recover the interoperability (Indicator #6).

TC#4: In an interoperable environment, submit the same change with a growing number of participants and determine the time that it takes to recover the interoperability (Indicator #6).

4.5.4.Challenges

The challenges proposed to this proof-of-concept were to mimic the CDF environment in a conceptual but clear way so that the stakeholders would identify their business case as applicable to their environment.

This was a difficult challenge as the author needed to determine an environment that would be abstract and general enough to maintain the coherence of a Ph.D. research and detailed enough so that the software engineers in ESA-CDF would recognise it into their models.

Another challenge was to determine if the negotiation environment indeed contributes to an increase of the sustainability of the participating domains' interoperability, thus responding to the research question.

4.5.5.Negotiation Rules

Rules in the NEGOSEIO framework work in several levels and are used in scenarios with multiple purposes:

- To perform the negotiation steps (negotiation interactions);
- To capture the interoperability needs (factors → decisions);
- To store the negotiation model evolution (see section 3.1);
- To store the negotiation objects and decisions, which include the basic (negotiation over the best solutions for interoperability, but may also be used for auxiliary negotiations:
  - Determination of a proper solution or family of solutions when facing a set of problem characteristics;
  - Determination of the best service or service orchestration;
  - Determination of the cloud resource needs according to the demand;
  - Determination of methods for transformations in the abstract enterprise models.

The proof-of-concept aimed to demonstrate the application of the Drools platform for defining the business and negotiation rules. As an example, in the defined proof-of-concept the negotiation knowledge is acquired in the decisions made by the trigger agents. Whenever a change is detected by the Monitor agent associated with the enterprise (section 4.3.1), it triggers a negotiation instance. At that moment, Figure 4.20 illustrates an example of a questionnaire made by the initiator of a negotiation.

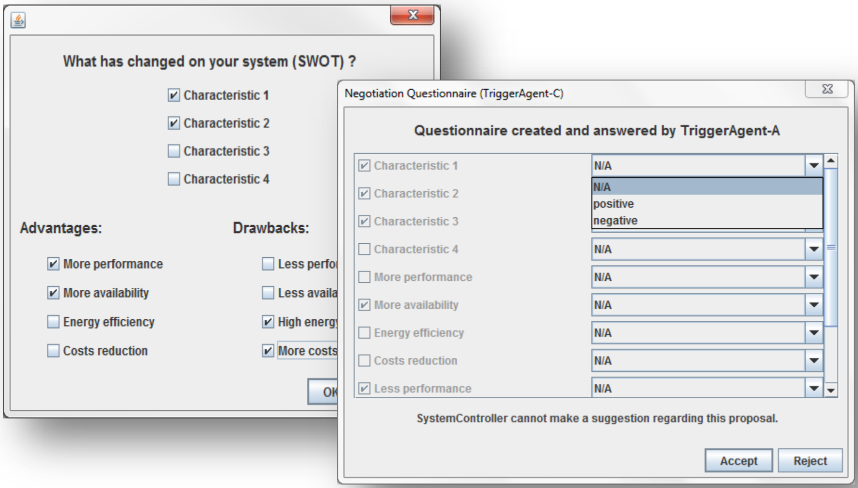


Figure 4.20: Questionnaire made by negotiation initiator

Similar questionnaires are posed to all negotiation participants and all the answers made by the participants are saved in the knowledge base, within the form of a knowledge rule. The rules are defined and particular for each participant, as to separate the motivations and decisions of each involved player in the negotiation.

```
rule "Rule-Example-Reject"
    when
        $map : java.util.HashMap(
            this["Energy efficiency"] == "false",
            this["High energy consumption"] == "false",
            this["Costs reduction"] == "false",
            this["More costs"] == "false",
            this["Characteristic 2"] == "false",
            this["Characteristic 1"] == "true",
            this["Characteristic 4"] == "false",
            this["Characteristic 3"] == "false",
            this["Less availability"] == "false",
            this["More availability"] == "false",
            this["Less performance"] == "false",
            this["More performance"] == "false",
            this["Agent"] == "TriggerAgent-2")
    then
        resBean.addReject();
end
```

Figure 4.21: Example of Drools rule for negotiation

Figure 4.21 shows a typical rule modelled for a decision, where it can be seen that when the negotiation initiator created a questionnaire that had the value “Characteristic 1” selected, the response of the player who has this rule defined and connected to this questionnaire was to reject it.

Of course future decisions shall not be affected by a single rule, because the knowledge base can contain more rules of this type with different results. Several decisions shall be taken and determined according to scenarios and to other motivations and characteristics.

The joint effect of the rule engine and MaS is then to ascertain its artificial intelligence qualities to determine separately the impact of each parameter into the final decision, much similar of what is performed using parametric estimations (as described on section 4.3.3), to propose a new decision to be taken on the proposal.

4.5.6.Modules

The proof-of-concept negotiation environment is composed by two main modules: the System Controller, which is a dedicated module for each interoperability node, and the Trigger Agent, which is defined and deployed on the Negotiation Manager component (section 3.3).

4.5.6.1. System Controller

The System Controller module is defined as a centralised process in the NEGOSEIO framework (so it doesn't weigh on the applications' environment) which handles the interoperability between a set of participants registered in NEGOSEIO.

This is called an interoperability node, where the joined participants share a virtual environment in the cloud, associated with a set of services which provide the necessary means for interoperability between the participants. These may include communication services, a dedicated reference ontology and other collaboration services.

Figure 4.22 shows the main console of the System Controller, where we can view four Trigger Agents connected in the same interoperability node.

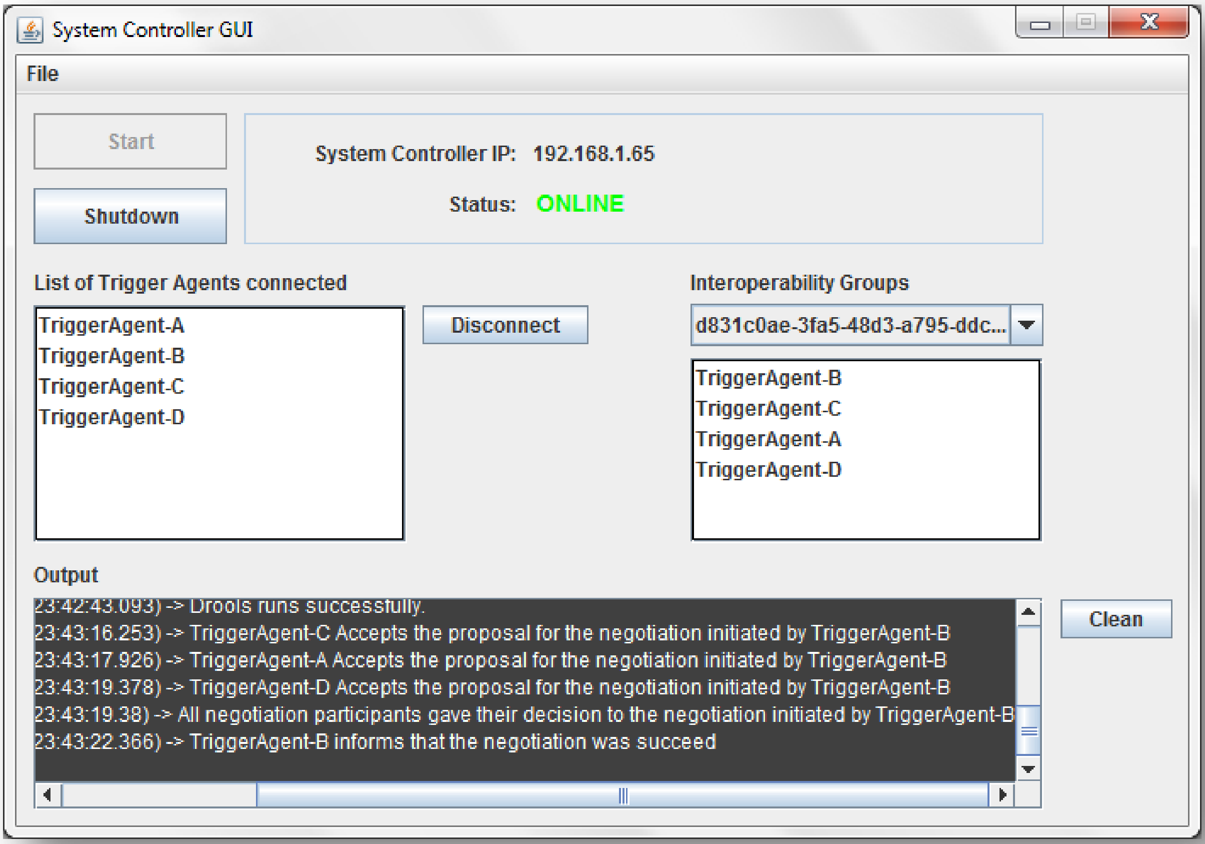


Figure 4.22: System Controller connected to Trigger Agents

All the interactions and logs are also saved in the database for future recall and consolidation in the subsequent negotiations.

4.5.6.2. Trigger Agent

This is software component deployed with the Negotiation Manager, which actually is composed by a set of intelligent agents (see section 4.3.1): a Monitor Agent to detect the changes, a Negotiation Manager Agent to handle the interface with the Negotiation Manager, a Negotiation Coordination Agent to implement the negotiation strategies, a Communicator Agent to deal with the inter-agent communications and an auxiliary Persisting Agent for maintaining the consistency of the environment.

In the developed prototype, instead of having an automated agent to detect changes, changes are simulated by pressing the corresponding button (Figure 4.23).

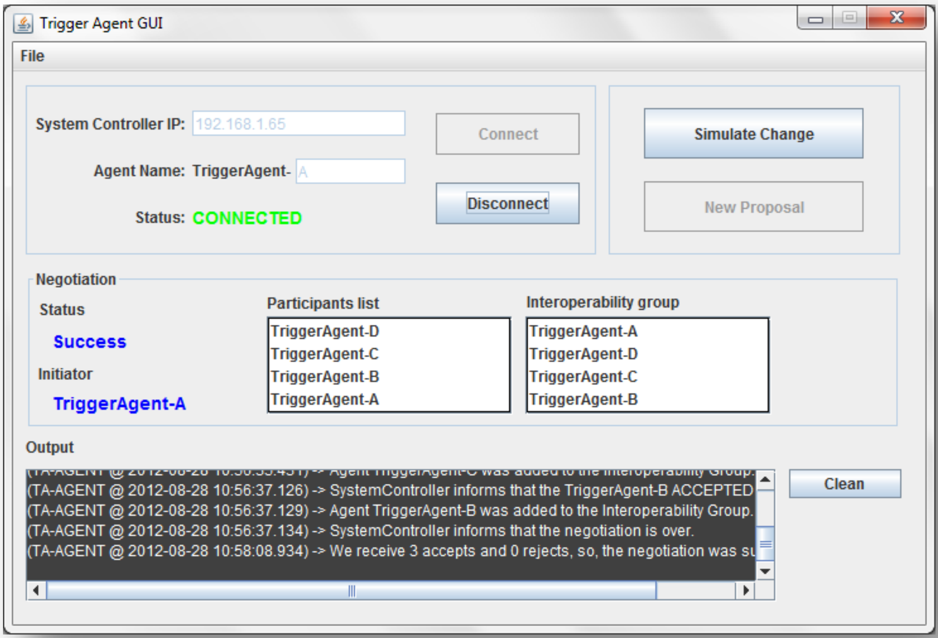


Figure 4.23: NEGOSEIO Trigger Agent

5. TESTING AND VALIDATION

> "
> *If you're searching out for something, don't try so hard*
> *If you're feeling kind of nothing, don't try so hard*
> *When your problems seem like mountains,*
> *You feel the need to find some answers,*
> *You can leave it for another day, don't try so hard.*
> "
>
> *Don't Try So Hard – Queen*

This section presents the testing activities which were performed to validate the negotiation aspects of NEGOSEIO, and the outcomes that resulted from these tests in terms of consolidated analysis and in terms of the reporting results that were validated by a panel of selected experts of the responsible personnel of ESA-CDF.

5.1. Functional Testing

The functional tests that were performed concerned the four test cases defined in section 4.5.3:

- Section 5.1.1 describes the validation of TC#3, submitting changes which evolve in terms of complexity, and measures the recovery time in a system without NEGOSEIO (TC#1), and in the system with NEGOSEIO (TC#2), it measures the negotiation time and the recovery time;
- Section 5.1.2 describes the validation of TC#4, submitting changes which evolve in terms of complexity, and measures the recovery time in a system without NEGOSEIO (TC#1), and in the system with NEGOSEIO (TC#2), it measures the negotiation time and the recovery time;

5.1.1. Interoperability problems with growing complexity

The functional testing approach to NEGOSEIO using the proposed validation scenario described on section 4.5, with sample CDF domains and defining a set of interoperability changes with growing complexity, and determining the resources needed to restore the interoperability in these cases, with and without negotiations. This test tends to demonstrate the path of the evolution in interoperability, as the evolution of the Future Internet foresees a coming future with semantically-rich relationships.

The proposed tests included injecting to the two established scenarios (prior and with NEGOSEIO) a set of interoperability changes which make interoperability not possible between the players, with growing complexity, and determining for each case the needed resources for restoring interoperability.

To simplify the testing, the impact of redeploying and restarting the applications was discarded, as this is necessary in all scenarios. Also, in terms of negotiations, the aspect of people not being available to perform the negotiations was also not considered. The measured time was then rounded to minutes for simplification, which is OK since most determinations regarded a particularly defined code, whereas the code density and complexity may change a lot. Factors like programmers' productivity rate and other were also not considered.

These numbers are particular for the tested code as it was considered that the aspect that matters is the relation between the measured values and not their absolute values.

This example shows an example generic web service that performs the weight calculation for objects. Notice that weigh is not the same as mass, which is one of the main parameters of the CDF studies. The web service server offers one method called "calculateWeight" that receives a battery for input and returns a value for its resulting weight.

```
public double calculateWeight(Battery bat)
```

Change of complexity #1: Configuration change

In the first change, a configuration parameter was changed stating that the gravity to be considered for the calculations would be the Moon's instead of Earth's.

<gravity> = 1.622 m/s^2

TC#1 – Current: This change required a simple change in the configurations file, and the restart of the services, therefore the overall needed time and resources can be neglected.

TC#2 – After NEGOSEIO: As the change needed to be formalised and accepted, the interaction took about 4 minutes.

Change of complexity #2: Server configuration change

This test considered the change of the web server port from 8080 to 8081.

TC#1 – Current: This change required a change in the WSDL file, and it was necessary to rebuild some of the software client classes, and restart the clients. Time spent: 30 seconds to 1 minute.

Note: Changing the technology to use C# instead of Java meant it was only necessary to change a port number in the "app.config" file hence the time was reduced to 20 seconds.

TC#2 – After NEGOSEIO: The change required a negotiation to establish if the best alternative would be to change port or to change server. Both solutions were proposed, and the decision was rapid, but it added an additional step to the change of 15 minutes. Therefore the overall time was 16 minutes.

Change of complexity #3: Simple interface change

```
public double calculateWeight(Battery bat, String unit)
```

This change requires the addition of a new parameter in the invocation of the service, which now provides different values according to the specified unit. This change was performed on the server side, to provide additional functionality and clarification about the method's behaviour.

TC#1 – Current: This change required a change in the WSDL file, and also a change on the code of all the calls of the web service. If the client code also performed the conversion of unit, this can be as well erased. In the performed tests, several results were found:

- Some client applications (20%) simply already had a central method called with the old interface and hence the solution took short time (12 minutes), as the code was barely affected; anyway, the new functionality is not really being used;
- Some client applications (60%) created a central method for calling the web service or simply performed a find-and-replace on all code to replace interfaces; this took a medium time of 4 minutes/call, and also doesn't benefit from the new functionality;
- Some (20%) performed a thorough change in the code and besides replacing the interface, also removed additional code used for local unit conversion while using the new functionality at its full potential. This meant an average of 12 minutes/call, as the change simply comprised removing an additional calculation;
- Considering and the percentage of adopters, the average total time spent considering 20 calls to the web-service was of 98 minutes.

TC#2 – After NEGOSEIO: The change required a negotiation to determine the best solution for each client. The various solutions above and others were proposed, which took some time, and the decision was not so rapid as there were several proposals to consider, adding a negotiation time of 45 minutes. It was however noticed that most of the participants (80%) found that the best solution would be to perform the thorough change in the code, which is the most expensive change but the one that takes the most advantages of the server change.

The overall time spent considering also 20 calls to the web-service + negotiation time was 249 minutes. Here can be seen that although the negotiation time is considerable – it is almost half of the average time without negotiations – it is only 20% of the total time spent for the change, also because it elicited a more mature change (and also the one who meant more effort).

Change of complexity #4: Multiple Semantics

```
public double calculateWeight(Battery bat, String unit)
```

The fourth change concerns semantics. It was detected that this web-service, which was designed to weigh Lithium-ion batteries, was also being used to weigh Nickel-cadmium batteries, which obviously do not have the same mass. Hence, there is a semantic mislead in the term “battery” that needs to be solved.

TC#1 – Current: It’s necessary to develop an ontology (at least a taxonomy!) in order to correctly define what a battery is, and the advantage of the ontology is that it may define that “battery” can be divided into “Li-ion battery” and “Ni-Cd battery” (and possibly others as well). Developing an ontology includes some steps, that are represented below:

- Create an ontology and store it;
- Defining classes in the ontology;
- Arranging the classes in a taxonomic (subclass-superclass) hierarchy;
- Defining and describing allowed values for this slots;
- Filling in the values of the slots for instances.

The average time to perform these activities, for a very basic ontology just considering these terms is at least 1 or 2 days. There is an additional time associated to changing the calls from Ni-Cd batteries to “weighNiCdBattery()”, but this is residual compared to the time to build the ontology.

TC#2 – After NEGOSEIO: A reference ontology already exists in NEGOSEIO. The task would simply be to add the two concepts as deriving from “battery” and clearing out what each means, which means negotiation activities which lasted an average of 215 minutes. Then there was the solution to change all calls to their new web-services, which means new negotiations and implementation, taking about 340 minutes.

Change of complexity #5: Semantics change

This interoperability change occurs when an entity wants to do a modification in the Battery ontology used in the previous example. For example, previously the ontology represents an AA battery, whereas now, the ontology represents an industrial-sized battery.

This change may take very long time to be matched, since the client needs to know exactly what the server is “talking”. It may mean a new parameter for size, or a new definition on characterisation, or other ways to describe the new item and differ it from the previous. Below are represented all steps that the client needs to do to continue interoperable with the server:

- Know exactly what has changed;
- Redefining classes in the ontology;

- Rearranging the classes in a taxonomic hierarchy;
- Redefining and re-describing the allowed values for these slots;
- Refilling in the values for slots for instance.

TC#1 – Current: The average time to perform these activities, as it concerns communication and understanding, as the participants are not used to perform these activities (at least at this level) for a very basic ontology change the average impact was 4 to 6 days. All additional activities, which still take an average of 6 hours, can be neglected.

TC#2 – After NEGOSEIO: The negotiation mechanism can be used at its full potential, whereas the triggering agents made an early detection the proposed change in the semantics and handled the negotiation, suggesting solutions that were successful in previous negotiations and estimating the success rates for each solution, thus achieving more rapid decisions. Negotiations took an average of 1.15 days. Nevertheless one still has to account the implementing activities, which after the negotiations take about 6 hours to conclude.

Change of complexity #6: Process change

The last change tested is the most complex, since is a change in the process of an operation. In this case is a change in the weighing process. The previous process of weighing batteries expected them to include the battery pack case. The current process expects batteries to be weighted without the heavy protection case to bring more realistic measurements. What seems to be a very simple change may reveal to have several other implications:

- Battery weighing was performed dropping the battery into a scale. Now without the protective case the batteries are much more fragile, hence a change must be done to avoid breaking the battery – possibly an electric arm could gently place the battery in the scale, but this will delay the process; other solutions can be found, all with their pros and cons;
- After the weighing, the battery used to be placed in its place before painting the structure. As this is performed at a high temperature, the battery without its case cannot resist the temperatures, so the painting must be performed before the battery is placed. However, by that time, not all elements were mounted yet;
- …

Hence it can be seen that a small process change is bound to have a deep impact in the whole process of building the spacecraft. This is as truer as the processes get more and more specialised and optimised, meaning all manufacturing activities are linked and dependent.

TC#1 – Current: The average time to perform these activities, as it concerns communication and understanding, scenario analysis, multiple parameters and variables, is very uncertain, but most process changes if unsupervised may have an impact of 15-20 days.

TC#2 – After NEGOSEIO: The negotiation mechanism has the formalisation of the changes that need to be performed and has a parametric analysis of the impact of each factor change, hence it can determine various scenarios from past experience or simulation to analyse the scenario with least effort/time/impact. For each of these actions, the previous experience determines the set of services which need to be enabled and the best solutions for the required needs. Despite being complex, the negotiations have a proper development environment to maximise performance and minimise the impacts. It can make use of the intelligent agents which don't have human limitations and may continue processing new possibilities and scenarios and assess its feasibility. Nevertheless the measured average negotiation time was in average of about 3-4 days to reach a solution which was validated by the various stakeholders. The changes to be performed by reusing services and strategies took an average length of 2 days.

The overall graph of the interoperability restore time against the increasing complexity of the problems can be depicted in Figure 5.1.

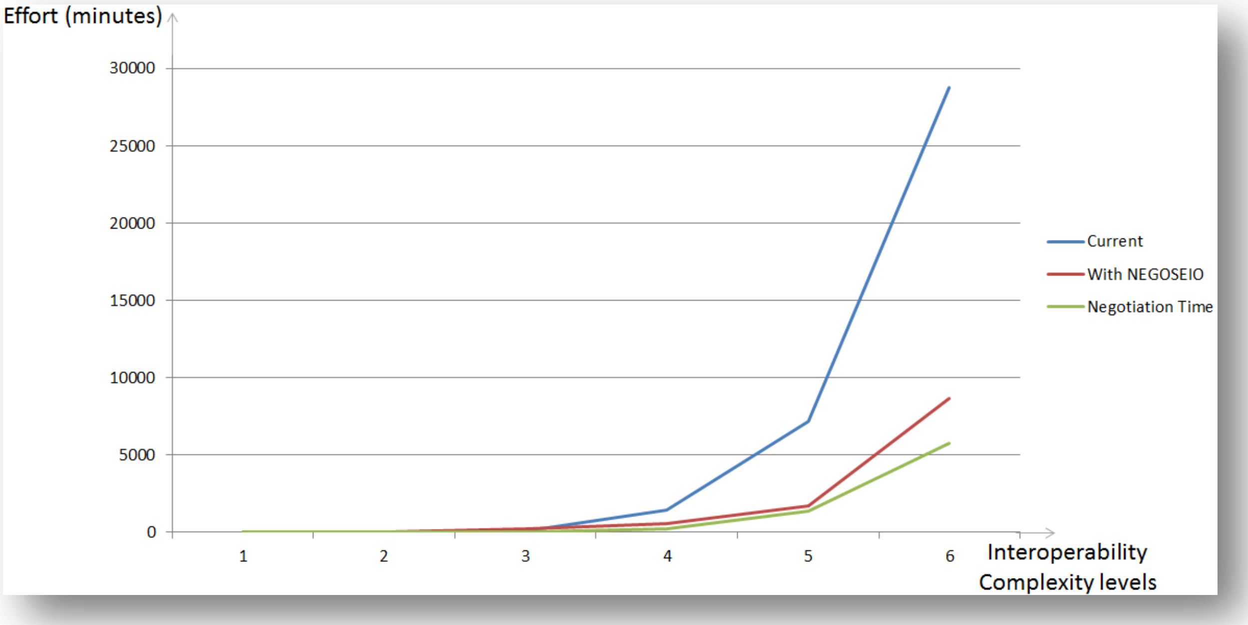


Figure 5.1: Metrics of Restore Time vs. Complexity

5.1.2.Interoperability problems with growing number of participants

This testing consisted in determining an interoperability problem – in this case, it was a semantics change – and measuring the time that it took to restore the interoperability as the number of participants to the interoperability scenario grew.

The number of communication channels is geometrically dependent of the number of participants in a negotiation, such that

$$\text{Channels} = \frac{N \cdot (N-1)}{2}$$

The results were that in the current systems, the increase due to the addition of a new participant in the consolidation of the meaning of a particular term increases with the number of channels and negotiations that need to occur, in a similarly geometrically way, while the NEGOSEIO approach shows a large amount of time to establish negotiations, but this number keeps stable either because the negotiations are performed by the agents in parallel, and also because due to the stability in time the rules engines give answers with greater confidence levels. The very implementation had little change according to the increase of participants, as the solutions are already determined by the negotiation, as can be seen on Figure 5.2.

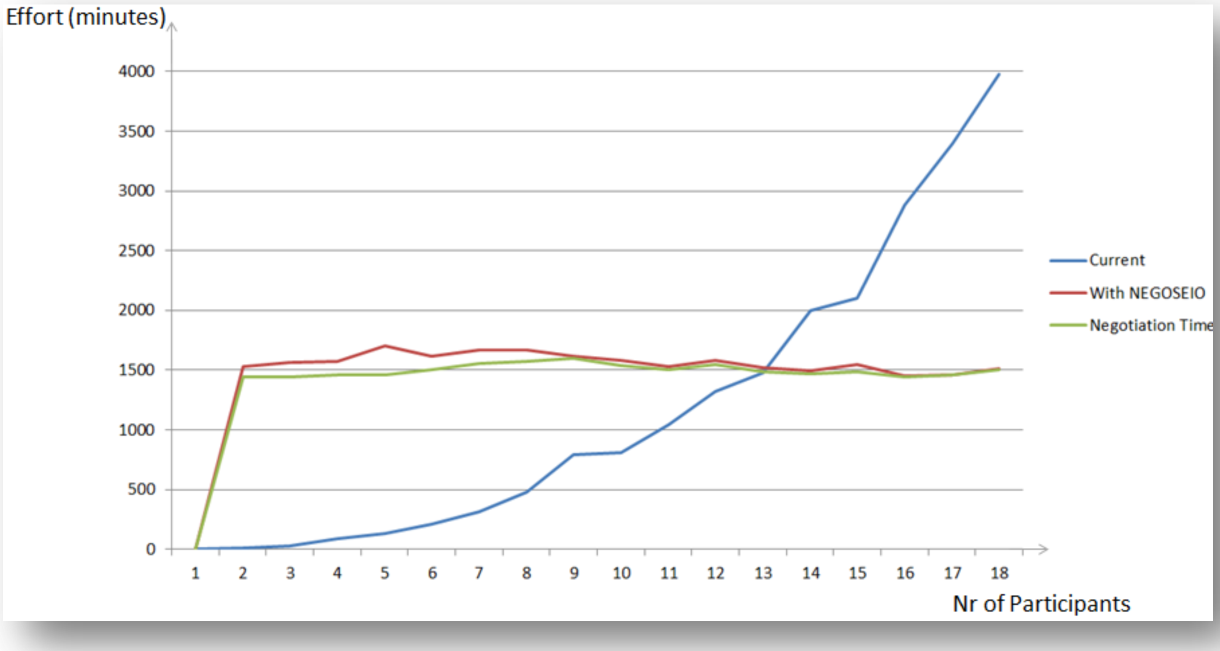


Figure 5.2: Metrics of Restore Time vs. Nr of Participants

5.2. Test Results and Conclusions

The results that can be obtained from the testing is that overall interoperability complexity grows linearly with time, following the market needs for tighter interconnections and more context-aware dependencies. What can be derived from the tests performed is that complexity with NEGOSEIO also grows with time; however, after some time this growth is softened, mainly due to the fact that several formal mechanisms are implemented, together with the Multi-agent System which permits an early detection of changes in the models, and the fact that the negotiation rules grow the body of knowledge in terms of permitting an easier selection and reuse of tools, services and decisions, as can be seen on Figure 5.3.

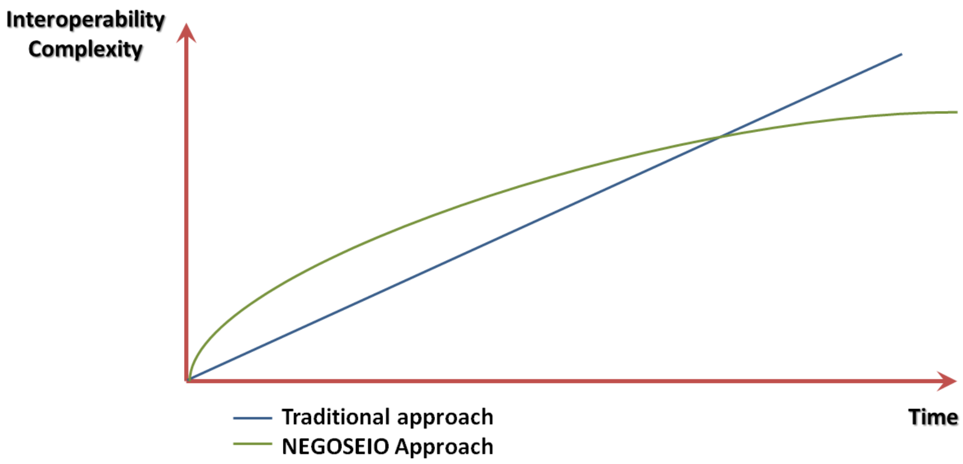


Figure 5.3: Interoperability Complexity in Time

Additionally, Figure 5.1 does not show in full detail the relation between the time of reestablishment of interoperability (also called "Downtime") and the growth of the interoperability complexity. This is because the defined steps have a very large impact when complexity rises and the figure does not properly show the behaviour in the first steps of interoperability complexity.

The observed results were that while the traditional approaches are very fast to re-establish when the complexity of the interoperability was simple (interface changes, syntax changes), the downtime increases geometrically when the complexity grows (semantic changes, conceptual and process changes), as result not only of the change itself but due to poorly documented decisions and subsequent rework, together with the repetition of already performed solutions ("reinventing the wheel").

On the other hand, the NEGOSEIO approach has a drawback which is the negotiation time itself. While this time could be seen as growing exponentially as well due to the increasing complexity of the interoperability, the results showed that the formalisation of the negotiations together with the increasing knowledge accumulated by the rule engine from previous negotiation allowed the rule engine to suggest the negotiating partners decisions based on previous negotiations, which improved a lot the negotiation time, so generically this time could be observed as growing linearly with the complexity of the systems.

Finally, the complete NEGOSEIO (negotiation + reestablishment time) was also determined, and while this approach had the drawback of being penalised with the negotiation time when the systems were very simple, with the growing complexity, it could be observed that this time was even higher than the sum of negotiation time and traditional reestablishment, especially because the negotiations promoted more reusable and mature solutions (which take a bit longer to implement).

But it could be noticed that with the growth of the complexity, the growing reuse of these mature solutions caused a stabilisation on the reestablishment time, making it become very close to the negotiation time, as most of the collaboration services were already defined. Figure 5.4 depicts the evolution of the reestablishment time (or of the downtime) with the growth of the complexity of the interoperability.

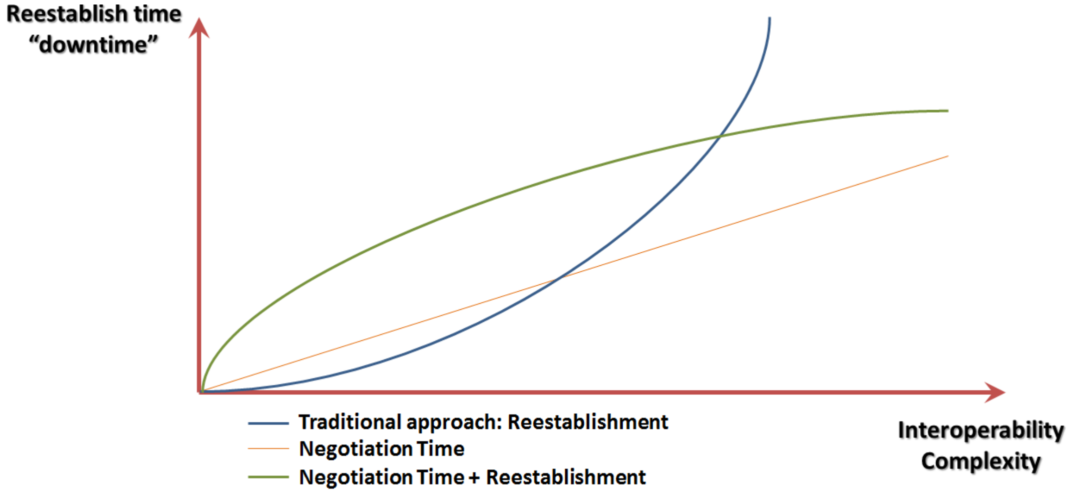


Figure 5.4: Traditional approach vs. NEGOSEIO approach

The conclusions that can be elicited from this proof-of-concept are that NEGOSEIO is a framework that aims to be helpful in the future internet and in the near-future systems, which shall be more and more complex, where concern is not being placed on the syntax and middleware, but instead on complex and intricate specifications of semantics and processes. In the current systems, NEGOSEIO can be less performant than the traditional methods, but only because it thrives to reach a better solution, one that is able to be reused in a later phase.

5.3. Hypotheses Validation

Based on the tested outcomes, the research is able to determine that the proposed framework, which formalises knowledge and alternatives analysis, indeed promotes the search for the solutions that maximise the performance. However, in the current state-of-the-art of the somewhat immature internet and collaboration requirements and needs, this is still not able to be observed, as it is faster to patch a quick solution than to invest on a mature and enduring solution. Nevertheless, it could be observed that even in the current days, as soon as the collaboration demands increase slightly, the NEGOSEIO framework already is able to perform better than the traditional approach (see Figure 5.4). Considering the natural trend of computer systems, this is a reality that is foreseen as not very far.

The NEGOSEIO framework is able to capture knowledge not only from the static assets (technology, documentation, tools, practices) but the formalisation of the negotiation steps also allows the capture of the business interactions, the strategies, motivations and intentions, and by having all this knowledge formalised it will be possible then to establish policies and access restrictions which permit the definition of private knowledge and public knowledge. This is concluded to be better than the current state of relying knowledge to be maintained solely on the heads and notes of the Expert Engineers.

The fact that the NEGOSEIO framework submits early questionnaires and surveys and assesses negotiation processes from the very start of the interactions. This practice intends to apply the project management discipline of Risk Management to the enterprise interoperability. This indeed permits the establishment of a sustainable and progressive knowledge base, improves the estimates and predictability of the system. One of the drawbacks that were detected was the large effort that is needed to perform these activities, and it indirectly promotes that enterprises get a bit reserved regarding the establishment of new partnerships.

Indeed the NEGOSEIO framework allows the definitions and justification by stating the pros but also the cons of the interoperable environment definitions, providing formal justification for the performed decisions.

The use of a negotiation framework has proven (see section 5.1.1) that the preconized solutions are mature and hence that it promotes the use of standards and durable solutions.

Based on these conclusions, the author considers that negotiations indeed improve interoperability and that this is not achieved at the cost of losing the private knowledge assets, that the proposed framework (which uses negotiations) help to the establishment of new and stable interoperable environment by promoting the use of better solutions. The testing results also demonstrate that when the complexity of the interoperability starts to grow, the use of negotiations help stabilising the "Downtime" (see Figure 5.4).

Hence, with its contribution for the formalisation, improve and endurance of business exchanges, it can be concluded that negotiation indeed is able to contribute to improve the sustainability of Enterprise Interoperability.

5.4. Validation by Industry/Stakeholders

The validation of the current framework and its application to the proposed business case, together with the results and conclusions were compiled in a report that was submitted to the directorate of ESA-CDF [143]. A total of six stakeholders with different backgrounds and interests reviewed this document and answered a short questionnaire which has the purpose of peer validation. The reviewers of the proposed report were:

- **Massimo Bandecchi**, founder and head manager of the ESA-CDF, and CDF Team Leader;
- **Sam Gerené**, ESA Space Systems Engineer, and application developer;
- **Hans Peter de Koning**, ESA Space Systems Engineer, developer of SysML and STEP models, responsible for the ESA-OCDT project;
- **Carlos Corral**, ESA Space Systems Engineer, and CDF Team Leader;
- **Tiago Soares**, ESA Space Systems Engineer;
- **Arne Matthyssen**, former ESA Space Systems Engineer, manager of J-CDS.

The review of the proposal consisted in three different areas. The first considered the general opinion about the framework, the presented ideas and the outcomes of the proposal. The reviewers in general concluded that the analysis that was performed to the ESA-CDF environment captured the CDF environment and processes adequately, though the described document was seen as too theoretical. This was considered by the author a natural comment, considering that the purpose of the document was academic and the focus of the CDF engineers is essentially practical, and focused into transforming the described concepts into real assets.

The reviewers admitted that the described problems are factual and actually occur in the ESA-CDF, though it was considered that the impact of the assessment was too focused on interoperability, whereas their concern lies more in the mission itself and in its constrains. Some considered the described approach important in the present environment and especially for the future, while others stated that the interoperability problem is still not seen as determinant in the CDF environment, as they trust the experience and maturity of the involved aerospace engineers. Besides, some considered that the fact that the CDF relies on a very rigid formal structure for exchanging data parameters is very effective in containing major issues regarding interoperability.

The author’s opinion is aligned with them, as most current enterprises are still able to work with the current interoperability levels. This research is not targeted in defining a short-term solution for a problem that is still rarely occurring, rather to a medium-term reality when Enterprise Interoperability shall be much more complex in volume, contents and shape of the exchanged information, where the abstraction level will no longer relate to data syntax, numbers and figures, but to semantics, knowledge, beliefs, trust and understanding.

Another thing that is important to figure, even in the current environment, and that was also the opinion of some of the interviewed reviewers, is that CDF should not concern only with the contained and controlled interoperability between domains, but it must extend this concern to each domain’s supply chain, the SMEs that provide tools and services that are then used in the work of each domain. Though the business case of this research is the ESA-CDF, one should not forget that the actual target for its application is the common networked SME, which is by far less controlled and less mature.

Regarding the subject of negotiations, it was consensual that formal negotiation mechanisms can be very useful to solve interoperability problems, especially with regards to decision making and trade-off solutions.

It was considered that one of the major problems, which is described and tackled in this research is that the knowledge is lying mostly on the domain engineers and they recognise the dangers in this behaviour, however they found that it is still quite difficult to capture this tacit information and knowledge in order to reuse it. However, they stated that if it would be possible to capture, model and store this knowledge and to develop a quick way of making this information available, then this would be a major improvement for the CDF business.

The opinion of some of the reviewers is that while the approach of using MDA and MDI is very good for capturing knowledge, the technologies lying in its implementation (especially the transformation from CIM to PIM, and MDI) are still quite immature for providing short-term solutions, and they are afraid that a proper solution is still far from feasible. Another thing that was noted was the fear of not being able to control and master such solutions.

There are also concerns that the excessive knowledge capture in the CDF environment might somehow contribute to limit/constrain the current pre-disposition of the engineers on creativity.

One very interesting comment from a reviewer regarded an important concept which is the heterogeneity of the real environment, which regards the difficulty in being able to describe all the factors which characterise a scenario. The author agrees with this concern and trusts that harmonisation and categorisation processes are being handled with the help of statistic processes, as this was also a concern regarding enterprises and projects, and it is being dealt with currently.

Regarding the approach of distribution/decentralisation proposed with the use of Cloud Computing, it was consensual that these technologies are very beneficial and provide appropriate means for the future, although some stated that the cloud approach though valid and useful for system and IT infrastructure management (i.e., IaaS and PaaS), does not fit for the storage of the important CDF proprietary data, due to its security problems which are not yet solved.

While the author agrees with these comments and concerns (as they are stated in the future work session, section 6.4, and while there is still some difficulties in handling the legal problems regarding the storage of proprietary data outside of the EU space, as this is quite difficult to be assured, there are already several approaches to solve this problem. One is the use of private clouds, but while this might be feasible in a large organisation like ESA, it is hardly possible for SMEs. There are however very interesting developments in the area of resilient information over clouds [144] which might provide some means to ensure privacy while storing the information in clouds.

A question was also stated concerning the best contributions and major concerns regarding the application of NEGOSEIO to the CDF environment and the most considered aspect regards knowledge management and capture, while the other aspects were less considered especially because the reviewers considered the structure to be too complex for the engineers to handle and manage it.

In terms of the applicability of NEGOSEIO on the CDF environment, all the reviewers stated it would be positive and useful to develop a small-scaled experiment within CDF, despite some concerns that the knowledge gathering and management on the CDF will be very complex due to the amount of information. Therefore, it was considered viable to build a small-scaled model although considering it won't be a trivial task. Most of the interviewed experts also agreed in participating in such a proof-of-concept, contributing with their expertise and knowledge of the interoperating environment, although agreeing that this is a project to be considered in a mid-term conception (3-4 years distance).

The CDF engineers review committee was also asked to provide their opinion regarding the platform disclosure itself, the structure of the proposal and whether the exposed ideas would be clear, easy to identify and map with the business case real problems and solutions. The consensus was that the proposal's ideas and concepts were well understood and clear though sometimes the details and abstractions were not so easy to follow, and that the document could be enriched if it provided more examples using the CDF reality instead of remaining at a more abstract level.

Regarding a final consideration about the proposal, the reviewers stated that the parts of the document that were more clear regarded the analysis of the CDF business and the description of the proposed solutions, including the ideas concerning the sustainable Enterprise Interoperability. The author was satisfied by this feedback, although realising that the sent document was very focused on the practical targets of the proposed solution and intentionally skipped the theoretical and mathematical specificities of the negotiation model that are detailed on section 3.

6. Final Considerations and Future Work

> *“*
> *You may say I'm a dreamer, but I'm not the only one.*
> *I hope someday you'll join us, and the world will be as one.*
> *”*
>
> *Imagine – John Lennon*

6.1. Synthesis of the Work

This research work is the result of five years of academic research, which complement more than fifteen years of hands-on experience in the field of IT business. This complementing experience resulted in a clear and unbiased view of the real world. It started by developing a set of research areas and questions, which were then synthesised into a proposed research question and sub-questions, and the elaboration of the corresponding hypotheses.

The study comprised an initial period of about one year of performing a thorough review of the literature, where the researched subjects were scrutinised and where an analysis was performed using the described methodology to ascertain the most important sources and related literature, mainly focusing on the problems and subjects of interoperability, services and model-driven engineering. This included a revision on some developing outcomes of the ATHENA project, which enlightened the undergoing state of the art in interoperability and MDA/MDI research.

The work developed over this research was then enriched by the author’s participation in the ESA project OCDS, which provided the foundations for the development of the proposed framework. Other motivations for this work were the author’s extensive participation in CMMI activities, which provided the basis for the need of formalisms and best-practices, and in project management activities which provided the background for techniques like risk management or parametric estimations.

The early analysis from this research on ESA and the interaction with their business and their Concurrent Engineering practices led to the conclusion are that their business interoperability was often developed over tacit knowledge, that it was fragile and with a high probability of breaking apart. Although the CDF model in ESA is very strict, in order to promote order and understanding, this is not the case in the regular SME market, and so the ESA business case problems were easily generalised.

As negotiations were considered by literature the best way to solve business conflicts and as they also provide the proper formalism that was necessary, this was also the way to be followed. The purpose aimed allowing negotiation knowledge, decisions, and responsibilities to be exchanged and documented to improve business relationships. The ability to negotiate the interoperability solutions leads to maximised results, stronger capabilities and relationships, contributing to reduce the risk of losing interoperability. Negotiations were studied, followed by the development of a negotiation model, the negotiation strategies and all the mathematical and formal aspects.

This paradigm changed the course of the research towards the impact of negotiations and how they could and would be a means for achieving the resource purposes for interoperability. The research questions and hypotheses were then refined to their final shape. The developing framework then faced the challenge of needing mechanisms for adapting to the surrounding environment. MaS agents were then added to the framework to handle the detection of interoperability changes and to trigger the negotiation services into handling the reestablishment solutions, allowing the selection of the best solution by consensus of all parties.

The next step was to provide agents with a strong and formal model that they may analyse and use to detect changes (using variance analysis) and thus trigger the negotiations. This was the motivation to provide formal procedures for modelling, developing a holistic view of the enterprise business, and storing and documenting the business activities and data into MDA, MDI, B-MDA and B-MDI. These models were a natural evolution that allowed a comprehensive analysis of the processes throughout the different abstraction layers. This integration also allowed an optimised analysis of the alternative solutions and by adding the analysis of lessons-learned on past activities leads to maximised negotiation results, stronger negotiation capabilities and relationships.

As the target of the research was to build an adaptive framework to allow flexibility in terms of rearrangement, composition, reuse and scalability for seamless, sustainable interoperability, it was considered that the path to go would be to implement the resulting outcomes of the business modelling in the shape of services, particularly focusing on the blooming concept of cloud computing. Hence the framework was enriched with the definition of a SOA supported by a SaaS platform, and the data structure and other infrastructures were defined on an IaaS and PaaS virtual environment.

These were the motivations and major steps towards the creation and proposal of NEGOSEIO to solve interoperability discrepancies in enterprise interactions, allowing a sustainable interoperability networking environment along its life cycle. This led to the development of a proof-of-concept prototype which would prove the research concepts.

A difficult challenge for the author, as well as for its validation reviewers also, was to be able to separate the research target which regards negotiations for determination of interoperability solutions and the ESA-CDF business which regards “negotiating” parameters towards a space mission.

The next natural step was to integrate and adapt the prototype to tackle the problems detected on ESA-CDF, and finally, to provide a report for validation and proposal for a future implementation on the ESA-CDF environment.

Throughout its course, this research was enriched by the collaboration of several partners and colleagues into very profitable brainstorming sessions and the participation in research meetings and international conferences (e.g., I-ESA, OTM), where the developed workshops and discussion work groups helped a lot in clarifying concepts and shedding light into the ongoing work. The research continued its development, consolidating all the gathered artefacts and developing concepts into the elaboration of this final report, which is proposed as the author’s major contribution to the science basis in the underlying subjects.

6.2. Main Scientific and Technical Contributions

The scientific contributions of this research are clearly highlighted in this document in the sections entitled “Research Contribution” in each researched topic on section 2, from which a natural emphasis should be given to the fields of Sustainability of Enterprise Interoperability (section 2.4.2) and Negotiations (section 3). The major scientific contributions also include the definition of the NEGOSEIO framework (section 4).

This includes the delineation of the NEGOSEIO methodology for capturing knowledge, the modelling of knowledge in the shape of MDA, MDI, B-MDA and B-MDI definitions, the creation of a cloud service-based ecosystem to handle interoperability transformations, the creation of an Agent-based infrastructure for detecting changes in the interoperability, but especially, the establishment of a negotiation model and negotiation infrastructure to handle the interoperability issues.

These were built with the objective of improving the solutions to achieve interoperability where it does not exist, and to determine the best solution to re-establish the interoperability when it was lost, in a way to maximise its endurance while reducing the time, effort and cost to accomplish the task, thus contributing to improve the sustainability of the enterprise interoperability.

The research contributions also include the analysis of the state of the art in the related fields, and the creation of a proof-of-concept where the proposed innovations were implemented, in the scope of a real business case in the European Space Agency (ESA). The proof-of-concept was then tested and its results were compiled into a report that was sent to ESA for peer validation.

Besides the referred peer validation from the real business case experts (section 5.4), this research was also supported by peer validation in the acceptance and publication of papers in academic conferences and journals, and reinforced by live participation in referenced conferences and workshops in the fields of research.

International Journals:

- A. Cretan, C. Coutinho, B. Bratu, and R. Jardim-Goncalves, “NEGOSEIO: A Framework for Negotiations toward Sustainable Enterprise Interoperability”, paper published to the IFAC Journal Annual Reviews In Control (ISI-WoS / IF = 1.319);
- R. Jardim-Goncalves, A. Cretan, C. Coutinho, C. Silva, and P. Ghodous, “Negotiation Framework Based on Ontologies for Supporting Enterprise Interoperability in Cloud-Based Environments”, paper published to the International Journal of Electronic Business Management;
- C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Sustainable Interoperability on Space Mission Feasibility Studies”, Paper submitted to the Journal Special Issue of Computers in Industry, no. Interoperability and Future Internet for Next-Generation Enterprises (ISI-WoS / IF = 1.529).

Conferences with Peer Revision:

- A. Tøn, R. Richardson, A. Relvas, T. Christiansen, J. Haenisch, and C. Coutinho, “The ESA OCDS Project - Enhancing the Concurrent Design Concept”, in 10th NASA-ESA Workshop on Product Data Exchange (PDE2008);
- A. Relvas, R. Richardson, C. Coutinho, A. Ribeiro, and C. Dunne, “Domain Design Application (DDA) and enhancements of communication layer for CDF”, in 3rd International Workshop on System & Concurrent Engineering for Space Applications (SECESA 2008);
- C. Coutinho, “Sustainable Interoperability in Business Relationships”, in Doctoral Symposium - 6th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2012);
- C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Sustainable Interoperability Framework for supporting Negotiation Processes”, in 6th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2012);
- C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Cloud-based negotiation for sustainable Enterprise Interoperability”, in 18th International Conference on Engineering, Technology and Innovation (ICE2012);
- R. Jardim-Goncalves, A. Cretan, C. Coutinho, M. Dutra, and P. Ghodous, “Ontology Enriched Framework for Cloud-based Enterprise Interoperability”, in 19th ISPE International Conference on Concurrent Engineering (CE2012);
- A. Cretan, C. Coutinho, B. Bratu, and R. Jardim-Goncalves, “A Framework for Sustainable Interoperability of Negotiation Processes”, in 14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM’12);
- C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Negotiations Framework for Monitoring the Sustainability of Interoperability Solutions”, in International IFIP Working Conference on Enterprise Interoperability (IWEI 2012);
- C. Coutinho, R. Jardim-Goncalves, and A. Cretan, “A Framework for Negotiation-Based Sustainable Interoperability for Space Mission Design”, in 7th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N’2012).

6.3. Final Considerations

This research focused on describing common problems found in the Enterprise Interoperability domain, particularly on the proposed business case of feasibility studies for aerospace missions, and proposed a collaborative framework to enhance business knowledge, to allow the adaptation of enterprises, and the negotiation of solutions for the detected interoperability changes. By adopting the proposed framework, which servitises the technological, business and people views, SMEs are expected to improve the interoperability with its partners, achieving a stable stage of interoperability between the existing partners, and a faster integration with new ones.

Also, more than just supporting SMEs in their operation, the author foresees that by adopting the proposed framework, major contractors and final customers like the ESA-CDF will have a mechanism that allows them to follow and influence (monitor and control) the interoperability solutions that are taken in subcontracted enterprises, aligning them with their objectives. This monitoring and control won't be a problem for SMEs, because the formalisation of the negotiations into the defined negotiation model allows SMEs to perform the negotiation keeping their knowledge assets encapsulated in the negotiation model's entities, thus allowing them to negotiate without exposing them to the competition. Through the negotiation mechanism, the interoperable environment will be robust and easy to maintain.

This research concluded that interoperability typically breaks because it is based on poorly described knowledge. The integration of formal procedures for modelling, storing and documenting the business activities into MDA, MDI, B-MDA and B-MDI models allows a comprehensive analysis of the processes and of the possible alternatives. Adding the ability to negotiate the interoperability solutions leads to maximised results, stronger capabilities and relationships, thus contributing to reduce the risk of losing interoperability. The use of an adaptive framework that allows flexibility in terms of rearrangement, composition, reuse and scalability will result in a seamless, robust interoperability.

More than simply being interoperable, the ontology-enriched framework and the modelling and management of parallel and concurrent negotiations, aims to open the market to broader discovery of opportunities and partnerships. The ability to reach and interoperate with more enterprises stimulates more business opportunities and stronger and healthier interactions, thus promoting sustainability.

The conclusions from this research are that sustainability of the enterprise interoperability is possible when a constant and periodic maintenance is performed to the interoperable environment, and that negotiations are essential when acting on complex business environments to reduce the time where there is no operation, and improve resource and time management.

Currently, interoperability between the involved parties in a negotiation is often not reached or maintained due to failure in adapting to new requirements, parties or conditions. The use of the NEGOSEIO adaptive platform results in a seamless, sustainable interoperability which favours its maintenance across time.

6.4. Future Work

This research plans to improve the functionalities of the proposed framework in the near future, adding a renegotiation mechanism based on the experience learned from past negotiations, intending to accelerate the actual negotiation process and achieve more intelligent, accurate and faster results towards the sustainability of networked EI environments.

Further, still within the negotiation subject, it foresees the definition of other framework tools to support contract management, negotiation and renegotiation, secure access to the negotiation arena, share/isolation of data between negotiation subjects, the possibility to reuse/retake stalled negotiations, improving the negotiation processes to avoid negotiation deadlocks, and provision for each negotiating agent of a library of protocols.

The development of this framework is still conditioned to the improvement of issues that remain under research on MDI, concerning the horizontal transformations in the various abstraction levels, and SOA, with issues still rising against service discovery and service composition and orchestration.

Work is being developed in the analysis of existing implementations in Cloud systems and efforts and advances in the Cloud federation area; concerns about multi-tenancy, security and privacy regarding the valuable proprietary knowledge stored in Cloud support; the resilience of the stored negotiation data in the cloud, and managing privacy aspects as the parties should be able to seamlessly interoperate but still to maintain their data free from prying eyes. Also several issues need to be solved from non-disclosure of participating parties to secure access to the interoperative environment.

Future work also regards the assembly of the framework environment in the ESA-CDF facility, as well as the analysis of other use-cases related to the studied business case that commonly lead to loss of interoperability.

Another area which may benefit a lot from the researched work concerns the growing fears regarding business continuity and resilience of business processes and assets throughout time. The NEGOSEIO framework poses itself as an effective approach to model the technological assets but also the business process activities which are required to establish a good basis for business continuity management, and a next step in the field of configuration management assets.

7. REFERENCES

“

That’s the time you must keep on trying,
Smile, what’s the use of crying?
You’ll find that life is still worthwhile
If you just smile.

”

Smile – Charles Chaplin

[1] “Future Internet Enterprise Systems (FInES)”, 2012. [Online]. Available: http://www.fines-cluster.eu. [Accessed: 06-Dec-2011].

[2] L. Rodriguez-Roselló and M. Campolargo, “The Future Internet”, *Lecture Notes in Computer Science: The Future Internet*, vol. 6656, 2011.

[3] Eurescom, “European Future Internet Portal”, 2012. [Online]. Available: http://www.future-internet.eu. [Accessed: 10-Jun-2011].

[4] M. B. Chrissis, M. Konrad, and S. Shrum, *CMMI for Development - Version 1.3*, Third Edit. Addison Wesley Professional, 2011, p. 688.

[5] PMI, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. Project Management Institute, 2008, p. 459.

[6] “ESA-OCDS Community Portal”, 2012. [Online]. Available: http://atlas.estec.esa.int/uci_wiki. [Accessed: 20-Mar-2012].

[7] C. Agostinho, R. Jardim-Goncalves, J. Sarraipa, F. Lampathaki, S. Koussouris, Y. Charalabidis, J. Psarras, P. Assogna, M. Missikoff, K. Popplewell, E. Silva, and J. Ferreira, “FInES Deliverable D2.4: EISB Models & Tools Report”, 2011.

[8] Google, “Google Scholar”, 2012. [Online]. Available: http://scholar.google.com. [Accessed: 12-Jun-2012].

[9] A.-W. Harzing, “Publish or Perish”. 2007.

[10] L. V. Bertalanffy, *General System Theory: Foundations, Development, Applications (Revised Edition)*. George Braziller Inc., 1969.

[11] OMG, “Model Driven Architecture”, 2011. [Online]. Available: http://www.omg.org/mda. [Accessed: 20-Dec-2011].

[12] J. Ullberg, R. Lagerstrom, M. van Sinderen, and P. Johnson, “Architecture Modeling for Interoperability Analysis on the Future Internet”, in *6th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA2012)*, 2012.

[13] Y. Ducq, D. Chen, and B. Vallespir, “Interoperability in enterprise modelling: requirements and roadmap”, *Advanced Engineering Informatics*, vol. 18, no. 4, pp. 193–203, Oct. 2004.

[14] OMG, “OMG Systems Modelling Language”, 2012. [Online]. Available: http://www.omgsysml.org/. [Accessed: 24-Apr-2012].

[15] J. A. Estefan, “Survey of Model-Based Systems Engineering (MBSE) Methodologies”, 2008.

[16] OMG, “OMG MBSE”, 2012. [Online]. Available: http://www.omgwiki.org/MBSE/doku.php.

[17] J. Ferreira, C. Agostinho, J. Sarraipa, and R. Jardim-Goncalves, “Monitoring morphisms to support sustainable interoperability of enterprise systems”, in *6th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N’2011)*, 2011, pp. 71–82.

[18] J. N. Martin, *Systems Engineering Guidebook*. CRC Press, 1996.

[19] C. E. Dickerson, “Model Driven Architecture for Model Based Systems Engineering: Emerging Concepts for Systems Engineering”, in *Paper presented at the SEDC Integration Workshop*, 2007, no. September.

[20] ISPA, *Parametric Estimating Handbook*, no. April. 2008.

[21] ESA/ECSS WG, “ECSS-E-TM-10-25A Technical Memorandum”, 2010.

[22] A. Ton, R. Richardson, A. Relvas, T. Christiansen, J. Haenisch, and C. Coutinho, “The ESA OCDS Project - Enhancing the Concurrent Design Concept”, in *10th NASA-ESA Workshop on Product Data Exchange (PDE2008)*, 2008.

[23] “Internet of Things - An action plan for Europe”, CORDIS, p. 13, 2009.

[24] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, and P. Doody, “Internet of Things Strategic Research Roadmap”, in *Internet of Things: Global Technological and Societal Trends*, River Publishers, 2011, pp. 9–52.

[25] “Internet of Services”, 2012. [Online]. Available: http://www.internet-of-services.com.

[26] J. Cardoso, K. Voigt, and M. Winkler, “Service Engineering for the Internet of Services”, in *10th International Conference on Enterprise Information Systems ICEIS 2008*, 2008, pp. 15–27.

[27] P. P. Maglio, S. Srinivasan, J. T. Kreulen, and J. Spohrer, “Service Systems, Service Scientists, SSME, and Innovation”, *Communications of the ACM - Services Science*, vol. 49, no. 7, pp. 81–85, 2006.

[28] W. Murphy, “IBM Service Science”, 2012. [Online]. Available: http://service-science.info/. [Accessed: 14-Mar-2012].

[29] “MSEE Project”, 2012. [Online]. Available: http://www.msee-ip.eu/project-overview. [Accessed: 10-Apr-2012].

[30] ISO/CEN, “EN/ISO 19440: Enterprise integration - Constructs for enterprise modelling”, 2004.

[31] G. Doumeingts, B. Vallespir, and D. Chen, “GRAI Grid Decisional Modelling”, in *Handbook on Architectures of Information Systems*, Springer, 2006, pp. 321–346.

[32] Athena Consortium, “Athena Interoperability Framework”, 2011. [Online]. Available: http://www.modelbased.net/aif. [Accessed: 20-Dec-2011].

[33] R. Grangel, R. B. Salem, J.-P. Bourey, N. Daclin, and Y. Ducq, “Transforming GRAI Extended Actigrams into UML Activity Diagrams: a First Step to Model Driven Interoperability”, in *3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007)*, 2007, pp. 447–458.

[34] I. S. C. 21 Committee, *IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications , and Loads*, no. September. 2011.

[35] C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Sustainable Interoperability on Space Mission Feasibility Studies”, *Paper submitted to the Journal Special Issue of Computers in Industry*, no. Interoperability and Future Internet for Next-Generation Enterprises, 2012.

[36] “European Community Enterprise & Industry”, 2012. [Online]. Available: http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/index_en.htm. [Accessed: 10-Jun-2012].

[37] C. Coutinho, “Sustainable Interoperability in Business Relationships”, in *Doctoral Symposium - 6th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2012)*, 2012, pp. 1–10.

[38] L. Nie, X. Xu, D. Chen, G. Zacharewicz, and D. Zhan, “GRAI-ICE Model Driven Interoperability Architecture for Developing Interoperable ESA”, in *Enterprise Interoperability IV*, Springer, 2010, pp. 111–121.

[39] C. Agostinho, F. Correia, and R. Jardim-Goncalves, “Interoperability of Complex Business Networks by Language Independent Information Models”, in *New World Situation: New Directions in Concurrent Engineering - Proceedings of the 17th ISPE International Conference on Concurrent Engineering (CE 2010)*, 2011, pp. Part 2, 111–124.

[40] F. Farinha, R. Jardim-Goncalves, and A. Steiger-Garcao, “Integration of cooperative production and distributed design in AEC”, *Advances in Engineering Software*, vol. 38, no. 11–12, pp. 772–779, Nov. 2007.

[41] “International Virtual Laboratory for Enterprise Interoperability”, 2012. [Online]. Available: http://www.interop-vlab.eu. [Accessed: 14-Mar-2012].

[42] B. Pataki, L. Kovács, C. Guglielmina, and A. Arezza, “ABILITIES to Support a Federated Architecture Based Interoperability Bus with Groupware and Multimedia”, in *Enterprise Interoperability II*, vol. 60, 2007, pp. 415–426.

[43] Athena Consortium, “ATHENA Project: Specification of interoperability framework and profiles, guidelines and best practices”. 2007.

[44] Y. Lemrabet, M. Bigand, D. Clin, N. Benkeltoum, and J.-P. Bourey, “Model Driven Interoperability in practice : preliminary evidences and issues from an industrial project”, in *First International Workshop on Model-Driven Interoperability (MDI’10)*, 2010, pp. 3–9.

[45] D. Chen, G. Doumeingts, and F. Vernadat, “Architectures for enterprise integration and interoperability: Past, present and future”, *Computers in Industry*, vol. 59, no. 7, pp. 647–659, Sep. 2008.

[46] IST, “FUSION Project”, 2010. [Online]. Available: http://www.fusionweb.org. [Accessed: 27-Jun-2010].

[47] A.-J. Berre, F. Liu, J. Xu, and B. Elvesaeter, “Model Driven Service Interoperability through Use of Semantic Annotations”, in *International Conference on Interoperability for Enterprise Software and Applications (IESA’09)*, 2009, pp. 90–96.

[48] R. Jardim-Goncalves and A. Grilo, “SOA4BIM: Putting the building and construction industry in the Single European Information Space”, *Automation in Construction*, vol. 19, no. 4, pp. 388–397, Jul. 2010.

[49] M. Lopez-Sanz, C. J. Acuna, C. E. Cuesta, and E. Marcos, “Defining Service-Oriented Software Architecture Models for a MDA-based Development Process at the PIM level”, *Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, pp. 309–312, Feb. 2008.

[50] U. Zdun and S. Dustdar, “Model-driven and pattern-based integration of process-driven SOA models”, *International Journal of Business Process Integration and Management*, vol. 2, no. 2, pp. 109–119, 2007.

[51] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge Engineering : Principles and methods”, *Data & Knowledge Engineering*, vol. 25, pp. 161–197, 1998.

[52] J. Sarraipa, R. Jardim-Goncalves, and A. Steiger-Garcao, “MENTOR: an enabler for interoperable intelligent systems”, *International Journal of General Systems*, vol. 39, no. 5, pp. 557–573, Jul. 2010.

[53] M. Dutra, P. Ghodous, and O. Kuhn, “A Generic and Synchronous Ontology-based Architecture for Collaborative Design”, *Concurrent Engineering*, vol. 18, no. 1, pp. 65–74, Feb. 2010.

[54] J. M. Torres and Q. Wijnands, “Distributed Agents for Multi-Rover Autonomy”, *20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 53–58, Jun. 2011.

[55] A. Grilo and R. Jardim-Goncalves, “Value proposition on interoperability of BIM and collaborative working environments”, *Automation in Construction*, vol. 19, no. 5, pp. 522–530, Aug. 2010.

[56] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Service-Oriented Computing: a Research Roadmap”, *International Journal of Cooperative Information Systems*, vol. 17, no. 02, p. 223, 2008.

[57] H. Demirkan, R. J. Kauffman, J. A. Vayghan, H.-G. Fill, D. Karagiannis, and P. P. Maglio, “Service-oriented technology and management: Perspectives on research and practice for the coming decade”, *Electronic Commerce Research and Applications*, vol. 7, no. 4, pp. 356–376, Dec. 2008.

[58] S. Dustdar and W. Schreiner, “A Survey on Web Services Composition”, *International Journal of Web and Grid Services*, vol. 1, no. 1, pp. 1–30, 2004.

[59] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. 2005.

[60] S. Ran, “A Model for Web Services Discovery With QoS”, *ACM SIGecom Exchanges*, vol. 4, no. 1, pp. 1–10, 2003.

[61] I. Foster, “Globus Toolkit Version 4 : Software for Service-Oriented Systems”, in *Lecture Notes in Computer Science*, Springer, 2005, pp. 2–13.

[62] W.-T. Tsai, X. Sun, and J. Balasooriya, “Service-Oriented Cloud Computing Architecture”, in *Seventh International Conference on Information Technology: New Generations (ITNG 2010)*, 2010, pp. 684–689.

[63] D. A. Menascé, E. Casalicchio, and V. Dubey, “On optimal service selection in Service Oriented Architectures”, *Performance Evaluation*, vol. 67, no. 8, pp. 659–675, Aug. 2010.

[64] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”, *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[65] R. Sharma and M. Sood, “Cloud SaaS and Model Driven Architecture”, in *International Conference on Advanced Computing and Communication Technologies (ACCT11)*, 2011, no. Acct, pp. 978–981.

[66] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges”, *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, Apr. 2010.

[67] K. Jeffery and B. Neidecker-Lutz, “The Future of Cloud Computing: Opportunities for European Cloud Computing Beyond 2010”, *Analysis*. p. 71, 2010.

[68] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, “Capacity Leasing in Cloud Systems using the OpenNebula Engine”, in *Cloud Computing and Applications 2008 (CCA 08)*, 2008.

[69] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan, “The Reservoir model and architecture for open federated cloud computing”, *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 1–11, 2009.

[70] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, “Green Cloud Computing : Balancing Energy in Processing , Storage , and Transport”, *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.

[71] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing”, *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[72] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, “Key Challenges in Cloud Computing to Enable the Future Internet of Services”, *IEEE Internet Computing*, no. 99, 2012.

[73] “Wikipedia: Definition of Sustainability”, 2012. [Online]. Available: http://en.wikipedia.org/wiki/Sustainability. [Accessed: 10-Jan-2012].

[74] R. Jardim-Goncalves, C. Agostinho, and A. Steiger-Garcao, “A reference model for sustainable interoperability in networked enterprises : towards the foundation of EI science base”, *International Journal of Computer Integrated Manufacturing*, vol. 25, no. 10, pp. 855–873, 2012.

[75] Y. Ducq, D. Chen, and G. Doumeingts, “A contribution of system theory to sustainable enterprise interoperability science base”, *Computers in Industry*, Sep. 2012.

[76] S. Khouri, I. Boukhari, L. Bellatreche, E. Sardet, S. Jean, and M. Baron, “Ontology-based structured web data warehouses for sustainable interoperability: requirement modeling, design methodology and tool”, *Computers in Industry*, pp. 1–14, Sep. 2012.

[77] R. Jardim-Goncalves, K. Popplewell, and A. Grilo, “Sustainable interoperability: The future of Internet based industrial enterprises”, *Computers in Industry*, pp. 1–8, Sep. 2012.

[78] R. Jardim-Goncalves, C. Agostinho, and A. Steiger-Garcao, "Sustainable Systems ' Interoperability : A reference model for seamless networked business", in *2010 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2010)*, 2010, pp. 1785–1792.

[79] R. Jardim-Goncalves, A. Grilo, C. Agostinho, F. Lampathaki, and Y. Charalabidis, "Systematisation of Interoperability Body of Knowledge: the foundation for Enterprise Interoperability as a science", *Enterprise Information Systems*, vol. 6, no. 3, pp. 1–26, 2012.

[80] A. I. Oliveira and L. M. Camarinha-Matos, "Electronic Negotiation Support Environment in Collaborative Networks", *Advances in Information and Communication Technology*, vol. 372/2012, pp. 21–32, 2012.

[81] L. Telesca, "Towards Open Decentralized Self-Learning Negotiation Environment for Business Ecosystems", in *2nd International Conference on Communication System Software and Middleware and Workshops*, 2007.

[82] J. L. de la Rosa, N. Hormazábal, S. Aciar, G. A. Lopardo, A. Trias, and M. Montaner, "A Negotiation-Style Recommender Based on Computational Ecology in Open Negotiation Environments", *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2073–2085, 2011.

[83] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", 1993.

[84] E. Oliva, P. McBurney, A. Omicini, and M. Viroli, "Argumentation and Artifacts for Negotiation Support", *International Journal of Artifical Intelligence*, vol. 4, no. S10, pp. 90–117, 2010.

[85] P. Tolchinsky, S. Modgil, K. Atkinson, P. McBurney, and U. Cortés, "Deliberation Dialogues for Reasoning about Safety Critical Actions", *Autonomous Agents and Multi-Agent Systems*, 2011.

[86] T. Miller and P. McBurney, "Propositional Dynamic Logic for Reasoning about First-Class Agent Interaction Protocols", *Computational Intelligence*, vol. 27, no. 3, pp. 422–457, 2011.

[87] F. Parandoosh, "Evaluating Agent-Oriented Software Engineering Methodologies", in *2nd IEEE International Workshop on Soft Computing Applications (SOFA 2007)*, 2007, pp. 169–174.

[88] K. D. Ahmadi and N. M. Charkari, "Multi Agent Based Interactive Recommendation and Automated Negotiation System in E-Commerce", in *6th International Conference on Advanced Information Management and Service (IMS)*, 2010, pp. 279–284.

[89] K. D. Ahmadi, N. M. Charkari, and N. Enami, "E-Negotiation System Based on Intelligent Agents in B2C E-Commerce", *Advances on Information Sciences and Service Sciences*, vol. 3, no. 2, pp. 60–70, 2011.

[90] C. Russ and A. Walz, "MACSIMA : An Agent Framework for Simulating the Evolution of Negotiation Strategies in B2B-Networked Economies", *Lecture Notes in Business Information Processing*, vol. 98, no. 3, pp. 191–218, 2012.

[91] L. Duan, M. K. Doğru, U. Özen, and J. C. Beck, "Multi-agent Negotiation for Distributed Production Scheduling Problems", in *3rd International Workshop on Agent-based Complex Automated Negotiations*, 2010.

[92] L. Duan, M. K. Doğru, U. Özen, and J. C. Beck, "A negotiation framework for linked combinatorial optimization problems", *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 1, pp. 158–182, Apr. 2012.

[93] X. Luo, C. Miao, N. R. Jennings, M. He, Z. Shen, and M. Zhang, “KEMNAD : A Knowledge Engineering Methodology for Negotiating Agent Development”, *Computational Intelligence*, vol. 28, no. 1, pp. 51–105, 2012.

[94] H. Jazayeriy, M. Azmi-Murad, N. Sulaiman, and N. I. Udzir, “A review on soft computing techniques in automated negotiation”, *Scientific Research and Essays*, vol. 6, no. 24, pp. 5100–5106, 2011.

[95] M. Cao, “Multi-strategy Selection Supported Automated Negotiation System Based on BDI Agent”, *45th Hawaii International Conference on System Sciences*, pp. 638–647, Jan. 2012.

[96] A. Cretan, “Negotiation processes within inter-organizational alliances”, *Informatica Economica*, vol. 3, no. 3, pp. 41–46, 2008.

[97] G. Qing and C. Chun, “An Approach to Solving Deadlock in Multi-issue”, *The IEEE International Conference on Information Reuse and Integration (IRI 2003)*, pp. 48–52, 2003.

[98] R. J. Lin and S. T. Chou, “Mediating a Bilateral Multi-Issue Negotiation”, in *The IEEE International Conference on E-Commerce (CEC 2003)*, 2003, pp. 76–83.

[99] Z. C. Yang, S. Yang, Z. Q. Lei, Z. Y. Lin, and W. Z. Fei, “Research of the Strategy in Multi-issue Negotiations Deadlock”, in *The 2nd IEEE International Conference on Information Engineering and Computer Science (ICIECS)*, 2010, no. Grant 60473015, pp. 1–4.

[100] F. Huan and W. Lili, “Colored Petri Net Based Multi-Agents Dynamic Task Allocation Modeling and Analysis”, in *The 2nd IEEE International Conference on Information Science and Engineering (ICISE)*, 2010, pp. 1080–1083.

[101] Q. Liu, X. Cui, and X. Hu, “Conflict Resolution within Multi-agent System in Collaborative Design”, in *2008 International Conference on Computer Science and Software Engineering*, 2008, pp. 520–523.

[102] L. Pan, “Towards a Framework for Automated Service Negotiation in Cloud Computing”, in *The IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS2011)*, 2011, pp. 364–367.

[103] M. Wang, X. Wu, W. Zhang, F. Ding, J. Zhou, and G. Pei, “A Conceptual Platform of SLA in Cloud Computing”, in *The 9th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC2011)*, 2011, pp. 1131–1135.

[104] T. Groléat and H. Pouyllau, “Distributed inter-domain SLA negotiation using Reinforcement Learning”, in *The 12th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2011, pp. 33–40.

[105] N. R. Jennings, K. Sycara, and M. Wooldridge, “A Roadmap of Agent Research and Development”, *Autonomous Agents and Multi-Agent Systems*, vol. 38, no. 1, pp. 7–38, 1998.

[106] “AOS JACK Agent development platform”, 2011. [Online]. Available: http://aosgrp.com/products/jack/index.html.

[107] “JADE Agent development framework”, 2012.

[108] A. Kaddouci, H. Zgaya, S. Hammadi, and F. Bretaudeau, “Multi-agents Based Protocols for Negotiation in a Crisis Management Supply Chain”, in *8th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS ’09)*, 2009, pp. 143–150.

[109] C. Coutinho, A. Cretan, and R. Jardim-Goncalves, "Negotiations Framework for Monitoring the Sustainability of Interoperability Solutions", in *International IFIP Working Conference on Enterprise Interoperability (IWEI 2012)*, 2012, pp. 168–180.

[110] J. Hu and L. Deng, "An Association Rule-Based Bilateral Multi-Issue Negotiation Model", *Fourth International Symposium on Computational Intelligence and Design*, no. 1, pp. 234–237, Oct. 2011.

[111] A. Cretan, "A Model for Coordinating Negotiations among Virtual Enterprises", *Journal of Applied Quantitative Methods*, vol. 4, no. 1, pp. 54–68, 2009.

[112] K. Sycara and T. Dai, "Agent Reasoning in Negotiation", in *Advances in Group Decision and Negotiation Volume 4: Handbook of Group Decision and Negotiation*, D. M. Kilgour and C. Eden, Eds. Springer, 2010, pp. 437–451.

[113] E. A. Emerson, "Time of Time", in *17th International Symposium on Static Analysis, SAS 2010*, 2010, p. 1.

[114] G. Berry and G. Boudol, "The Chemical Abstract Machine", *Journal of Theoretical Computer Science*, vol. 96, no. 1, pp. 217–248, 1990.

[115] J. S. Pinto, "Sequential and Concurrent Abstract Machines for Interaction Nets", in *3rd International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2000)*, 2000, pp. 267–282.

[116] J. Andreoli, P. Ciancarini, and R. Pareschi, "Interaction Abstract Machines", *Communication*, vol. 280, pp. 1–18, 1993.

[117] A. Cretan, C. Coutinho, B. Bratu, and R. Jardim-Goncalves, "A Framework for Sustainable Interoperability of Negotiation Processes", in *14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'12)*, 2012.

[118] A. Relvas, R. Richardson, C. Coutinho, A. Ribeiro, and C. Dunne, "Domain Design Application (DDA) and enhancements of communication layer for CDF", in *3rd International Workshop on System & Concurrent Engineering for Space Applications (SECESA 2008)*, 2008, no. October, pp. 1–8.

[119] R. Jardim-Goncalves, N. Figay, and A. Steiger-Garcao, "Enabling interoperability of STEP Application Protocols at metadata and knowledge level", *International Journal of Technology Management*, vol. 36, no. 4, pp. 402–421(20), 2006.

[120] N. Georgantas, M. A. Ahaman, H. Ameziani, A. Pathak, and V. Issarny, "A Coordination Middleware for Orchestrating Heterogeneous Distributed Systems", in *Lecture Notes in Computer Science Volume 6646: Advances in Grid and Pervasive Computing*, 2011, pp. 221–232.

[121] A. Cretan, C. Coutinho, B. Bratu, and R. Jardim-Goncalves, "NEGOSEIO: A Framework for Negotiations toward Sustainable Enterprise Interoperability", *IFAC Journal Annual Reviews In Control*, 2012.

[122] R. Jardim-Goncalves, C. Agostinho, P. Malo, and A. Steiger-garcao, "Harmonising technologies in conceptual models representation", *International Journal Of Product Lifecycle Management*, vol. 2, no. 2, pp. 187–205, 2007.

[123] H. P. D. Koning, "MBSE and the ESA CDF", in *MBSE Workshop, INCOSE International Workshop 2012*, 2012.

[124] C. Coutinho, R. Jardim-Goncalves, and A. Cretan, “A Framework for Negotiation-Based Sustainable Interoperability for Space Mission Design”, in *7th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N’2012)*, 2012, pp. 93–102.

[125] C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Cloud-based negotiation for sustainable Enterprise Interoperability”, in *18th International Conference on Engineering, Technology and Innovation (ICE2012)*, 2012.

[126] C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Sustainable Interoperability Framework for supporting Negotiation Processes”, in *6th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2012)*, 2012.

[127] R. Sharma, “A Model-Driven Approach to Cloud SaaS Interoperability”, *International Journal*, vol. 30, no. 8, pp. 1–8, 2011.

[128] “RESERVOIR/OpenNebula Project”. [Online]. Available: http://opennebula.org. [Accessed: 29-Mar-2012].

[129] T. O. Group, “TOGAF® 9.1”, 2012. [Online]. Available: http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html.

[130] B. Henderson-Sellers and P. Giorgini, *Agent-Oriented Methodologies*. Idea Group Publishing, 2005.

[131] “OWL - Web Ontology Language”, 2012. [Online]. Available: http://www.w3.org/2004/OWL. [Accessed: 14-Nov-2011].

[132] R. Jardim-Goncalves, A. Cretan, C. Coutinho, M. Dutra, and P. Ghodous, “Ontology Enriched Framework for Cloud-based Enterprise Interoperability”, in *19th ISPE International Conference on Concurrent Engineering (CE2012)*, 2012, pp. 1–12.

[133] NASA, “NASA Parametric Cost Estimating Handbook”, 2002. [Online]. Available: http://cost.jsc.nasa.gov/pcehhtml/pceh.htm. [Accessed: 18-Feb-2009].

[134] M. Bandecchi, “The ESA Concurrent Design Facility (CDF): concurrent engineering applied to space mission assessments”, in *2nd Nordic Systems Engineering Boat Seminar (FinSE2001)*, 2001, pp. 1–36.

[135] “ESA-CDF”, 2012. [Online]. Available: http://www.esa.int/esaMI/CDF/. [Accessed: 10-Jan-2012].

[136] G. Kolfschoten, A. Matthyssen, and M. Fijneman, “Theoretical foundations for Concurrent Design”, in *4th International Workshop on System & Concurrent Engineering for Space Applications (SECESA 2010)*, 2010, no. 1.

[137] C. Coutinho, “ESA-CDF Interoperability Survey”, 2012. [Online]. Available: https://www.surveygizmo.co.uk/s3/856009/ESA-CDF-Interoperability-Issues.

[138] H. P. D. Koning, H. Eisenmann, and M. Bandecchi, “Evolving Standardization Supporting Model Based Systems Engineering”, in *4th International Workshop on System & Concurrent Engineering for Space Applications (SECESA 2010)*, 2010, no. 1.

[139] ESA/ECSS WG, “European Cooperation for Space Standardization”. [Online]. Available: http://www.ecss.nl. [Accessed: 12-Apr-2009].

[140] “ESA-SEIM”. [Online]. Available: http://atlas.estec.esa.int/uci_wiki/SEIM. [Accessed: 10-Jan-2012].

[141] "Eclipse Graphical Modelling Project", 2012. [Online]. Available: http://www.eclipse.org/modeling/gmp. [Accessed: 12-Jul-2011].

[142] "ESA-SERDL". [Online]. Available: http://atlas.estec.esa.int/uci_wiki/SERDL. [Accessed: 10-Jan-2012].

[143] C. Coutinho, "Proposal for a Negotiation Mechanism on Interoperability Solutions for the Concurrent Design Facility of the European Space Agency (ESA-CDF)", 2012.

[144] P. Verissimo, A. Bessani, and M. Pasin, "The TClouds architecture: Open and resilient cloud-of-clouds computing", *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, pp. 1–6, Jun. 2012.

ANNEX A – QUESTIONNAIRE ON ESA INTEROPERABILITY

Interoperability of High-performance Businesses (a Ph.D. research)

The ESA-CDF Business Use-case

Picture: ESA-CDF

This survey has the purpose to assess the interoperability issues on the CDF environment, regarding to the integration between CDF domains and the integration between each domain and its own set of tools / partners / subcontractors / suppliers. The survey is included in the scope of a Ph.D. research on interoperability of high-performance businesses. It will only take a few minutes of your time.

1) On a study, which is/are the CDF domain(s) you're regularly assigned to?

(Please select all that apply)

☐ AOCS (Attitude & Orbit Control)

☐ Communications

☐ Configuration

☐ Cost

☐ Data Handling

☐ Ground Systems and Operations

☐ Instruments

☐ Mechanisms and Pyrotechnics

☐ Mission Analysis

☐ Power

☐ Programmatic and AIV

☐ Propulsion

☐ Risk

☐ Simulation

☐ Structures

☐ Systems

☐ Thermal

☐ Other (Please Specify):

2) **In an average CDF Study, which are the domains that are more tightly related to YOUR domain, i.e. the domains that own important parameters for your design?**

(Please select all that apply)

- ☐ AOCS (Attitude & Orbit Control)
- ☐ Communications
- ☐ Configuration
- ☐ Cost
- ☐ Data Handling
- ☐ Ground Systems and Operations
- ☐ Instruments
- ☐ Mechanisms and Pyrotechnics
- ☐ Mission Analysis
- ☐ Power
- ☐ Programmatic and AIV
- ☐ Propulsion
- ☐ Risk
- ☐ Simulation
- ☐ Structures
- ☐ Thermal
- ☐ Other (Please Specify):

3) **In an average CDF Study, how many suppliers/partners do you interact with (other than CDF domains), i.e. subcontractors, suppliers, external or internal companies/departments not represented in the study?**

(Please do not include other CDF domains, already stated previously)

4) **Can you enumerate the main tools that are used on your domain to perform the study design, e.g. CATIA or other DST (domain-specific tool)?**

(Please exclude common tools/environments like Windows, Office/Excel)

5) **What type of interoperability do you have with your suppliers/partners/ subcontractors?**

(Please select all that apply)

☐ Shared Data/Tool Interfaces

☐ Web Services

☐ Shared Data workspace (e.g. a shared database)

☐ Business Workflows (i.e. shared work)

☐ Shared Semantics, Technical Concepts and Definitions

☐ Shared Tools

☐ Shared Knowledge Base (e.g. processes, policies)

☐ Strictly Formal Interfaces

☐ Other (Please Specify):

6) **If needed to develop your work, how would you select a new Design Tool / Subcontractor / Partner / Supplier? What are the main drivers for this choice?**

(Please select all that apply)

☐ Previous experience from other Studies or Projects

☐ They have Semantics/Concepts/Definitions similar to your own

☐ THEY take care of the integration with your environment

☐ Prices: Cheaper services or More Expensive services

☐ Rank in Forums and Magazines

☐ Lobby/Influence from Top Management

☐ Other (Please Specify):

7) **Which are your main difficulties regarding interoperability with subcontractors/ suppliers/partners?**

(Please select all that apply)

☐ Non-matching Semantic Concepts and Definitions: One of the parties needs to update its semantic data;

☐ Superficial knowledge regarding Business Concepts (on both sides): Not enough information available;

☐ Rigid Definitions and Structures: Allows data exchanges but does not allow proper update of data semantics;

☐ Other (Please Specify):

8) **Which are your main difficulties regarding interoperability with the other CDF Domains in a Study?**

(Please select all that apply)

☐ Non-matching Semantic Concepts and Definitions: One of the parties needs to update its semantic data;

☐ Superficial knowledge regarding Business Concepts (on both sides): Not enough information available;

☐ Rigid Definitions and Structures: Allows data exchanges but does not allow proper update of data semantics;

☐ Other (Please Specify):

9) **What are the most common problems when dealing with Tools and support software?**

(Please select all that apply)

☐ The Tools' inputs/outputs do not match the CDF design and data conversion needs to be performed;

☐ The Tools' terminology (units, parameters, operations) do not match the domain's;

☐ There are no direct interfaces between the tool and the domain parameters (manual actions need to be performed);

☐ The Tools have other great features that are not being used due to hard learning curve;

☐ Reuse of tools from previous studies is not performed due to data/semantics problems;

☐ Other (Please Specify):

10) **How are new needs and requests handled (e.g. a new functionality, a new service, a new data type, a new parameter)?**

(Please select all that apply)

☐ Changes are made on the Provider side to match the Customer requirements;

☐ Changes are made on the Customer side to match each Provider's rigid specifications;

☐ Negotiations are held between Customer and Provider to ensure the least impact on both sides;

☐ Other (Please Specify):

ANNEX B – ESA VALIDATION ON NEGOSEIO

1) Opinion about the framework, the ideas and the results:

1.1) Do you think the problems that were described are aligned with the current reality in ESA-CDF? Would the proposed framework provide improvement towards solving these problems?

1.2) Do you think negotiations would promote reuse and the establishment of best-practices for your CDF domain? Why?

1.3) Do you consider technologies like web-services and cloud computing to be a potential benefit for your work, or a trend that will fade in time?

1.4) How do you think the ideas about gathering knowledge about your domain would benefit your work? Do you think it would be possible to convert some of your expertise into a stable set of reusable services?

1.5) What would you consider to be the best contributions of this framework to your CDF work? And the major concerns about using it?

2) Applicability of the proposed framework to the ESA-CDF, viability to implement these ideas, and in which time frame:

2.1) Do you think the proposed ideas are suitable to be applied in the ESA-CDF environment?

2.2) Do you think the negotiation framework is viable to be tested in a small-scaled proof-of-concept in ESA?

2.3) Would find it interesting to participate in such proof-of-concept with your expertise and contribution?

2.4) When do you consider the conditions in ESA-CDF would permit the adoption of this kind of framework (in terms of time frame)?

3) Opinion about the documentation and information (if the ideas are well explained and are satisfactory):

3.1) Did you find the problems and ideas in the paper well explained and comprehensible?

3.2) What were the parts of the document you found easier to perceive?

3.3) What were the parts of the document you found more challenging for understanding? Do you think further explanation should be provided?