Pedro Miguel Negrão Maló

Mestre em Engenharia Informática

Hub-and-Spoke Interoperability: An out of the skies approach for large-scale data interoperability

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

Orientador: Prof. Doutor Adolfo Sanchez Steiger Garção, Professor Catedrático Jubilado, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Júri:

Presidente:	Prof. Doutor Paulo da Costa Luís da Fonseca Pinto
Arguente(s):	Prof. Doutor Manuel Martins Barata Prof. Doutor João Pedro Mendonça de Assunção da Silva
Vogais:	Prof. Doutor Ricardo Luís Rosa Jardim Gonçalves Doutor Peter Friess

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro 2013

Hub-and-Spoke Interoperability:
An out of the skies approach for large-scale data interoperability

Às mulheres da minha vida,

Diz o Miguel Esteves Cardoso que “Só quando os homens chegam a uma certa idade é que podem dizer com certeza que as mulheres são melhores do que eles em tudo”. E conclui: “Nos vintes, fica-se com a certeza. Nos trintas, aprende-se a disfarçar. Nos quarentas, ganha-se juízo e desiste-se”.

Eu que já ganhei juízo, digo que de facto as mulheres são de uma estirpe superior: mais sábia, mais emocional, mais inteligente, enfim assumamos, mais forte. As da minha vida então, são as melhores do mundo: únicas e fantásticas.

… únicas, desde logo porque que me aturam; é que ser-se aturado mete logo uma certa pena, e elas têm paciência para me aturar; porque (acredito) gostam de mim, mas também porque acham que sozinho nunca me governaria… pura verdade;

… fantásticas, porque sabem amar, porque são divertidas, porque são complicadas, porque sentem, porque se aguentam, porque se levantam, porque se emocionam, porque são unidas, porque são tão diferentes, porque se completam, porque tudo.

Eu aprendi a compreender estas mulheres minhas; compreendi não estar ao seu nível; todas têm uma força que eu nunca tive, nem nunca irei ter, mas que compreendo agora, não preciso ter. Elas completam-me e, dentro das minhas limitações aspergerianas, fazem de mim uma pessoa melhor.

É um privilégio té-las comigo; seguramente muito mais do que me terem com elas. Ao lado que qualquer uma delas, este meu trabalho é coisa menor. Até porque isto custou-lhes muito mais a elas do que a mim. Ainda assim, dedico-lhes este pouco de mim.

À minha família,
***Aos meus pais e meu irmão**,*

Por me terem tornado uma pessoa forte,
Por sempre me terem incentivado,
Por esperarem sempre o melhor de mim,

Por nunca me terem deixado desistir,
Ou optar pelo caminho mais fácil,
Por não me incentivarem a tomar atalhos,
Por serem exigentes e firmes,

Por serem pessoas rectas e honestas,
Por me terem ajudado a ser quem sou,
Por estarem sempre lá para mim,

Este trabalho dedico-vos também a vós.

Acknowledgements

The conclusion of my PhD thesis signals the finish of great walk on a long and winding road. And like all great journeys of life, I had the pleasure and honour of having the best of companions along the way. I would like to thank here to those who walked along with me and make my journey easier...

Thanks Professor Adolfo Steiger Garção! Thanks Professor Steiger for the great advisory and supervision of this work, which made it go in such a good direction. This acknowledging that I'm not the easiest person to be supervised. Thank you also for the good companionship, support and for believing that I would mature and get stronger.

Thanks Bruno Almeida! Thank you Bruno for all support, fresh ideas, bright discussions, great friendship, for all. Without your support this thesis would not have become what it is, or maybe would have never happened. Thank you so very much.

Thanks Márcio Mateus and Tiago Teixeira! Thank you Márcio for your great work, outstanding MatLab skills and for being so determined on doing things. Thank you Tiago for your great work, ability to respond positively to many different challenges and constructive criticism. Without both of you this would have been a Never-ending Story.

Thanks Ruben and Marco! Thank you Ruben for the great help and good talks all along the way. Thank you Marco for your good support and outstanding discussions. The friendship and companionship of both of two means a lot to me.

Thanks to Professor Luís Gomes! Thank you Professor Luís Gomes for being such a great mentor and model, especially in teaching, but also thanks you for looking out for me all the time. Thank you for caring.

Thanks Manuela Fernandes. Thank you Manuela for the validation of the mathematical representations presented throughout the thesis. The formalisation of Interoperability is one of the most important elements of this work and you helped to make it right.

Thanks to Sofia Coelho! Thank you Sofia for the English proofreading that was more than just that. Your valuable comments and suggestions much improved this document.

Thanks to all my other colleagues and co-workers at the research centre and at the university that somehow contributed to this work. Thank you Professor Ricardo Jardim Gonçalves for all the good support and colleagueship for all these years.

Thanks to those close partners from European research. Thank you Philippe Cousin for opening the Interoperability in the Internet-of-Things arena to me and for believing in my skills. Thank you Scott Hansen for being the best project coordinator ever and such a good friend. Thank you Harald Sundmaeker for having invited me to the CuteLoop project, which was an outstanding project and provided the good motivation for this research.

And thanks to all others who helped me, consciously, and also to those who helped me without knowing that they were doing it. Thanks to all of you that were there for me during the past years of my life, especially in those that coincided with this research.

To all, a big and sincere, thanks!

Abstract

Data Interoperability is a key challenge in large-scale heterogeneous environments. In here, interoperability via standards is not feasible or even possible; then, the classic approach, Point-to-Point (P2P) Interoperability, presents here two key problems: the trouble of non-modifiable systems that inhibit full possible interoperability and the excessive quantity of interoperability resources needed for establishing interoperability. A new approach is required for sustaining interoperability in those environments!

Laterally thinking, commercial air transportation environments exhibit similar properties and problems to Data Interoperability environments and therefore face comparable difficulties. Outstanding approaches such as scissor-hub operations and the hub-and-spoke paradigm have managed to address those challenges in commercial air transportation environments. Which, looking from data interoperability perspective, raises the idea of Mediated Interoperability and Interoperability Compositions.

From there, a novel approach for data interoperability is proposed, the Hub-and-Spoke (H&S) Interoperability, as the hypothesis for addressing data interoperability in large-scale environments. The H&S Interoperability approach fully solves the interoperability coverage problem and significantly reduces the number of resources needed for realising interoperability, thus outperforming P2P Interoperability. At the end, it is provided a technological realisation of the H&S approach, as the Plug'n'Interoperate solution, built upon plug-and-play principles applied to data interoperability.

Keywords: Interoperability; Hub-and-Spoke; Data Interoperability; Dynamic Interoperability; Sustainable Interoperability; Plug'n'Interoperate.

Resumo

A Interoperabilidade de Dados é um desafio chave em ambientes heterogéneos de grande dimensão. Nestes ambientes, a interoperabilidade não é exequível, às vezes nem sequer possível através de normas. A abordagem possível - Interoperabilidade Ponto-a-Ponto (P2P) - apresenta dois problemas chave: a condição dos sistemas não modificáveis que inibem uma completa interoperabilidade e a quantidade excessiva de recursos necessários para que se atinja a interoperabilidade. É então necessário uma nova abordagem para sustentar a interoperabilidade nestes ambientes! Olhando em redor, o transporte aéreo comercial apresenta propriedades e problemas similares àqueles dos ambientes de Interoperabilidade de Dados, enfrentando dificuldades semelhantes. Abordagens notáveis como operações *hub* em tesoura e o paradigma *hub-and-spoke* conseguiram lidar com esses desafios no transporte aéreo comercial. Observando estas abordagens da perspectiva da interoperabilidade de dados conduz à ideia de Interoperabilidade Mediada e das Composições de Interoperabilidade. Com base nestas ideias, uma nova abordagem para a interoperabilidade de dados é proposta - a Interoperabilidade *Hub-and-Spoke* (H&S), como hipótese para tratar a interoperabilidade de dados em ambientes heterógenos de grande dimensão. A abordagem H&S soluciona o problema da cobertura de interoperabilidade e reduz grandemente o número de recursos necessários para estabelecer interoperabilidade, superando assim a abordagem de Interoperabilidade P2P. No final, é apresentada uma realização tecnológica da H&S, a solução *Plug'n'Interoperate*, construída através da aplicação de princípios *plug-and-play* à interoperabilidade de dados.

Palavras-chave: Interoperabilidade; *Hub-and-Spoke*; Interoperabilidade de Dados; Interoperabilidade Dinâmica; Interoperabilidade Sustentável; *Plug'n'Interoperate*.

Table of Contents

Table of Figures

Table of Tables

Acronyms

ADC	Analogue-to-Digital Converter
API	Application Programming Interface
ARC	Australian Research Council
ASN.1	Abstract Syntax Notation One
BBS	Bulletin Board Systems
CAM	Contract Air Mail
COV	Interoperability Coverage
COV_{abs}	Interoperability Absolute Coverage
CRI	Centre for Intelligent Robotics
CRUD	Create, Read, Update and Delete
CSV	Character Separated Values
CTI	(ETSI) Centre for Testing and Interoperability
DEE	Department of Electrotechnical Engineering
DIY	Do-It-Yourself
DoD	Department of Defense

EC	European Commission
EIF	European Interoperability Framework for pan-European e-Government Services
ERA	Excellence in Research for Australia
ETSI	European Telecommunications Standards Institute
FCT	Fundação para a Ciência e Tecnologia
FCT-UNL	Faculdade de Ciências e Tecnologia of Universidade Nova Lisboa
FDDI	Fiber Distributed Data Interface
FET	Future and Emerging Technologies
FP6	6th Framework Programme for Research and Technological Development from the European Commission
FP7	7th Framework Programme for Research and Technological Development from the European Commission
H&S	Hub-and-Spoke
IA	Interoperability Artefact
IATA	International Airport Transport Association
ICT	Information and Communication Technologies
IEEE	Institute of Electrical and Electronics Engineers
IERC	European Research Cluster on the Internet of Things
IF	Interoperability Function
IFC	Industry Foundation Classes
IOL	InterOperability Laboratory
IoT	Internet-of-Things
IS	Interoperability Specification

ISG	Implementation Support Group
IST	Information Society Technologies
MDA	Model Driven Architecture
MDD	Model Driven Development
MDI	Model-Driven Interoperability
MNIS	Maximum Number of Interoperable Systems
MSc	Master of Science
NA	Number of Adaptors
NAT	Network Address Translation
NE	Number of Exporters
NEI	Number of Exporters and Importers
NI	Number of Importers
NIA	Number of Interoperability Artefacts
NIS	Number of Interoperable Systems
NR	Number of Readers
NW	Number of Writers
openBIM	Open Building Information Modelling
P2P	Point-to-Point
PhD	Doctor of Philosophy
PIM	Platform Independent Model
PnI	Plug’n’Interoperate
PnP	Plug’n’Play
RAD	Regulation for Professors’ Assessment

RAE	UK Research Assessment Exercise
RCC	Research Computing Center
RGSI	Research Group on System’s Integration
RTD	Research and Technology Development
SFDR	Spurious-Free Dynamic Range
SIP	STEP-based Integration Platform
SNDR	Signal-to-Noise and Distortion Ratio
SNR	Signal-to-Noise Ratio
STEP	Standard for The Exchange of Product data models
THD	Total Harmonic Distortion
UDI	User Defined Interface
e-GIF	(UK) e-Government Interoperability Framework
UKOLN	UK Office for Library Networking
UNH	University of New Hampshire
UNL	Universidade Nova de Lisboa
XML	eXtensible Markup Language

Preamble:
Origins and Crossroads

"You can't connect the dots looking forward;
you can only connect them looking backwards"
– Steve Jobs, entrepreneur, marketer and inventor, June 2005

The way I see it, I have been an engineer and a researcher nearly all my life! This, starting in my young days, while playing and experimenting with anything; all throughout my education, especially higher education; working in industry, namely at the Ford-Volkswagen AutoEuropa Automotive Manufacturing plant, doing Industrial Automation and Integration; and then coming back to academia to lecture and research – at the Faculdade de Ciência e Tecnologia of Universidade Nova de Lisboa (FCT-UNL) and the UNINOVA research institute. And it is now close to fifteen years that I've been performing research in Information & Communication Technologies (ICT), and yet, only recently did I start to do work that I consider genuine scientific research...

Take notice that this whole section is not mandatory reading of my thesis, but here you will understand the path, the meanders and shortcuts, the battles and deviations, the encounters and disappointments, etc. that finally led to my PhD research work. Here, looking backwards, I connect the dots of my work life making it clear that if you believe in what you are doing and in yourself, somehow things will eventually find a way to get on track. I don't really know how, but they do! And, if you care to know more about this, please keep reading.

Me– I started with a background of Computer Science with a touch of Computer Engineering, acquired through the Informatics Engineering degree and masters at FCT-UNL, which became more and more Computer Engineering as time went by. Once, during a lecture, I asked Professor Camarinha Matos: "What's an Engineer?" He promptly replied: "He/she is an individual (professional) who solves problems (difficulties, projects, etc.)!" I agree, but however add to his simple but wise definition, that he/she is an individual that solves problems with method, in the sense that he/she possesses the technique to go straightforward from problems to solutions; methods (and tools) are indeed the essences of Engineering. Academic studies and especially carrying out research and technology development provided me with the right methods and tools to address head on, the issues of computer systems and their applications.

However, long before, during my early years, I was already building and experimenting with things. I wanted to understand how stuff works by tearing things apart, altering them and then putting them back in order to see the results. For instance, as a late teen I spent lots of my spare time tampering with my two-stroke Honda NSR50R motorbike. So many times I disassembled its engine, modified the cylinder head and piston, tampered with the engine electronics, etc., just to experiment and get more performance. I changed things and tested in the garage (in-lab tests) and then even went to perform track (field) tests, all of which I recorded in paper notebooks. I had meagre resources back then, but I was able to improve the engine's performance by at least some 10% to 15%. Sometimes, naturally, the engine broke down which meant that I had to fix it, but even then there was something important to learn.

By that time, my enthusiasm for computers and electronics was growing rapidly. And as one might expect – at this young and restless age – I learned a lot and not for the best of reasons: phreaking and gaming! This was the time of the BBS (Bulletin Board Systems), Analogue Telephony and Modems, and we all wanted to download computer software (ok, mostly games, I admit) and these existed in BBS located above all United States! So, we had to find a solution to call the USA, for free of course or otherwise we would be grounded for life due to the telephone bills! And we did it; we actually developed a so-called blue box[1], which in essence was capable of generating

[1] Blue box. (n.d.). An electronic device used to bypass payment on a tone pulse telephone system. In Definitions.net. Retrieved on July, 2013, from http://www.definitions.net/definition/bluebox.

certain special tones (2600Hz+2400Hz) that gave us full control over switching into long distance dialling systems. We then used this technology to call any part of the world (especially the USA) for free and download software (ok, games). We even developed black boxes[2] that allowed our family and friends to call us without being charged. With this, I came to understand the principles of computer communications using modulation principles (modem), Hayes commands and much more.

Later on, and while attending the university, I had to learn a lot of computer systems and networks to play Doom in multi-player mode. Doom (and Ultimate Doom) is a science fiction horror-theme first-person shooter video game, which was THE game of those days. Now, to play multi-player we needed to create an in-house Ethernet network with coaxial cables between the three of us – no sweat. Problem was that we had a heterogeneous systems environment there: one Commodore Amiga 4000[3] running AmigaOS and two PCs with Linux 1.2.x and Microsoft Windows 95. Establishing a fully functional network was a nightmare considering the much different hardware (especially networking interfaces) and operating systems. So often we had to tweak the network – including patching the Linux kernel to get more overall stability and improved network performance – so as to engage in the most awesome and memorable battles. I still recall being so immersed in Doom that I literally moved in the chair to avoid opponents' shooting; likewise I remember so well that my cousin João Ventura felt dizzy when playing Doom for more than 15 minutes...

So, I consider myself both an engineer and a researcher, in the full sense, both by-nature and in my life. My motivation was always to investigate things and create solutions that could have practical usefulness. The practicality of the things I do has always been an essential element for doing good work and taking matters to an end. And, that is also why I've became involved, right from the start of my professional life, in applied research ICT projects. Applied research gives a purpose to the Research and Technology Development (RTD) that I do, and ultimately it does fulfil my nature.

[2] Black box (phreaking) was a small electronic circuit added to a telephone, which provided the caller with a free call. In Wikipedia, The Free Encyclopedia. Retrieved on July, 2013, from http://en.wikipedia.org/wiki/Black_box_(phreaking).

[3] The Amiga 4000 (1992) was the last desktop computer made by Commodore. A4000 was equipped with a 32-bit Motorola 68040 central processor at 25MHz and introduced the AGA graphics co-processor chips. Retrieved on July, 2013, from http://en.wikipedia.org/wiki/Amiga_1000.

Interoperability– I work in the Interoperability research field; I would say more specifically on the interoperability of data systems – Data Interoperability. Many times people ask me what Interoperability is and about the research I do in it. I must say that explaining Interoperability is already quite hard, let alone explaining research in Interoperability. Over time I have begun to better explain Interoperability by using the metaphor of the Tower of Babel from the Book of Genesis of the Bible [Genesis 11:1-9]. Me of all people, I'm not even a believer.

We all know the general outlines of the tale of the Tower of Babel: the epic journey of a united humanity, speaking one same language, which decided to build a tower to reach unto heaven. But, this was considered an outrage by God, Who came down to earth and decided to confound the speech of humans, scattering them all over the face of the earth. The tower project halted, as it failed the intention of humanity to be at the same level as God. The Tower of Babel is the (poetic) explanation of how humans spread around the globe and talk so many different languages. So, God created an Interoperability problem that, if not solved, would halt projects/systems. This came to be the best way I know to more easily explain the Interoperability scenario.

Thus, Interoperability deals with the circumstance of entities (people, objects, systems, etc.) that need to communicate and understand one another. Interoperability is a property that qualifies this ability, this capacity, of entities to "talk" to each other. The lack of interoperability may result in the degradation of communication performance, and this can ultimately lead to disastrous conditions. One should notice, however, that Interoperability relates not only with the ability of entities to exchange information, but also about their ability to understand the information that has been exchanged. So, let me explain now how Interoperability became the major subject of interest of my professional life up until now.

When you're young and restless, usually you team-up to work on projects with someone you identify with and look up to, thinking you can learn a lot and evolve. Also, you hope that there might be money involved too (e.g. from research projects) than can at least grant you subsistence and basic living conditions. That was also my case, when I joined Professor Adolfo Steiger Garção's research group to work on projects that were under his supervision. We were in the late 90s, early 2000, and I was yet to start my Masters research work.

Back then the research domain where we were working was called Systems Integration. Even the research group, part of the CRI (Centre for Intelligent Robotics) was called RGSI (Research Group on System's Integration). At that time we were working with product data models (especially those based on ISO10303 STEP – Standard for the Exchange of Product model data (Pratt, 2001) – and developing methods & technologies to support integration of product data industrial systems. Our main concept was that of SIP (STEP-based Integration Platform (Jardim-Gonçalves, Sousa, Pimentao, & Steiger-Garcao, 1999)) a technological platform aiming to ease the development of STEP-based systems and applications.

Our crown jewel, and a part of the SIP concept, was the GENESIS tool (Sousa, Pimentão, Jardim-Gonçalves, & Steiger-Garção, 1999) for STEP data binding, a technology that enabled the generation of an object-oriented early-binding "executable" representation of a STEP product data model. Very similar to today's widely used data binding technologies (McLaughlin, 2002), GENESIS was a breakthrough at the time, which provided developers with an easy-to-use library/facilitator/adaptor for the rapid development of STEP-based applications. I implemented much of the tool and we used a lot of GENESIS in projects up until 2003 (Maló, Jardim-Gonçalves, Borràs, & Steiger-Garção, 2002), (Jardim-Gonçalves, Tavares, Grilo, & Steiger-Garção, 2000), (Sousa, Jardim-Gonçalves, Pimentão, Pamiés-Teixeira, & Steiger-Garção, 1999), (Jardim-Gonçalves, Silva, Vital, Sousa, & Pamiés-Teixeira, 1997). This is where a lot of my experience of working with data, data models and its representations comes from, as well as my experience in designing and developing technological supports/facilitators for promoting data exchange.

It is only then that Interoperability settled in, motivated by a strong push from the European Commission (EC) around the general idea of Interoperability between organisations and the need for researching in Enterprise Interoperability. This rose all the way up to the debate of how to shape up FP6 – the 6th Framework Programme for Research and Technological Development from the European Commission – in respect to the Information Society Technologies (IST) thematic priority. This was when the European Commission funded a series of twenty-five roadmap projects to identify the research challenges in different domains, to assess Europe's competitive positioning and to derive strategic roadmaps for applied research to drive the development of some key areas to the next level.

Interoperability, namely Enterprise Interoperability, was one of the domains selected for road mapping via FP5-IST-2001-37368 ‘Interoperability Developments for Enterprise Application and Software’ (IDEAS) project (Doumeingts & Chen, 2003). We (I) participated in IDEAS. The IDEAS project provided extensive suggestions towards shaping up RTD around Interoperability in FP6, which then materialised in first call for proposals, strategic objective ‘2.3.1.9 Networked business and governments’, requesting “Technologies for interoperability supporting open networks of intelligent, autonomous, self-adaptive, self-configurable, and scalable software components for networked organisations” (European Commission, 2003). This gave birth to the two reference projects in Enterprise Interoperability research in Europe – ATHENA and INTEROP (Chen & Doumeingts, 2003). We will talk about them further ahead.

The beginning– A PhD is always a major endeavour in life and when you decide to go and do a PhD, you probably always start thinking that it will be the unique opportunity to learn and become expert (a notable one, preferably) in the selected research domain! So, naturally, when I started to move towards a PhD, my ambition was to take the time (initially anticipated not to be too long) to go deep into Interoperability and its the problems and especially to conceive new solutions. My ambition was to do some outstanding science that would have an application, some sort of use. I came to realise, like many others before me, that in the end you really just want to finish with it and move on with (personal and professional) life!

Like most people embarking on a PhD, I started off by trying to better understand what a PhD thesis is and what it comprises. I read some books on the topic, for instance (Eco, 1977), to get acquainted with the outline and writing process of a (doctoral) thesis. I learned many things in this process. Among them, that I could have a Preamble, to give the reader some insights about the road that led me to the PhD; also that I could have a Prologue, where I could introduce a motivational scenario that is usually included in the Introduction chapter but that generally creates confusion (to the reader, to the evaluator) that the thesis is solving that specific application problem; also managed to understand that the thesis outline should be simple and direct: (brief) Introduction → (sharp and to the point) State-of-the-Art → (comprehensively defined) Problem → (Background and lateral) Research → (ambitious) Hypothesis (following a novel scientific concept) → Evaluation (of the hypothesis) → Application (of the proposed concept) → Conclusions and Future Work.

Also, I knew the domain where I wanted to work: Interoperability. Interoperability (and Integrability) was the area of study that I had worked in all of my research life and naturally I wanted to capitalize on it at the same time as I could go forward and deeper in Interoperability problems and solutions. A PhD is such a long task – in duration and effort – that it needs to have a purpose other than academic degree. My view is that this purpose should be to mature and become a specialist in a research field while developing something really new and useful. And so, all I needed now was to find a research problem, a research problem in the realms of interoperability, one that justified scientific research to be conducted.

Even before engaging in the PhD, Professor Steiger and I had various brainstorming discussions on the problems and challenges of interoperability. Many times we ended up thinking that Interoperability is a hard domain in which to perform research when compared with others, for instance Electronics research. Electronics folks know exactly what they have to do: an analogue-to-digital converter (ADC) that advances state-of-art because it performs better, uses less power (electricity), occupies less die area, etc. The way I see it today, this was only partially true. Electronics researchers have one major issue sorted, and that's true: they know exactly the problem, which is to improve ADC performance (SNR, THD, SFDR, SNDR, etc.) in view of the existing constraints (thermal noise, power, area, etc.). And by knowing that, they have the validation (of the hypothesis) part also solved. Researchers in electronics can conduct several measurable tests and, by comparing with the specifications of state-of-art solutions, are able to promptly say that it is better (due to this or that).

Our difficulty, however, was that we didn't really know (in numbers, precisely) the interoperability problem. To some degree we knew the problems, yes; we managed to use "special lens" to look at things (environments, systems, scenarios) and look for the presence of the so-called "scientific essences" – non-linear dynamics, high-dimensionality, massive heterogeneity, hard to model, etc. – to reach interoperability problems of a complex nature. And we did find a few, I may say: how to manage the condition of interoperability interfaces (adaptors) changing in a non-controlled manner (e.g. due to the need to conform with new data format versions, due to new systems entering in the environment all the time, etc.) in an interoperability environment? How to interoperate many disparate (enterprise) systems altogether, which talk different data formats, but are required to exchange data anyway?

In some conversations, we even discussed ideas on how to approach some of these problems. For instance, we talked about using technological principles like the one used to plug disparate devices (e.g. printers, drives or any other peripherals) into computers – the so-called Plug'n'Play (PnP) approach – in which systems might include some kind of interoperability driver and in this way promptly plug and interoperate with others. We also discussed non-linear behaviours and their solutions by looking at control techniques, for instance used in active road traffic management, where artificial "moving bottlenecks" (slow vehicles that disrupt upstream traffic flow) are introduced in key sections of the road network for queue protection and to keep (linear) traffic flow from becoming non-linear.

But we did not manage to go far at that time, simply, I can now say, because I did not have the maturity (or intelligence, perhaps) to go from these fragments into substance. In the very same way that I was unable to get from a general idea of problems into a concrete view on what and where they were. So, I had certain sights on what certain interoperability problems might be, but I did not understand their exact nature and characteristics. And so, how could I improve a thing that I did not truly understand? And naturally, how could I validate something for which I had no measures?

As it happens, I was more a person of the (engineering) applications/solutions and less of the (scientific) problems, and this was preventing me from abstracting and generalising into concepts. And this was, as I see it, due to some "wrong" way and focus of my research life. Nonetheless, the talks with Professor Steiger were of great importance and the good ideas we reached would later conduct to the need for better understanding interoperability problems and help shape my new concept regarding data interoperability.

False start– I was told that a good way to do a PhD was to work on (European) collaborative research projects and try to find there some rough gem that I could then cut, polish, and turn into my PhD. This meant getting deeply involved in research projects, engaging the international research team and helping to fulfil the tasks of responsibility of our research group, and just hoping to find some scientific motive (i.e. a scientific research problem) along the way. So I did, and worked at that time in the ATHENA and INTEROP projects – more on ATHENA, less on INTEROP – in the quest for a scientific research problem.

The FP6-IST-507849 ATHENA Integrated Project (Ruggaber, 2006) aimed at being the most comprehensive and systematic European research initiative in IT to remove barriers to interoperability, to transfer and apply the research results in industrial sectors, and to foster a new-networked business culture. ATHENA was a huge project – in fact it was even considered a programme (of projects) – with renowned and well-experienced players that presented a good opportunity for learning and performing good research on Enterprise Interoperability.

In the end, and in the opinion of the project reviewers, ATHENA achieved some outstanding research results but failed to provide the anticipated impact. The reasons for this were, for sure, many and multifaceted: the novelty of the Integrated Project instrument, size of the consortium, etc. but also, in my personal perspective, major tensions between the need to perform outstanding research and develop new technologies, and that of complying with the requirements of the four application cases defined for validation/piloting. We will come to this briefly. Note: you can find a lot of the ATHENA results, well-organized and packed, at http://athena.modelbased.net.

The FP6-IST-508011 INTEROP Network-of-Excellence (Bourrières, 2006) was funded to provide the support (travels, logistics, coffees) for the Enterprise Interoperability community (especially academia and research organisations) to meet around Europe, so as to create the conditions for technological breakthroughs in Interoperability focusing on the three key thematic components: Semantics & Ontologies; Enterprise Modelling; Architectures & Technologies. The ambition of INTEROP was not to perform research, but rather to provide the conditions for people to debate towards new theory, methods, technologies and eventually projects in the Enterprise Interoperability research domain. The INTEROP network lives today via the INTEROP-VLab[4].

Indeed, the participation in ATHENA and INTEROP presented a very good opportunity for learning and taking part in large-dimension projects, but in the end it did not help me that much in shaping my PhD. Some important concepts and methods arose from these projects that are still today cornerstones for Interoperability Research, such as

[4] INTEROP-VLab is the "International Virtual Laboratory for Enterprise Interoperability", officially created as a non-profit organisation (http://www.interop-vlab.eu). INTEROP-VLab helps, develops and durably maintains the new European research community founded by the FP6-IST-508011 INTEROP Network-of-Excellence in its three and a half years of intense research in the domain of Enterprise Interoperability.

MDI – Model-Driven Interoperability (Elvesæter, Hahn, Berre, & Neple, 2006), etc. Unfortunately our research group was not that involved in the core RTD work of the ATHENA and INTEROP projects and, amidst the need to do a lot of work to fulfil our duties in the projects and the inexistence of good scientific focus, I did gain lots of experience but was not able to find that rare gem. This, alongside other problems, added up to a big feeling of failure, disillusion and discontentment.

The way I see it today, I was a bit (to say the least) undeveloped to embrace a doctoral undertaking. Doing a PhD requires a certain maturity that I did not possess back then. I did not have the ability to look at things and identify a problem that justified a good research. In fact, as I came to realise, for doing real science, most times you need to be able to see out-of-the-box and out-of-time in order to perceive a problem that might not present itself as such yet, but that will become one, in the future, when certain conditions are met (scale, heterogeneity, complexity, etc.). In reality, and despite the fact that I had the special lens on at all times, it happens that I was not sufficiently skilled to use them in projecting situations and circumstances to find problems that justified research efforts.

Interestingly enough, by the time I learned this ability – of looking out for problems that are not yet there –, I managed to understand many of the troubles with EU research projects that I have been participating in. It happens that typical cases/scenarios promoted by user partners are not challenging enough to justify the research and technological development novelty that the projects wish to uphold, forcing research partners to enrich (invent, in fact) the cases/scenarios so that the work (and associated funding) could be justified and results validated.

In the end, the research and developed technologies are outstanding, but projects fail to exploit most simply because the problem that they aim to solve is not there in the first place! I understood that it is so critical – already at the proposal preparation stage – to define and detail challenging cases/scenarios, which are promoted by visionary users (champions), as it is to meet the projects' research ambitions. The way I see it now, this is one of the key secrets that allowed us to be so successful in getting proposals awarded with funding in European and Portuguese competitive research programmes during the past few years.

And due to this all, doing a PhD out of an applied research project is very hard, if not impossible, to accomplish. Projects start from the principle that the scientific research is already there and the objective is then to put it into practice – usually to develop technology and integrate it with other partners' solutions – to create some outstanding applications (in the form of proof-of-concept/prototypes). Furthermore, projects have their own, demanding pace, which is not compatible with the going-back-and-forth, thinking-and-rethinking nature of true scientific work. Work in these projects is about technology research, development and integration rather than scientific undertakings.

Still, I see that you can do a PhD, one that aims at foundational breakthroughs in science and engineering, if done in the context of basic/fundamental research projects. Programmes for this include the Portuguese Fundação para a Ciência e a Tecnologia (FCT) on Scientific Research and Technological Development Projects or the European Future and Emerging Technologies (FET) research-funding scheme. This requires a very clear idea and especially a research plan in order to make sure that it is possible to be accomplished within the timeframe (and budget) of the project. But then again, this idea and planning is exactly what makes the core text of a project proposal and, therefore, it is required beforehand to make sure that funding is awarded.

Maturing– But it was not all complications and frustration back then. There were a lot of good creations, important learning processes and companionship that, I only came do realise later, have been extremely helpful. I promoted some internal technological projects, for supporting the research centre/department, that made me understand various and relevant facets of science. Also, I provided lots of guidance and assistance to colleagues and co-worker friends who valued my positive ideas and good support, which enabled me to grow and mature in respect to research.

Starting in 2006, and after identifying major difficulties of the centre/department administrative staff to collect, manage and explore data from research activities (publications, theses, projects, prizes, etc.), I began a quest to solve the problem at marginal costs. I looked into the open-source software market and found the interesting EPrints solution that was supported by the University of Southampton. EPrints is a platform for building institutional repositories of research information and promote open-access and open-data. And so, I installed, customised and maintained the centre/department research repository based on EPrints – the OA.uninova.pt.

The OA.uninova.pt repository enabled the administrative staff to perform much faster and with more accuracy than before on their duties of reporting and accounting for the research production of the centre/department. It also allowed management to have a much clearer and correct view of research figures and make better-informed decisions, e.g. in respect to scientific publication strategies. Personally, it allowed me to understand about the meanders of scientific assessment and publication strategies, which would eventually help me considerably. It also enabled me to known more about publisher's copyright policies and their views on self-archiving and open-access.

I dug into reference activities like the UK Research Assessment Exercise (RAE[5]) and the Excellence in Research for Australia (ERA[6]) initiative. The former establishes a well-structured methodology for research assessment based on information-submission following a well-defined data format; the latter defines a discrete classification system (A*, A, B, C) for ranking journals and conferences in terms of quality/relevance/notoriety/impact/etc. Both are the root of today's school-wide Regulation for Professors' Assessment (RAD) (FCT-UNL, 2012). This has enabled me to pursue a "publish less, publish better" culture, but one that is also scored higher. Also, I managed to learn a lot about publications' meta-data, open archives interoperability, open-access, self-archiving, science publisher copyright policies, etc.

As a follow-up of this work, and mandated by our management, I drafted a plan in June 2007 for the adoption/implantation/upgrade of OA.uninova.pt to the whole school (Maló & Steiger-Garção, 2007). A decision at the school-wide level never came and later the University started with a strategy for the whole of the university and its organisms (schools, institutes) for managing research information: to acquire the commercial CONVERIS product. I participated in the CONVERIS technical commission, and even wrote an opinion of approval (Maló & Steiger-Garção, 2007) but still think today that the EPrints solution would have suited the needs and resulted in a better trade-off between capabilities and cost.

---

[5] Research Assessment Exercise (RAE) is an exercise undertaken approximately every 5 years on behalf of the four UK higher education funding councils (HEFCE, SHEFC, HEFCW and DELNI) to evaluate the quality of research undertaken by British higher education institutions.

[6] Excellence in Research for Australia (ERA) is a research management initiative of the Australian Rudd Government being developed by the Australian Research Council (ARC).

Around 2006/2007, I helped my co-worker and friend Ruben Costa a lot on his research work and MSc dissertation. Ruben was working on a framework to support Interoperability for Collaborative Business Processes in e-Procurement (Costa, 2007). During this period, we had a lot of fruitful discussions about many things related to his task, the research problem and its formalisation, the work methodology, the solution, the testing & validation approach, etc. We also thoroughly debated how to shape the document and bring it together as a coherent whole. Sometimes, we had some really good "fights" about what to include (or not) in the dissertation. I still remember not being able to convince Ruben to withdraw a small section on Collaborative Relationships that talked about channel masters, Zen masters and chameleons!?!?

It was around this time that I (and Ruben) started to look, with much more detail, at the principles of the scientific method[7] as the ruling methodology for scientific research. We tried to do our best, but the customised methodology was far from perfect and far from adhering to the scientific process. It is now clear that I was "poisoned" precisely by the principles and methods used in the collaborative research projects that I was working on. The development approach used in the ICT collaborative research projects is the engineering (design) process (Haik & Shahin, 2003), comprised of: defining a problem, establishing the design criteria (e.g. as requirements), doing specifications, building technological prototypes, testing & validating (& possibly redesigning) the solutions, and finally presenting results (including pilots and papers).

Now, although the scientific method and the engineering (design) process seem to look alike, they have fundamental disparities! The major difference of the scientific method lies precisely in the need to formulate a hypothesis, i.e. a proposed explanation to a phenomenon, and the way to test and prove it via experimentation in order to achieve theorisation. It was from then that I started to understand these basic differences between science work and engineering work: a scientist tests hypotheses through experimentation, while an engineer designs a solution to a problem and analyses the results. Scientists perform experiments using the scientific method, whereas engineers follow the creativity-based engineering design process (Tayal, 2013).

---

[7] Scientific Method (noun) a method of procedure that has characterized natural science since the 17th century, consisting in systematic observation, measurement, and experiment, and the formulation, testing, and modification of hypotheses. In Oxford Dictionaries. Oxford University Press, n.d. Web. 10 August 2013. <http://oxforddictionaries.com/definition/english/scientific-method>.

Then, during 2007/2008, I committed myself to helping co-worker and friend Marco Delgado on his MSc research work and dissertation. Marco is a bright guy from Cape Verde who always knew what he wanted and pursued things with perseverance and competence. Marco was having some difficulties on progressing with his Master's dissertation when he approached me, looking for assistance. I helped Marco to follow on with his research that was wrapped around the outstanding work that he did in the FP6-502917 VIVACE project, where he performed as principal researcher, locally, for the group. Marco was working on a solution to allow the harmonisation of STEP and Model-Driven Architecture (MDA) conceptual Platform Independent Models (PIMs) (Delgado, 2008) to ease the STEP-based application development process.

Marco, and I had important talks and sharp exchanges about research in general and interoperability research in particular. We debated many aspects related to the uses of model-driven development – models, meta-models, modelling languages, model transformations – in interoperability methods and tools. We concluded that model-driven interoperability is a high-class technique but that not all about interoperability (and interoperability research) is of a model-based nature. Also, we debated a lot about how to write about testing & validation, use-cases, applications, pilots, etc. in the dissertation. We discussed that there could exist a section on applications (of the concept) that would detail how to make use of the proposed solution, and that was done in Marco's MSc dissertation. And this has also provided inspiration for me on my doctoral thesis document to include a section on the application/use of the proposed concept as to provide guidance on how to bring it to practice.

To say the truth, these, along with some other interests, cost me time and energy. But they paid back in so many ways in the medium-to-long term: they provided me with friendship and support for the good and particularly for the bad moments along the way; they gave me the essential skills and competences to manage research work, including my own; and, what is highly significant, they laid down the foundations of a loyal, resilient and committed team, giving a strong guarantee of a prosperous future. It is my honest and humble opinion that you can never achieve big things alone, both because it hard to do it on your own, but also because you don't have someone close to share it with. All this time and effort created the breeding environment for success that would pay off later on...

Teaching– I love to teach. I don't know if I was predestined to become a professor but it seems I have some natural talent to educate and motivate others. And apparently students also tend to agree with this, as I have a high curricular performance judging by the pedagogic surveys: 5.42 out of 6.00 in the first semester of 2011/2012 and 5.46 out of 6.00 in the first semester of 2012/2013. Additional results, for past semesters, are not available, as answers to questionnaires were considered non-representative from a statistical perspective; other results were not available as I write this document.

My teaching tutor/supervisor has always been Professor Luis Gomes of the Department of Electrotechnical Engineering (DEE) of the Faculdade de Ciência e Tecnologia of Universidade Nova de Lisboa (FCT-UNL). I started teaching – supporting laboratory sessions to be more precise – with Professor Luis Gomes and to this day I teach with him. Professor Gomes helped me tremendously on the best ways to perform in class by focusing on important matters: the (design and engineering) methods. He also helped me on building-up my character and progress academically.

Right from the very start of teaching, I always looked for how to improve through better techniques and the best ways to motivate students and improve knowledge transfer. For instance, it was Professor Luis Gomes who introduced me to Tomorrow's Professor Mailing List – an electronic newsletter distribution list managed by Professor Rick Reis of Stanford University – that provided (and still provides) excellent hints and ideas for progressing in my academic career, on teaching, on learning, on researching, on institutional involvement, and on so many other things. From then till now, with the help of others, my perseverance and the good advice and support of Professor Gomes, I managed to become a good professor, respected by both students and colleagues.

Today, I consider myself a seasoned professor who managed to improve the formula of good teaching: First, by making students understand, right from the start and in each and every lesson, that they can learn a lot from me and that the classes are highly valuable; and this is not only related to the strict teaching themes, but also, and especially, to advanced technological matters. Secondly, by not handing out solutions easy and/or readymade answers, but making students think, and think hard, as the mind is (and will always be) their biggest asset. Finally, by being provocative and not letting students settle down, ever; making them see that there is a long way ahead and that they should take the opportunity to absorb knowledge and experience from me.

Examples of such a strategy include, for instance, lively debates on CPU architectures – e.g. ARM vs. Intel, what ARM is and if can it make Intel die; mobile operating systems – e.g. iOS vs. Android vs. Windows Phone, and what is the best-of-breed; technological ecosystems – e.g. Apple vs. Others, and their features, strengths and weaknesses; technological innovation – e.g. Airbus vs. Boeing, looking at the future technologies of airplanes; and so on. This enhances student's culture about technology (as is essential for an engineer) at the same time as it improves their critical thinking and argumentation capabilities (soft-skills). This also forces professors to be up-to-date about technology, in order to conduct, participate in and mediate the debate.

Now, I teach digital and perceptional systems, that is, methods and tools for the design and engineering of digital and perceptional systems. So, I always felt that a big gap existed between what I was giving lectures about and the subject of my research – interoperability, focusing on Enterprise and Industrial Systems. I understood that 'Integration and Interoperability Infrastructures' was a recognised research subject at the department (FCT-UNL, 2001), but I also realised that Digital/Perceptional Systems was another (very different) research domain at DEE. This left me with a sense that something was missing as I always thought that academia was about performing good research and then taking that to (higher) education for teaching engineers-to-be state-of-art concepts, theories and technologies. So, activating the research-to-education link of the so-called knowledge triangle (Maassen & Stensaker, 2011) has been my quest.

U-Turn– One of my responsibilities in the group/department was that to look after the IT infrastructure – servers and information systems. This was because I had suitable competences for dealing with computer servers and networks, which came from those early days of "playing" with Linux and Windows. As time passed, and because there were many systems to maintain and ideas of others kept flourishing, I sought help, free help in fact. Free support is only possible from students; students who value the experience of having the opportunity and pleasure of putting their hands on advanced systems and computing hardware. So, I tried to recruit a student for such tasks and he became known to us all as "Puto dos Servidores"[8]!

---

[8] 'Puto dos Servidores' is an affectionate Portuguese nickname (something like 'The Computer Guy') given to the person (student) that acted as an information system technician or computer repair technician.

Now, finding enthusiastic people, with similar interests and motivation to you and especially with the same foolishness and hungriness to learn and evolve, is extremely hard. I had a few “Putos dos Servidores” who worked with me – some were ok, others somewhat strange – but one in particular had this rare mix: Bruno Almeida. Bruno was in his second-to-third year in university when I recruited him. By that time he had already flunked two years just to improve his skills of table soccer, which to me meant he was strongly determined and stubborn. At the same time, he built the software to run his father’s restaurant business from scratch, which added competence and fearlessness of seeing things through, to real-life operation. Finally, Bruno had that natural ability for dealing with technology and some outstanding skills of Linux and computer systems (both hardware and software). I thought I might be able to “raise” him and take him with me along the road towards success and a good life.

I put Bruno in a technology-rich environment and gave him the best conditions to evolve and progress. I tried to teach him the best I knew and could by tutoring him into science, research and technology, and in the Interoperability (and other) domains. This, alongside his innate capability to learn fast and his strong technical skills, made him evolve into a state where he could discuss and help me think about harder scientific and research issues. In fact, in some areas, Bruno managed to surpass me, which only made me happy and proud.

At that time he was doing his MSc thesis and I helped him a lot, like a supervisor. Both of us had a lot of good ideas but also bad concepts and wrong constructions about Interoperability, motivated by some dogma and by the failure to question things thoroughly. By the time he was ending his MSc thesis, we were already in a process of questioning many of the misconceptions and incorrect views we had about Interoperability and Systems.

The work that we did together on better understanding concepts and methods of Interoperability enabled us to know that some things on others’ works (even those considered of reference) were incorrect and/or incomplete. And we were taking these for granted and treating them as irrefutable, without giving them the proper scientific lookout. Lots of these ideas were the core on Bruno’s MSc thesis, but however we felt that the work we’d was good enough (even considering others’ works) and that we should call it quits and defend it as it was.

So it was without surprise that the work done – Bruno's MSc thesis on the integration of collaborative industrial environments through Model-Driven Visualisation (Almeida, 2009) – was accurately criticised on several fundamental misconceptions about Interoperability and Model-Driven Development. I thank Professor Ricardo J Machado, of the Universidade do Minho, for such important support to our wake-up call. Nonetheless, and as we had anticipated, Bruno ended up with a classification of 19 (out of 20) points on his MSc thesis!

After this, we got back to the whiteboard and began a new, fresh and in-depth outlook on Interoperability by throwing many of the concepts and the doctrine away. We looked at the problems that would really require Interoperability solutions and the existing practical methods and tools to address the problems. And while trying to understand the general purpose of the solutions, we tried to match them with the generic concepts of Interoperability that we knew about. We had fruitful discussions and reflections on the scientific intricacies of Interoperability, both from a scientific-technological perspective but also, and with strong criticism, from a practical/application view. I came to realise that it is much easier, faster and joyful to think about issues if you have someone who will listen and discuss with you, even when you could have reached them out by yourself.

A first insight was about Model-Driven Interoperability (MDI). Model-Driven Interoperability is the application of model-driven methods and techniques for solving interoperability problems (Panetto, 2007). Most of the current international research on (Enterprise) Interoperability follows the path of MDI. But as we came to realise, there is a lot more onto Interoperability without needing to consider Model Driven Development (MDD) principles; and as such, it is not mandatory (or even necessary) to consider model-driven approach in all Interoperability concept and solutions.

And from this, we went towards another important clarification on the reference forms of Interoperability: Integrated, Unified and Federated Interoperability (ISO14258, 1998). Most of us think that this is a general concept of Interoperability whereas this only applies to Model-Driven Interoperability (MDI). This is so clear in the document where it was first established but it became confused in Interoperability research domains where the three approaches are considered as Interoperability concepts per se. There is more to interoperability approach aside from those approaches of MDI.

Another important finding was that the most typical solution – and possibly the only one that really works today – for Interoperability of Enterprise/Industrial systems is the use of standards. Looking at industry examples on how to address interoperability – the buildingSMART[9] for Engineering & Construction industries, the funStep[10] for Furniture industries, or the Automotive Industry Action Group for Automotive industries, among others –, we saw that all of them follow the standardisation path – the ISO16739 IFC Data Model (Liebich & Wix, 1999), the ISO10303-236 funStep Data Model (Nuñez, et al., 2006), and the AIAG Joint Automotive Data Model (Chituc, Toscano, & Azevedo, 2008), respectively. As it is, the Interoperability using standards seems to suffice and work for industry; this I learned from my involvement as technical leader of the Portuguese forum of the Iberian chapter of the very successful buildingSMART interoperability initiative from 2004 to 2009.

However, the types of situation in which I was especially interested were those where standardisation was not feasible or possible, i.e. statistically speaking, those of the long tail (Anderson & Andersson, 2007). The kind of settings that I wanted to observe were those where there are lots of systems, where many of these systems want to interoperate with many other systems, where there are disparate data formats which are described in very different representations, where it is virtually impossible to reach out for agreements and where the promoters of such systems are too small to invest in interoperability, but where interoperability is a clear need.

So, I got to look back at environments and situations where interoperability is a clear requirements and where the interoperability problem would need a novel approach, and where the current solutions do not apply or are not fit: a real thing, with sufficient scale, enough diversity, ample heterogeneity, appropriate dynamics, i.e. complex, which could really motivate research in Interoperability!

---

[9] buildingSMART, formerly the International Alliance for Interoperability (IAI), is an international organisation which aims to improve the exchange of information between software applications used in the construction industry. It has developed and maintains the ISO 16739 Industry Foundation Classes (IFCs) as a neutral and open specification for Building Information Models (BIM). In Wikipedia, The Free Encyclopedia. Retrieved on July, 2013, from http://en.wikipedia.org/wiki/BuildingSMART.

[10] funStep is a community setup in the late 1990s with the support of the European Commission that, led by AIDIMA and UNINOVA, implements an European research strategy for better interoperability in the furniture sector. It has developed and maintains the ISO 10303-236 Application Protocol that specifies the use of ISO 10303 STEP integrated resources necessary for the scope and information requirements for furniture catalogue and interior design.

Inspiration– Luckily, and maybe not coincidentally, I was working at the time on the FP7-216420 CuteLoop project (Sundmaeker, Scholze, Stokic, & Faltus, 2008). CuteLoop was about 'networked devices enabled intelligence for realising proactive customers integration as drivers of integrated enterprise'. To put is simply, CuteLoop used networked embedded technologies and distributed/intelligent systems to promptly integrate actors within chains realised via communication and information services.

CuteLoop had two very different application scenarios but with similar technological needs: one about food chains, i.e. logistics of fresh produce (Reiche, Lehmann, Fritz, & Schiefer, 2011), and the other one related to maintenance support for construction craftsmen, the so-called Health-book-of-the-House (Sundmaeker & Kovacikova, 2010). The Health-book-of-the-House application is a very interesting case with tremendous exploitation potential; the logistics scenario was more interesting for Interoperability.

The logistics scenario was suitable and real! We managed to identify, possibly for the first time, a setting where the current solutions to Interoperability simply do not suffice; a scenario where we believe that the Interoperability problem is really present and needs to be tackled. The logistics scenario provided a setting with many systems/objects, which were required to interoperate with one another, and that use very different languages and formats for data. This provided us with the right setting for thinking about a new approach to data interoperability.

To this day, Bruno and I still believe that CuteLoop is the best-written proposal and well-structured project in which we got involved. The project concept was sound and innovative; key research themes perfectly aligned and presented in the proposal; work-plan impeccably structured (two-iterations development, specification and methodology per research theme, optimisation tasks at the end, etc.); and applications that were so different but yet so similar in technological needs and research challenges, and to which CuteLoop could respond. And all of this thanks to Harald Sundmaeker from ATB – the Institute for Applied Systems Technology Bremen – who put it all together.

I just wished that we had more skills at that time, and made CuteLoop an even greater project with far superior research and developed technologies. Unfortunately, at that stage, we did not yet have the necessary competences for such and we somewhat failed to perform and deliver – we have apologised for that and grown since then.

However, the food-chains scenario provided the motivation but not the whole inspiration for new forms to interoperate systems. The CuteLoop logistics food-chain scenario led me to look further into other similar environments. I started looking into the broader transport-and-logistics domain to seek interesting ideas and solutions that could help solve the Interoperability problem in a better way. I've looked into computer (data) networks that can be seen as the transport-and-logistics of electronic information allowing systems (computers, devices) to exchange data; but the environment has some fundamental differences in comparison with those of the Data Interoperability domain, and the solutions in place are not so interesting – we will see this later in more detail. And then, I started looking in depth into commercial air transportation.

Air transportation was (and is) a keen subject for me! I have been interested in air machines and air transportation systems since forever. I think that it all began – probably as it did with many others of my generation – due to the Top Gun[11] movie. We, from the so-called "Top Gun generation", used to dream night and day about the piloting and dogfighting abilities of Maverick and his friends. Ever since the movie premiered in Portugal, I started studying all about airplanes and began learning how to fly airplanes in computer games simulators. My favourite airplane was the General Dynamics F-16 Fighting Falcon, which was not part of the movie by that time; Top Gun was mainly about F-14s Tomcat and MiG-28s (fictional military aircraft). Even today I can say by head most of the lines of the Top Gun motion picture.

In fact, I wanted so badly to fly one of those marvellous air machines that, at eighteen, I decided to try and apply for the Portuguese Air Force Academy. The Portuguese Air Force had just bought the F16's for its squad, and so this was totally worthy to me. It was a whole week of recruitment back in March 1992; I recall the date simply because it was on the same exact week that the Lusitânia Expresso ferryboat arrived at the territorial waters of East Timor, being intercepted on March 11th by Indonesian warships. The trials took place during the week immediately after a tour of my high-school class to Lisbon, for an open-day visit at the Faculdade de Ciência e Tecnologia of the Universidade Nova de Lisboa. Just imagine the coincidence!

---

[11] Top Gun is a 1986 American action drama film featuring stars Tom Cruise, Kelly McGillis, Val Kilmer, Anthony Edwards and Tom Skerritt, whose plot revolves around naval aviation and the fortunes and misfortunes of naval aviators in the navy's Fighter Weapons School (aka Top Gun).

After two full days of psycho-technical tests, only I (qualifying for pilot) and other individual (qualifying for navigator) were able to get through. Three days of physical and medical exams followed, and I failed in the very last test that I did: I have scoliosis – a medical condition in which a person's spine is curved from side to side – and that is incompatible with military flying due to the possible need of ejection. I never came to be a pilot, but I learned lots about airplanes and air transportation.

Later, this enabled me to establish a fascinating parallel between (commercial) air transportation systems and data interoperability systems. Interestingly, there is a strong equivalence between them. It so happens that commercial air transportation connects (at the transportation level) all airports in a somewhat efficient way, just like data interoperability that also aims to connect (at the data level) all systems in an efficient way.

Also, in commercial air transportation, originating airports "export" given types of aircrafts that can only land in (or are be "imported" by) destination airports that are able to accept them. This relates to the exact same basic principle of data exporters/importers in data interoperability systems where data exported from one system can only be taken by a system with a compatible data importer. Furthermore, in data interoperability systems there are those systems that cannot be modified so as to import/export other data formats – but still need to be made interoperable within the environment; identically, in commercial air transportation there are those airports that cannot be modified to support all types of aircrafts – but which still need to be somehow connected to all other airports in the world.

Thus, I started to examine in detail and reflect about existing solutions from commercial air transportation systems, and to see how those could be of interest and applicable to interoperability. In this way, and by thinking laterally[12], looking at how to solve the Interoperability problems from other – not direct – angles, while leaving aside most of the views and streams to Interoperability research, my creativity was pushed towards a novel concept of (Data) Interoperability!

---

[12] Lateral thinking is the solving of problems by an indirect and creative approach, typically through viewing the problem in a new and unusual light (Oxford Dictionaries). Edward de Bono coined it back in 1967.

The Gem– Finally, by 2010, I had developed a new idea for data interoperability that solves the fundamental problems of state-of-play interoperability of data systems – the Hub-and-Spoke Interoperability approach. The hub-and-spoke distribution paradigm (or model, or network) is a system of connections arranged like a chariot wheel, in which all traffic moves along spokes connected to the hub at the centre. The hub-and-spoke paradigm is present in transport systems, logistics systems, telecommunications systems, web news media, etc. providing improved efficiency and performance. It is remarkable, and it even has a poetic touch, that the concept is inspired by a motorcycle/bicycle spoked wheel[13] invention.

The hub-and-spoke concept is used especially in commercial air transportation for efficiently connecting together airports around the world. It makes it possible to reach out for a full coverage of worldwide airports, considering that these cannot service all kinds of airplanes and cannot be modified to support them, which relates to the equivalent challenge of how to interoperate all systems – especially those that cannot be modified to support (import/export) alternate data formats – in data interoperability environments. And it also makes is possible to efficiently achieve interconnectivity between airports using the least possible resources (airplanes), which relates to the comparable challenge of how to interoperate (at the data level) the many systems making optimal use of resources required for accomplishing interoperability.

And then, I conceived a technological realisation of the hub-and-spoke Interoperability approach – the Plug'n'Interoperate. The Plug'n'Interoperate solution – PnI for short – is about adopting a Plug'n'Play mechanism to Interoperability. As it was, we got the inspiration for PnI from the Plug'n'Play concept in computer systems, where a device (a printer, a pen drive, a peripheral) brings with itself the driver (or can be retrieved from a repository in the web) that the target system can take to properly use the devices' services. That is, a concept of self-configuration that had already been introduced already in the 80s in Amiga computers – named Auto Configuration, auto-config or AutoConfig – intended to automatically assign resources to expansion devices without the need for jumper settings (Commodore-Amiga, Inc., 1991).

---

[13] A spoke is one of some number of rods radiating from the centre of a wheel (the hub where the axle connects), connecting the hub with the round traction surface. In Wikipedia, The Free Encyclopedia. Retrieved on July, 2013, from http://en.wikipedia.org/wiki/Spoke.

The Plug'n'Interoperate solution explores the same basic principle of Plug'n'Play / self-configuration so as to automate, as much as possible, the configuration and participation of systems into the Interoperability environment. The solution is made possible by the existence of 'interoperability drivers' which define translations between data formats, and that are taken by an interoperability support system to fully enable interoperability of systems in a data exchange environment. In the Plug'n'Interoperate environment, systems simply plug (into the interoperability support system) and promptly interoperate with other systems present in the data-sharing environment.

Given the initially considered scenario, the Plug'n'Interoperate solution foresaw an application area of reference: the Internet-of-Things. This was due to the scenario nature, and because the CuteLoop project and then also the ARTEMIS/FCT-100261 SIMPLE project (Georgouleas, Kalaboukas, Otero, & Maló, 2012) were performing RTD on technologies and applications for the Internet-of-Things. However, the Plug'n'Interoperate solution might apply to other application areas; it could be applied to any other domains of use, even and especially for Enterprise/Industrial Interoperability applications, obtaining improved performance while comparing with classical interoperability methods. In fact, even on small-scale interoperability scenarios, the Plug'n'Interoperate outperforms traditional interoperability approaches, namely P2P interoperability, especially in respect to the interoperability coverage.

Remarkably, this fact – of Plug'n'Interoperate being focused on the Internet-of-Things domains – made total sense. First, the data interoperability problem in the Internet-of-Things is a much more clear condition than in the enterprise systems, as there are lots of disparate systems (sensors and applications), made by many manufacturers worldwide, and which might therefore "speak" many different "languages" (data formats). Secondly, and in consequence, it enabled me to assume a strong position and influence in the IoT interoperability European research domains – especially in the IERC[14] Activity Chain 4 on IoT Interoperability – which meant more opportunities and projects. And finally, it succeeded in completing the connection with teaching, as it enables me to bring this good research to academia and teach students on advanced state-of-research solutions for perceptional and digital systems.

---

[14] IERC – European Research Cluster on the Internet of Things (www.internet-of-things-research.eu)

And, to reach the results of the thesis, it was required a profound study of the Interoperability problem and of the proposed solution in order to compare them out and validate hypothesis. This meant implementing the algorithms and performing many simulations of the Point-to-Point and Hub-and-Spoke Interoperability approaches. I decided to use the MathWorks® MatLab® tool for doing this, as it is simple to use and especially targeted for technical computing. Engineer Márcio Mateus – a Master student of mine, who worked on a method for measuring data transfer in heterogeneous IoT environments – helped me a lot with the algorithms' implementations, on the definition and execution of the many required simulations, on analysing the result data and on the creation of visualisations (charts) that explain the behaviour and show the properties of both interoperability approaches. Márcio was indeed my MatLab Guru for my PhD work[15].

Furthermore, doing the Plug'n'Interoperate required the definition of reference architecture for me (and others to follow) to implement the solution. Doing good technological architectural work is not a trivial task, as it requires good knowledge of architecting methods and an outstanding ability to inspect and reason about ICT systems; ICT architects are one of the most valuable (and well paid) assets in computer industry today. Luckily, Engineer Tiago Teixeira – another Master student of mine – managed to help me a lot on this work. Tiago did his Master's work by conceiving an architecture for organising interoperability information on highly dynamic and heterogeneous environments, as he is very skilled in technology architectures. Tiago acted indeed as the chief architect of the Plug'n'Interoperate solution.

And so, I, together with Bruno Almeida, Márcio Mateus and Tiago Teixeira, united as a purposeful team for accomplishing this thesis. Each of them helped me lots in their specific areas of competence and this made my work so much easier and for sure so much better. Moreover, they provided good companionship and encouragement for the long days – and especially nights and weekends – that it took to accomplish this work. I am forever grateful to each of them in particular and proud of us all, as a team.

[15] Remarkably, and knowing MatLab® for so long, it was only now that I (we) realised that the "Mat" in MatLab® stands for Matrices and not for Mathematics as I thought (and most still think). MatLab is about matrix representations and that was exactly how I (we) represented algorithms in the thesis. As Márcio wisely noted once: "In MatLab® everything is matrices, even a simple scalar is a matrix of one by one!"

Prologue:
Motivational Scenario

"We can improve (food chains) if we link, in a intelligent way,
the information stored locally by the different players
to give the right information, at the right time, to the right user"
– Dr. Kurt Jäger, Manager Euro PoolSystem, CuteLoop partner

Supply-Chain– The motivational scenario is framed within a supply chain scenario, or more specifically, in temperature-controlled chains, such as cold chains. In the cold chain, all starts at the producer, which can be seen as fixed actor in the chain. The producer can start a short supply-chain (Van der Ploeg, et al., 2000), which means that all intermediaries are bypassed, enabling the possibility of selling directly to the consumer and to devise own marketing and selling strategies. With this approach, shown in Figure 1, producers can increase their income and more importantly, certify that all products are sold to consumers in optimal conditions, due to the minimisation of environmental and transport conditions, e.g. as temperature, humidity, shock, etc.

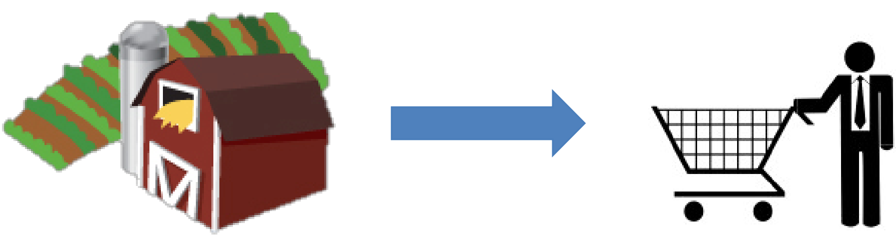


Figure 1: Short Supply Chain – Consumer gets products directly from Producer

Although, this way as some inconveniences, such as the fact that the market that the product is reaching is just local, i.e. to get this product the consumers need to go to where they are produced. And so, to increase the product's reach, the producers can arrange with a local transporting agent, to distribute it to the nearby stores.

This of course adds complexity to the supply chain, as it adds two more players, as it is depicted in Figure 2, the local transporting agency, which will pick up the product where it is produced and deliver it to the second added player, the retail stores where the consumers can acquire the product. With this new organisation, the supply chain can reach a broader market, by taking the product to further areas.

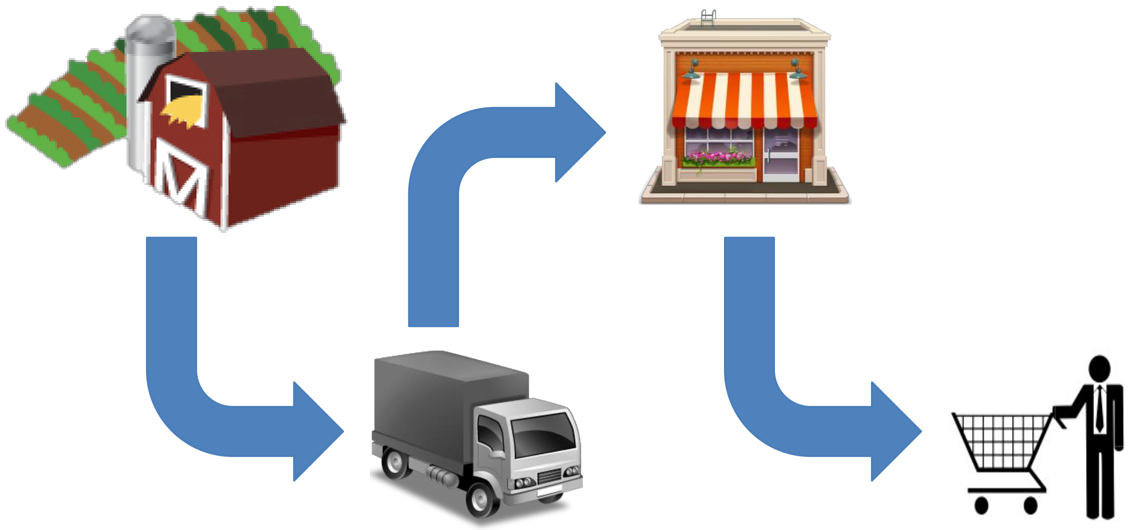


Figure 2: Enhanced Supply Chain - two more players between the producer and the consumer

One important aspect of the supply chain represented on Figure 2 is that it only represents one producer, which means that this supply chain is specific to that producer. Although, if it is considered a small demographic area (e.g. a town), there can be several producers, each one with its own supply-chain. Also in several numbers are the transporting agents, each one with different vehicles, and needing to be part of several supply chains.

This of course will increase the complexity of the system, because not only there is the need of a good organisation in order to ensure that the products arrives where it is needed, but also that it arrives in the best possible conditions to the consumer. The organisation present in Figure 3 represents the Town supply-chain scenario, where several producers are represented, using more than one local transporting agent to take the product from the place where it is produced, to deliver it to the retail stores where they can reach a higher number of consumers.

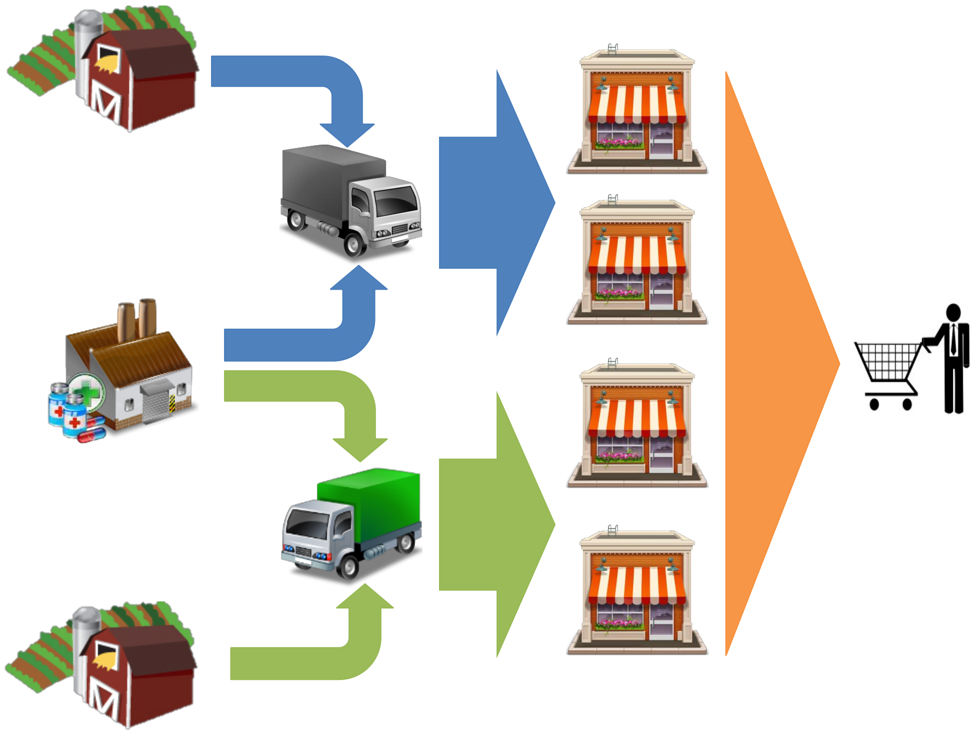


Figure 3: Town Supply Chain Scenario

As depicted in Figure 3, to reach consumers in further geographical areas, there is the need to use not only a variety of local transporting agents, in order not only to diversify the used agents, but also to reach different areas. However, to continue expanding the product's reach, there is the need to use bigger transporting agents, that will act not only as transporting agents, but also as traders, because they will pick up the product and deliver it to Distribution Centres, where the merchandise will be organised not by product, as it is in the previous exemplified cases, but by destination, i.e. the Distribution Centre will send several different products to the same retail stores, using the same transporting agent.

Figure 4 represents the Supply Chain using a Distribution Centre. The Supply Chain represented in Figure 4 demonstrates a simplified example of a real supply chain, because in a real example there are much more Producers, traders/transporting agents, Distribution Centres and Retail stores. This example can occur in a specific region, where there is a limited number of transporting agents, retail stores and

Distribution Centres, but if the supply chain expands to a country level, then there will be more of everything and the Supply Chain complexity will increase accordingly.

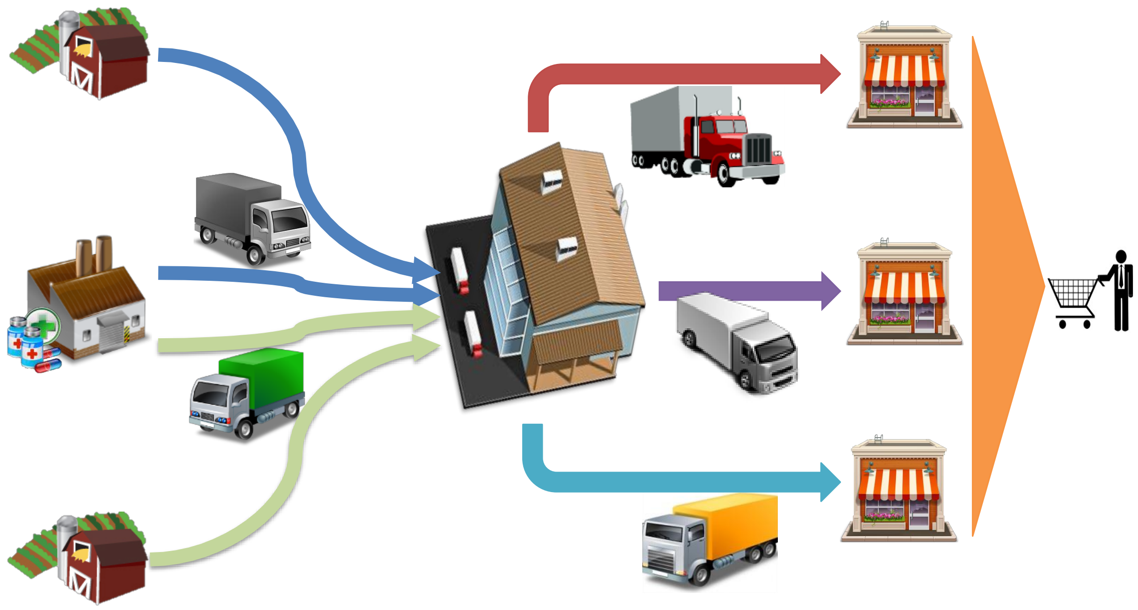


Figure 4: Supply Chain with Distribution Centre

Although, the Supply Chain will always be represented by the example depicted in Figure 5, which represents a generalisation showing two different paths from the product to the consumer. In one of them, representing a local area scenario, a local transporting agent will deliver the product directly to the retail store, and in the other one, representing a larger scenario, a distribution centre collects the products, organises them and sends them to the Retail Stores. These paths represent a real world approach, which is used in several areas.

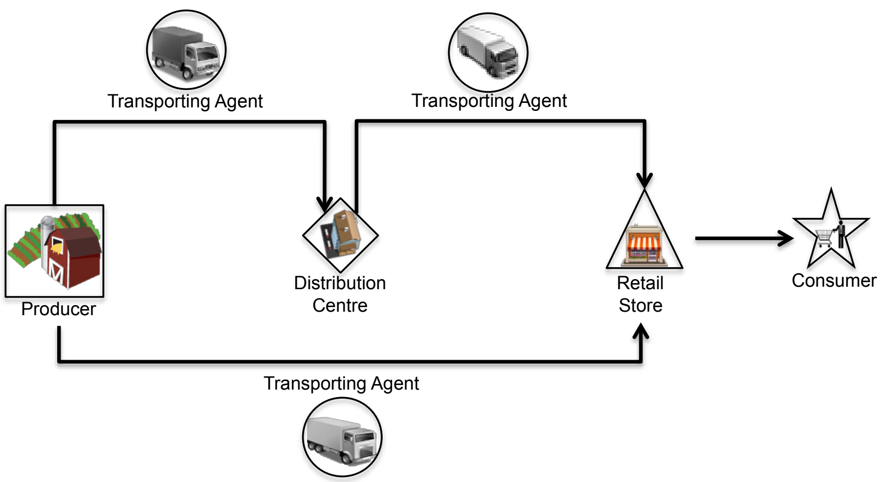


Figure 5: Generalised Supply-Chain

Although, the development of supply chains over the years has been slow. Companies developed individual parts of their supply chains starting with transportation and moving on to include warehousing, inventory, etc. (Lancioni, Smith, & Oliva, 2000). And so, companies also started to monitor the products flow within the supply chain, in order to minimize the costs, increase the flexibility in terms of delivery and minimize product losses and damages.

But, in order to allow an effective and efficient monitoring, there is the need to gather information in all supply chain stages, not only in fixed locations (e.g. Distribution Centres), but also when in transit. These supply-chain monitoring capabilities are based on collecting the required information during the route and provide it to the consumer, in order for the consumer to attest the conditions that the product was exposed. And, if each transporting company starts developing their own supply chain-monitoring system, there will be several different systems wanting to store all the information.

So, to provide reliable information to the consumer means that there is the need for companies to communicate amongst themselves so that the collected information may flow within each product's supply chain. Within each distribution centre it assumes especial importance, due to the fact that a distribution centre can receive products from several different transporting agents and with it, import all monitoring data collected during transport. While storing those products the distribution centre needs its own monitoring capabilities to collect data when the products are stored within the distribution centre warehouses. Afterwards, it needs to communicate to the transporting agents picking up the products all the supply chain information collected until then (both the received from the previous transporting agent as the collected in the warehouses).

However, the communication between Supply Chain parties is an actual problem, because each company may need to communicate with several others, from a variety of different supply chains. This number of communications can increase exponentially if we think of a global supply chain, where there are an infinity number of producers, transporting agents, Distribution Centres and Retail Stores. Each one of them having their own way to collect and handle the product monitoring-related information, and be able to communicate using only a limited (in not only one) number of ways. Another fact that contributes to the difficulty of information exchange between supply chain parties is the fact that most companies uses their own systems or uses other proprietary and unchangeable monitoring systems, hindering the way the exchange of information between heterogeneous systems that were not previously considered.

Figure 6: Example of potential Supply Chain companies (Google, 2013)

As each product as its own supply chain, each company may need to exchange information with an incredible vast number of other companies (in order to successfully monitor the supply chain products' flow) as the dots exemplified in Figure 6 demonstrate. Despite being an example collected from Google Maps, Figure 6 depicts several supply chain companies (producers, retailers and distribution centres) based in Central and Occidental Europe. The example aims to show, how many companies may need to communicate with each other, using their own way of collecting data, supply chain monitoring systems and their communication mechanisms, thus making communication between all parties in all supply chains very hard to accomplish.

1

Introduction: Interoperability, Methodology, Structure

"To make everything simpler in our life tomorrow in using any object, any information, anywhere, we need to solve complex interoperability issues today"
– Bernaghi Payam, Philippe Cousin, Pedro Maló, Martin Serrano, César Viho
IERC Cluster Book 2013 & IERC AC4 Interoperability Manifesto

To conduct or to guide research in Interoperability, one should first establish the basic ground of Interoperability. First, a definition of Interoperability must be established and agreed upon the author and the reader. Also, one must overlook the concerns of Interoperability and establish a focus on those of interest for the thesis. Next, one must check the approaches to Interoperability and see how these relate with work ahead. Lastly, the settings/boundaries that apply to this research must be detailed.

A proper research needs a sound and clear research methodology. The research methodology establishes the way to systematically solve the research problem. This section details the research methodology that has been followed in this thesis, to go from the research problem to a proper and validated solution.

Finally, a thesis is to be provided in the form of a well-organised document that reports on the (good) research that has been performed. In this chapter, the thesis organisation rationale is presented, through a short yet comprehensive description of each chapter, and how it contributes to the coherent whole of the document.

1.1 Interoperability

Interoperability definition– There are countless definitions of Interoperability that have been put forward by Interoperability practitioners, researchers, scientific and industrial associations, standardisation bodies, governmental agencies, militaries, etc. A comprehensive study by (Ford, Colombi, Graham, & Jacques, 2007), covering the 1997-2006 timeframe, identified some thirty-four different definitions of Interoperability used in research papers, standards and other government documents. Since then, even more authors have put forth new definitions of Interoperability. Some of these definitions are very different in scope, some relate specifically to a given application domain, but they all show the richness of the interoperability field of study.

In the scope of this thesis, the simple yet mature definition of Interoperability provided in the IEEE Standard Glossary of Software Engineering Terminology, suffices:

> *Interoperability. The ability of two or more systems or components to exchange information and use the information that has been exchanged* (IEEE, 1990)

Interoperability deals with the circumstance of entities – systems, components, objects – that need to communicate and understand one another (possibly to accomplish some common goal). Interoperability is thus a property that qualifies the ability of entities to “talk” and “comprehend" each other. It is important to note that Interoperability is not only about the ability to exchange information but also about the ability to understand the information that has been exchanged.

Interoperability scope– The premier civil research facility on Technical Interoperability is the InterOperability Laboratory (IOL) at the University of New Hampshire (UNH), founded in 1988. The UNH-IOL developed as a branch of the University's Research Computing Center (RCC), the group responsible for supporting the computing and networking needs of research groups at UNH. Back then, the RCC was testing Fiber Distributed Data Interface (FDDI) equipment for deployment in their network and found that equipment from two competing vendors did not interoperate. After debate, vendors came together to solve the problem, which stemmed from differences between the draft version of FDDI specification that one vendor used and the final version that the other adopted (UNH-IOL, 2013). At that time, focus on Interoperability was on technical issues related to networking equipment.

From then on, and all throughout the 90s, interest in Interoperability grew. Back then, Interoperability was still considered a pure technical problem linked to interoperability issues of computer networks and communication protocols. Gradually, the view on Interoperability evolved as more aspects were considered and new application domains were encompassed. And so, Interoperability became wider in scope to include new dimensions to broadly address interoperability problems; at the same time, Interoperability specialised more to focus on the specifics of a domain of application.

During the EU FP6-IST research programme (2000-onwards), Interoperability in the context of the Enterprise (Enterprise Interoperability) was considered along three axes: Architectures & Platforms, focused on defining a technical framework for interoperability; Semantics, addressing the semantic aspects of interoperability; and Enterprise Modelling, to model interoperable inter-networked organisations. This was first established in the FP5-IST-2001-37368 IDEAS project and later developed in the FP6-IST-507849 ATHENA Integrated Project and on the FP6-IST-508011 INTEROP Network-of-Excellence. They considered Interoperability a stack of ICT Systems (Data and Applications), Knowledge and Business, and with Semantics cutting across them.

Still around 2004, the EIF (European Interoperability Framework for pan-European e-Government Services) identified three aspects of interoperability: Organisational, Semantic and Technical. The organisational aspects concern defining business goals, modelling business processes and bringing about collaboration of administrations that wish to exchange information; the semantic aspects have to do with ensuring that the precise meaning of exchanged information is understandable; and the technical aspects cover the technical issues of linking up computer systems and services.

Around 2004-2005, the UK e-GIF (UK e-Government Interoperability Framework), that focused on establishing the technical policies and specifications governing information flows across the government and the public sector, talked about four orthogonal dimensions to Interoperability: Content Management, Access, Data Integration and Inter-connectivity (UK Cabinet Office, 2004). These dimensions of Interoperability were especially defined to address the specifics of the application domain, enabling the seamless flow of information across government and public service organisations. Thus, systems' interconnectivity, data integration, e-services access and content management specifications were now considered.

However, much earlier, the militaries had already started to look at Interoperability. (Ford, Colombi, Graham, & Jacques, 2007) state that the oldest definition of interoperability – dating back to 1977 – is given in the DoD (Department of Defense) Dictionary of Military and Associated Terms. By then, Interoperability was examined at four levels: Strategic, centred around harmonising world views, strategies, doctrines, and force structures of the US and its allies; Operational, related to force planning and battle management; Tactical, for interoperating forces within tactical operations; and Technological, focused on automated tools and compatible/secure communications.

Following all this, other initiatives and programmes looked at Interoperability with broader concerns (see Figure 7). For instance, the UKOLN (UK Office for Library Networking) Interoperability Focus distinguished six dimensions of Interoperability: Technical, Semantic, Political/Human, Inter-community, Legal and International (UKOLN, 2006). There was also Interoperability in e-Health (ICT technologies for the Health domain), which considered not merely a technical issue, but also legal, ethical, cultural, economic, social, medical, organisational and semantic aspects (EC, 2006).

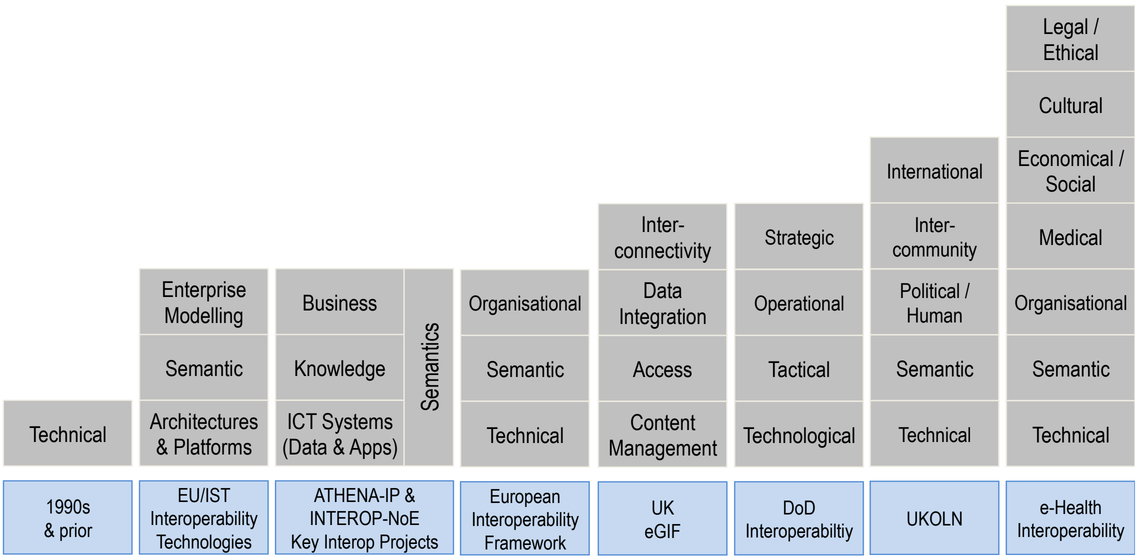


Figure 7: Enlarging Scope of Interoperability

These days, Interoperability is a key issue in collaborative environments so as to enable effective collaboration between systems and it covers a multiplicity of complementary, orthogonal and inter-dependent dimensions. Moreover, Interoperability is studied in a wide range of domains (of application), and is even observed in many different concerns, depending on the application domain. Interoperability practitioners and researchers thus have many areas to focus their studies and work on.

Interoperability focus– Interoperability can be looked at from many different perspectives and considering various concerns. Due to the fundamentally technological focus of this work, let us consider the reference levels of Interoperability as defined by the European Telecommunications Standards Institute (ETSI). ETSI distinguishes four different levels of interoperability: technical interoperability, syntactical interoperability, semantic interoperability and organisational interoperability, as shown in Figure 8.

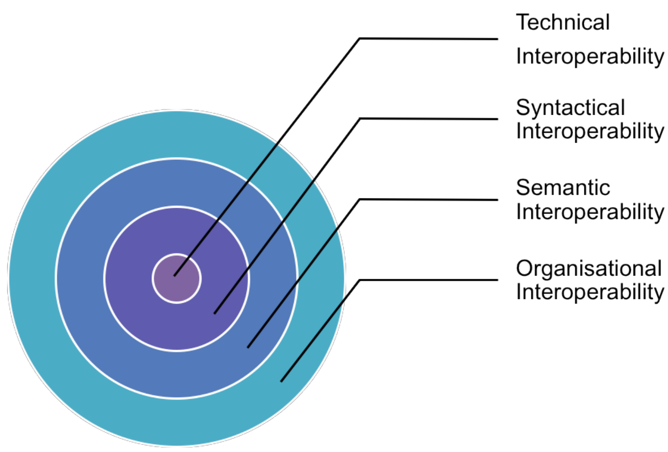


Figure 8: Different Levels of Interoperability, adapted from (ETSI, 2008) by the PROBE-IT project in support to Activity Chain 4 on Internet-of-Things Interoperability

According to ETSI, "*Technical Interoperability is usually associated with hardware/software components, systems and platforms that enable machine-to-machine communication to take place. This kind of interoperability is often centred on (communication) protocols and the infrastructure needed for those protocols to operate*" (ETSI, 2008). As it is, 'things' (components, systems, platforms, devices) are required to intercommunicate – i.e. to be technically interoperable – with each other before being able to exchange any data whatsoever. Technical interoperability qualifies the ability of 'things' to establish effective communication channels that makes it possible for data to be exchanged.

Then, *"Syntactical Interoperability is usually associated with data formats"* (ETSI, 2008), more specifically with the way that data is structured. As it happens, messages that are transferred by communication protocols still need to have a properly defined structure (syntax and encoding); commonly used data transfer syntaxes include CSV[16],

[16] CSV (Character Separated Values) is a (file) format that is used to store data in a structured table of lists that are separated of each other by a given Character (usually a comma or semi-comma).

STEP Part 21[17], ASN.1[18] and XML Documents[19]. Two ‘things’ are said to be syntactically interoperable, if one provides data according to a structure that the other can handle, i.e. can if interpret/read. Syntactical interoperability is a necessary condition for further interoperability (especially for semantic interoperability).

“Semantic Interoperability is usually associated with the meaning of content and concerns the human rather than machine interpretation of the content” (ETSI, 2008). Semantic interoperability means that there is an understanding between communicating ‘things’ on the meaning of the content (information) being exchanged. *“Beyond the ability of two or more (...) systems to exchange information, semantic interoperability is the ability to automatically interpret the information exchanged meaningfully and accurately”* (Wikipedia, 2012).

Together, syntactic interoperability and semantic interoperability designate a broader concept of interoperability: data interoperability. That is, both interoperability dimensions relate with the condition of systems being interoperable at the data level, i.e. being able to interpret/read the structure of data and understand the meaning of data elements. Data interoperability then certifies that systems are able to exchange data between themselves in a way that they can read and understand each other.

One should note, however, that systems could be semantically interoperable but not syntactically interoperable, or vice-versa. It might happen that one same semantic representation might refer to two very different syntaxes (data formats). For instance, one can represent the same (information) table by using the CSV[16] data format or by using an XML[19] representation; both represent the same semantics – data arranged by rows and columns – but enclose very different syntactical representations. On the other side, one given data format (e.g. XML[19]) might be used to represent completely different concepts, i.e. semantics.

---

[17] ISO 10303 (STEP) Part 21 “STEP-File Clear text encoding of the exchange structure” defines the encoding mechanism on how to represent data according to a given STEP EXPRESS schema.

[18] ASN.1 (Abstract Syntax Notation One) is a standard and notation that describes rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking.

[19] XML document relates to a data object constructed according to well-defined XML syntactic constructs.

The focus of this thesis is put exactly on data interoperability. The research work presented here looks especially at the issue of interoperating systems at the data level. However, the proposed theoretical and practical framework might also apply to technical interoperability (if needed); likewise, the proposed ideas and their solutions can also be considered for Organisational Interoperability, but however, the latter is not a primary concern in the scope of this thesis.

Interoperability approaches– Generically speaking, there are two main approaches to Interoperability: (1) Standards-based Interoperability and (2) Interoperability without Standards. Interoperability via standards means that parties adhere to common agreements/specifications and this provides a baseline for achieving interoperability. Interoperability without using standards is those approaches to interoperability that consider the use of standards not required or simply not possible.

A standard is a *"document established by consensus and approved by a recognized body that provides for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context"* (ISO/IEC, 2004). And focusing on technical domains, a technical standard is *"an established norm or requirement in regard to technical systems, (…) usually a formal document that establishes uniform engineering or technical criteria, methods, processes and practices"* (Wikipedia, 2013).

For instance, the technical level of interoperability (as defined by the ETSI model of Interoperability) is largely associated with a standards-based approach to interoperability. That is, the technical interoperability problem is mostly solved through the adherence to communication standards, that are well defined and maintained by very clear organisations worldwide (ETSI, ITU, IETF, etc.), with a proper dimension and the involvement of the industry; this, aside from experimental or niche protocols that do not present sufficient relevance or scale to be considered for technical interoperability (at least, today). As noted by (Kubicek & Cimander, 2011), *"ever since TCP/IP became available and widely accepted, technical interoperability no longer presented any relevant barrier to interoperation"*. This is so true that there are even all-inclusive solutions that act as interoperability enablers for multi-protocol and multi-systems integration at a technical level: Universal Gateways.

This makes sense, as networking supports usually come bundled at the hardware/firmware level, among other reasons because they are so linked to the medium of transport (e.g. type of radio in wireless communication). Therefore, it is natural that network systems manufacturers/developers make sure that their 'things' support, out of the box, a (few) set(s) of networking protocols and associated medium of transport. As such, it is logical that there be only a small set of communication protocols in the world of networked systems – few standards communication protocols considering operational restrictions and transport modes –, which enable disparate 'things' to internetwork.

Now, even standards may not be interoperable (ETSI, 2008). People that usually have very different personal and professional backgrounds define standards and it is not always easy to reach out a full understanding and agreement. Furthermore, it is often the case that there are not enough resources, or time, to turn all contributions into a coherent whole. Typical consequences of this include: incomplete specifications that miss essential aspects for achieving full interoperability; inadequate interfaces lacking clear definitions, thus hindering interoperability; poor handling of options, including inconsistencies and contradictions that lead to non-interoperability; lack of clarity, making it difficult to implement the standard, creating interoperability problems; etc.

An outstanding example of such difficulty is, for instance, the need for Implementer Agreements in the widely adopted ISO/IS 16739 Industry Foundation Classes (IFC) data model standard. The IFC data model is an open, neutral and standardised specification for Open Building Information Modelling (openBIM) in the AECO[20] industries promoted by buildingSMART[21]. In some cases the IFC specification offers alternatives for support by software applications and implementers are defining agreements to implement the IFC standard in a coherent and thus interoperable way. Once finalised and accepted (by the ISG – Implementation Support Group), an Implementer Agreement becomes a binding agreement for implementing the particular IFC view to which the implementer agreement applies (Liebich, 2010).

---

[20] AECO – Architecture, Engineering, Construction and Operations

[21] buildingSMART (formerly the International Alliance for Interoperability, IAI) is a neutral, international and unique non-profit organisation which aims to improve the exchange of information between software applications used in the AECO industries. buildingSMART develops and maintain international standards for openBIM such as: ISO/IS 16739 IFC data model, ISO 2948 IDM or the ISO 12006-3 based IFD Library.

Then there are interoperability approaches that consider standardisation not needed. Not needed, as for instance the number of (different) systems to interoperate is relatively small and it might be possible to straightforwardly interoperate systems altogether. In such cases, it is possible to promptly interoperate systems by using approaches such as point-to-point interoperability (Morris, Levine, Meyers, Place, & Plakosh, 2004). Point-to-point (P2P) interoperability is characterised by direct system-to-system connections that establish interoperations between systems. See Figure 9.

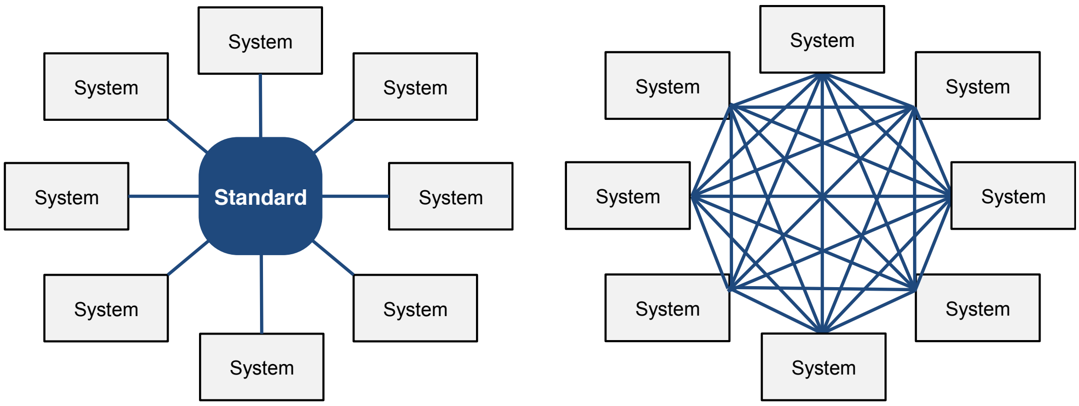


Figure 9: Illustration of Standards-based Interoperability (left) and P2P Interoperability (right)

And then, there are those approaches that consider standards-based interoperability not possible at all. Not possible, because the number of specifications in the environment, that need to be harmonised (e.g. into a standard) makes standardisation efforts prohibitive; not possible, as adherence to standards in a global environment is not guaranteed; not possible, due to fact that establishing point-to-point interoperations between many different systems in the environments is costly and/or hard to be made; etc. In such conditions, alternate interoperability approaches, aside from standards-based interoperability and point-to-point interoperability, are required.

In such cases, dynamic interoperability (IERC, 2013) approaches are necessary for systems to be made interoperable on demand and as needed. Dynamic interoperability is especially focused on making interoperability sustainable in complex – highly heterogeneous, dynamic and large-scale – environments. Sustainable Interoperability is defined as the "*aim of improving the quality of service by contributing to a more robust interoperability avoiding excessive consumption of resources (e.g. man-power and time) (…)*" (Jardim-Goncalves, Popplewell, & Grilo, 2012). Sustainable interoperability explores new theories, concepts, methods and tools to sustain interoperability of networked systems in complex interoperability environments.

An excellent example of a dynamic approach to Interoperability is that of the FP7-231167 CONNECT research project, aimed at making networked systems eternally connected. CONNECT focused on dropping the interoperability barrier by adopting a revolutionary approach to the seamless networking of digital systems, by synthesising on-the-fly the connectors via which networked systems communicate (CONNECT, 2009). In this sense, the project introduced the concept of the 'emergent middleware'[22] (and its enablers) so as to sustain interoperability in the increasingly connected digital world. Emergent middleware's are effectively synthesised according to the behavioural semantics of application-down to middleware-layer protocols run by the interacting parties (Issarny & Bennaceur, 2013).

This thesis concentrates exactly on those approaches that do not consider standards as the way to interoperability and that go past the limitations of point-to-point interoperability; that is, <u>thesis focuses on dynamic interoperability approaches in support of sustainable interoperability</u>. Particularly, this research targets dynamic approaches to data interoperability, i.e., novel theories and methods for dynamically interoperating systems at the data level in such a way that improve performance and competence of state-of-research / state-of-practice on data interoperability approaches.

Finally, it is worth mentioning that interoperability (irrespective of the approach used) can be greatly improved through supporting activities, particularly <u>Interoperability Testing</u>. Interoperability Testing is about making sure, via testing and validation, that interoperability is working properly, as anticipated. Testing will not eliminate all possible causes of non-interoperability but it can help a lot on improving interoperability. Thesis does not focus on Interoperability Testing but recognises it as of major importance.

An outstanding facility for Interoperability Testing is the ETSI Centre for Testing and Interoperability (CTI). ETSI-CTI principal task is to plan and develop conformance and interoperability test specifications related to ETSI standards. CTI also promotes the leading ETSI Plugtests™ events that make it possible for developers from different (even competing) organisations to come together and test/check their own implementations as to ensure proper interoperability between products.

---

[22] Emergent middleware is a dynamically generated distributed system infrastructure for the current operating environment/context, which allows functionally compatible systems to interoperate seamlessly.

IoT Data Interoperability– Perhaps the finest case where standardisation approaches are not feasible is data interoperability in the Internet-of-Things (IoT). The IoT is the idea of a dynamic global network infrastructure where physical and virtual 'things' communicate and share information with each other. The IoT is and will continue to be a heterogeneous, multi-vendor, multi-service and largely distributed environment; these are good ingredients for interoperability problems. *"It is not only the technical complexity that impacts significantly on drawing concise and coherent development lines, it is also the factor of exponential connectivity and thus complexity increase which leads to unnecessary sub-optimal solutions and spaces of chaos"* (Friess, 2012).

In fact, the IoT can be seen as both the first and the final frontier for data interoperability. First frontier, because it is the initial mile of data provided by sensing systems and where interoperability would enable things to talk and collaborate altogether for a higher purpose and provide their data to applications; and final frontier, because it is possibly the place where interoperability is more difficult to tackle due to the unavoidable complexities of the IoT that makes dealing with interoperability a hard task. Thus, addressing (data) interoperability in the IoT is as mandatory as it is a hard!

The IoT is highly heterogeneous environment; the IoT is composed of a vast number of 'things' (devices, sensors, smart objects, etc.) that are conceived by a lot of manufacturers and are designed for many different purposes and target diverse application domains. Each manufacturer/vendor presents 'things' data using a convenient data syntax and semantics, most times in a proprietary format, and it is the role of the application (software) developer to make sure that data is interpreted and understood. Considering the heterogeneous nature of the IoT, it is a hard problem for developers to extract data from many disparate data sources in the environment; i.e. it is difficult for many applications to interoperate with many sources in the IoT.

And, more and more, the IoT is a dynamic and continuous evolving environment; the Internet-of-Things is an environment where new 'things' (that were not even considered at the start) are entering (or leaving, or changing) all the time, 'things' that might support new unforeseen data formats (e.g. complete new data formats, new versions of existing data formats, etc.) but still need to communicate and share data with others in the environment. As such, it is hard to make sure that 'things' are kept interoperable in a constantly changing and highly dynamic environment like the Internet-of-Things.

And, the IoT tends to be high dimensional; in the IoT many ‘things’ co-exist (devices, sensors, smart objects) that need to communicate and exchange information. And with the evolution of the Internet-of-Things, more and more ‘things’ wish to connect (interoperate) to each other. Now, high dimensionality would not be a problem on its own, if it wasn’t adding to the high heterogeneity and dynamicity of the IoT; the huge scale of the Internet-of-Things of the future leads to even more heterogeneity and rapidly changing environments, which complicates data interoperability even more.

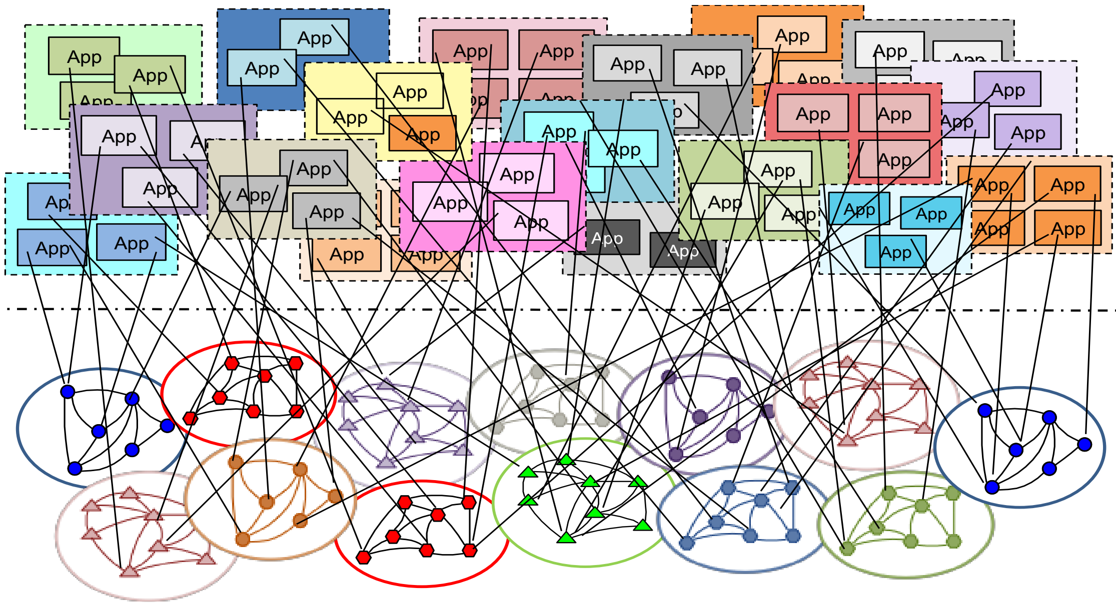


Figure 10: Illustration of the Internet-of-Things Complexity

In this way, the IoT can be a highly heterogeneous, dynamically changing, large-scale data environment, qualifying it as being of complex nature. Being able to deal with the increasing complexities of the Internet-of-Things is crucial for assuring that the right data is delivered to applications at the right time for prompt use. This, in what relates to data interoperability, means making sure that data is provided out of the IoT to many applications and in a way that can be processed and understood by these.

So, the research work of this thesis especially addresses the (data) interoperability problem in complex large-scale environments like the Internet-of-Things, i.e. heterogeneous settings with a considerable number of systems that support a variety of data formats and that need to be made interoperable. Furthermore, the primary application domain of the proposed ideas is also the Internet-of-Things, this despite the fact that the envisioned concept and related technological solution are presented in a general form in order to be applicable to any area of use.

Interoperability subjects– Any research work must define its boundaries or else it might lose focus and purpose. The thesis research specifically explores data-sharing environments where constituent systems ('things') effectively want to exchange data with others (in the environment). That is, systems that are sharing data (possibly using different data syntaxes/semantics, causing non-interoperability) but that enclose information of the same nature, e.g. data systems that exchange (send/receive) data about environmental parameters such as temperature, humidity, noise, etc. or other.

A good example of this is IoT for Home Automation. There are a huge number of Do-It-Yourself (DIY) sensing solutions for home automation available in the market today (and more will appear), that are able to provide valuable data, such as classical environmental parameters, but also energy and water usage, and much more. Usually, these solutions "speak" proprietary data formats and come with only one application out-of-the-box. On the other hand, there are a huge number of applications (and others to appear) for exploiting data from sensing solutions in the home, e.g. for energy and water monitoring, lighting and appliances control, surveillance, etc. Being able to interoperate (the possibly huge variety of) sensing systems with (the many interesting) other applications is something that would be very welcome by users but is hard to do.

Now, usually one sensing system has been designed for the use of one application – the one that comes bundled with the system –, and to provide data to other (data-incompatible) system requires changes in the sensing system to add new interoperability-enabling supports. Also, the application that comes with the DIY solution usually can only get data from the sensing system out-of-box and, to enable it to take data from others, might also require changes in the application to include more interoperability-enabling supports. But, these DIY home automation solutions are usually proprietary/closed/legacy systems, which cannot undergo changes. So, this type of systems hinder the most interoperability as the impossibility to change them might imply the need to alter some other systems to try and make it all interoperable or maybe those may never become interoperable in the environment.

So, the research in this thesis looks with particular interest at the properties and implications of systems that cannot be modified (to include new interoperability-enabling supports) and their impact on Interoperability; this, adding up to the observation and study of (data) interoperability scenarios of large-scale nature.

1.2 Research Methodology

The goal of this work is to perform fundamental/basic research in data interoperability *"directed towards finding information that has a broad base of application and thus, adds to the already existing organised body of scientific knowledge"* (Kothari, 2004). This research work aims, firstly, to provide a greater knowledge of the fundamental aspects (quantified inefficiencies, non-performance, etc.) of data interoperability, especially in large-scale settings, and secondly, to deliver a fit solution to data interoperability in such kind of scenarios; and this, without having any specific application in mind, although a few have already been identified. Now, for performing this systematic study an adequate (research) methodology is needed.

A research methodology establishes the way to systematically address the research problem. Essentially, the research methodology defines procedures and steps by which researchers go about their work of describing, explaining and predicting phenomena (Rajasekar, Philominathan, & Chinnathambi, 2013). The research methodology used in this thesis is based on the Scientific Method, which is generically composed of the steps depicted in Figure 11.

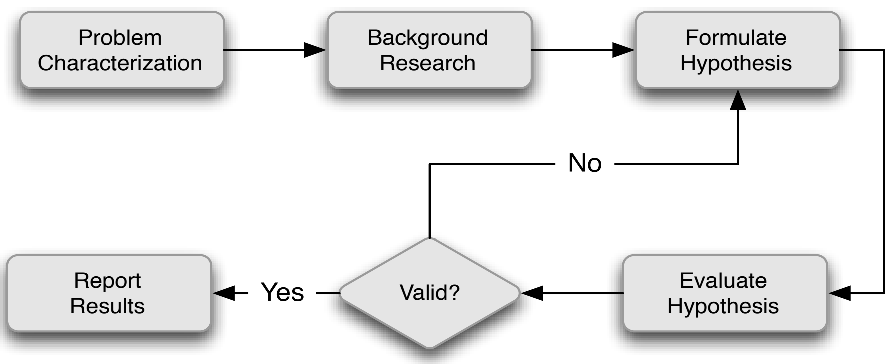


Figure 11: Research Methodology

The Problem Characterisation relates with the definition of the research problem that rules the onwards research work. In this step it is characterised the problem that is the subject of the research on the thesis. Step ends up with the so-called research question that focuses the research works. This is a clear, focused, concise, complex and arguable question around which any research is centred (Kishore, Vasundhra, & Anand, 2011). Onwards research will attempt to answer the question posed. This research of this thesis looks particularly at the limitations of Point-to-Point Data Interoperability approach in large-scale interoperability scenarios.

(Background) Research step follows immediately after the research problem having been clearly identified and characterised and the research question posed. The research step relates with looking, inspecting and reflecting on possible ways to address the problem. Research in this thesis looked into the data interoperability domains and especially to other domains of practice and their supporting systems that might relate to a data interoperability support system and where similar problems and equivalent challenges exist. The study of these comparable systems provided important hints and good inspirations to develop new concepts and solutions to (data) interoperability. Research looked at commercial air transportation systems that exhibit many similarities to the key challenges of data interoperability support systems.

Having done the proper background research it then possible to Formulate Hypothesis. Based on the background research performed a hypothesis is formulated to bring clarity, specificity and explain a possible solution that might respond to the (research) question in-place. In the scope of this thesis it is established the proposed solution as the novel approach for data interoperability routed in principles from commercial air transportation systems. The hypothesis set a new data interoperability approach that can outperform Point-to-Point Data Interoperability approach in large-scale interoperability scenarios.

And then, it is needed to Evaluate Hypothesis. This step represents the actions and procedures established with the purpose to evaluate the proposed hypothesis as a way to assess its validity. In this thesis, the evaluation will be performed via a comparative analysis – benchmarking using comprehensive computer simulations – of both interoperability approaches, and from which conclusions will be drawn. Benchmarking of the Hub-and-Spoke Interoperability and P2P Interoperability approaches primarily focuses large-scale conditions but a small-scale scenario analysis is also performed.

Finally, after hypothesis is considered valid, it is time to Report Results. This stage relates with reporting the results of the research. It relates, for instance, with the writing of the thesis document as a coherent whole that explains and details the research work. But it also relates to the communication of the results to the scientific community for wider peer evaluation.

1.3 Thesis Organisation

This thesis is organised into eight comprehensive chapters. As far as possible, each chapter is self-contained, meaning that the reader might take any given chapter and be able to fully understand its content and message, without needing to read any of the previous chapters. This way, the reader can take each chapter of this document in the order that he/she sees fit and/or whose contents interest him/her the most. This design is also intended to give the author the possibility of sharing any part of this thesis without having to distribute more content than strictly necessary.

Chapter one (this chapter) provides an introduction to the thesis document, composed of three parts. The first subsection provides a brief yet comprehensive introduction to Interoperability, in order to establish a common level of understanding between the reader and the author regarding Interoperability and its definitions, scopes, focuses, approaches, etc.; The second subsection presents and details the methodology that rules the research work; The third and final, part of chapter one (this subsection) explains how the thesis document is structured and the reasons for such organisation.

From this point onwards, the thesis is nicely organised following the steps of the methodology presented in the previous section. That is, chapter two goes into identifying, defining and characterising the state-of-the-art on data interoperability approaches that do not consider standards – particularly P2P Interoperability. Chapter three details the research problem towards establishing the question that is the centre of the research – the research question. Chapter four describes the research for moving from problem to solution (hypothesis), by studying a complex system that exhibits comparable challenges to those of data interoperability – the commercial air transportation system –, reflecting on its outstanding approaches and exploring how these can be explored in the interoperability domain. Chapter five establishes the proposed solution as the novel concept for (data) interoperability, concluding with the definition of the hypothesis. Chapter six presents then the evaluation of the hypothesis, performed via a comparative analysis – supported by a set of computer simulations – of both interoperability approaches, and from which conclusions are then drawn.

Then, chapter seven gives already a view on an application of the proposed concept – the Plug'n'Play Interoperability technological solution. And finally, chapter eight provides the conclusions and possible future developments related to this research.

Now in more detail, chapter two goes into describing, properly defining and thoroughly characterising the state-of-the-art on the established data interoperability approach that does not consider using standards – Point-to-Point (P2P) Interoperability. This chapter explains the main concepts of P2P Interoperability and how it works for establishing interoperability. Here, is provided a comprehensive definition and properties of the P2P approach to interoperability following a mathematical formalisation using set theory and already establishing a graphical notation for the defined elements of interoperability. Then, it is provided a definition of an algorithm to achieve the maximum possible interoperability in P2P interoperability approach scenarios using matrix theory. This algorithm defines the dynamics and behaviour of P2P interoperability while going about achieving interoperability to the maximum that is possible in a scenario.

Chapter three presents the interoperability problem to be researched: the unfitness of Point-to-Point Interoperability on the large-scale. To this, a much detailed analysis and in-depth study of P2P interoperability is provided, focusing especially on large-scale conditions, considering that nowhere in literature this is well defined and sufficiently detailed so that it enables the understanding of its exact problems and limitations. For such a purpose, first are defined the key metrics for assessing interoperability as the means to evaluate (P2P and other) interoperability approaches. Then, an exhaustive study is performed that pinpoints the fundamental problems of the P2P data interoperability approach in large-scale situations. And from there, a proper research question arises; that is, a clear, focused, concise, complex and arguable question around which any research is centred (Kishore, Vasundhra, & Anand, 2011). From then on, work is directed at answering the (research) question in place.

Chapter four presents the research work carried out to progress from the identified problems towards a proposed solution (hypothesis). Here, lateral research is performed, taking trails that are not the obvious ones by looking at adjacent domains of science and application to seek inspiration and fit solutions. Research goes deep into commercial air transportation systems after these are identified as having clear similarities and challenges to those of interoperability support systems. Then, by inspecting, studying and reflecting on the outstanding approaches and solutions of commercial air transportation systems, an idea is suggested: to explore the exceptional approaches and strategies of commercial air transportation systems in interoperability, i.e. to develop a new idea of “Interoperability out of the skies”.

In chapter number five a proposed solution is presented as the hypothesis. This chapter starts with the definition of a new, accurate approach to (data) interoperability. Like before, a full definition and properties of the proposed new concept for interoperability are given, again following a mathematical representation using set theory and establishing a graphical notation for the new objects. Alongside this, and once more, a definition of an algorithm is provided to achieve the maximum possible interoperability in light of the new concept for interoperability. Then, the results of a set of simulations that outline the behaviour of the new interoperability approach are presented. At the end of the chapter, and based on the above, a hypothesis is constructed as if this can to address the key problems of interoperability as defined initially in the thesis.

Then, in chapter six, the hypothesis is evaluated, that is, the new interoperability approach is analysed and observed to assess its ability to solve the problems as initially defined. The evaluation step is performed through a comparative analysis of the behaviour of both the proposed approach to interoperability and the classical point-to-point interoperability approach. A verdict in then established as to whether the proposed hypothesis is able to address the fundamental problems of interoperability. This verdict, proven to be correct, establishes the new proposal as a fittest approach to data interoperability, especially in large-scale settings. It establishes, without a shadow of a doubt, that the proposed approach to interoperability outperforms the traditional P2P interoperability in view of the identified problems.

Having proved the hypothesis as sound, chapter seven presents a way to go about implementing the new approach of interoperability. For this, and inspired in the principles of plug-and-play, a definition of a new technological solution is provided, that realises the new concept of interoperability – the Plug'n'Interoperate (PnI). Next, a reference architecture for the Plug'n'Interoperate technological solution is provided, one that can be used – by the author and also by others – for implementing the technologies that will make it work. Two views of the architecture are provided: a logical view – enclosing the logical modules/components and its relationships; and a functional view – presenting how the solution might be function in practice, following a distributed computing paradigm.

Finally, chapter eight wraps it up by providing the conclusions and already indicating some paths for future evolutions and developments built around this work. The conclusions ties together the research and other subjects addressed in the body of the thesis, to present readers with a final overview and observations. Then, and still inside the chapter, the future works that open up the stage for future research and development actions are presented. These might be used as starting points for follow-up research and/or technological developments related to the proposed activities.

And, as already noted, the core text of the thesis has been further complemented with a Preamble, presented at the beginning of this document, where the author provides a comprehensive view of the paths that led to this research and to the PhD thesis. Also, a Prologue is presented that details the realistic scenario from which the author got the motivation and inspiration for conducting the research works.

Then, at the very end of the document, a set of Appendixes is provided as a collection of supplementary material related to the thesis. First appendix provides complete and comprehensive details about the simulation methodology used to simulate the interoperability approaches and the code used to perform these simulations, as well as the code used to produce the many graphics presented in this thesis.

Second appendix encloses the many graphics obtained as result from the simulations of the Point-to-Point Interoperability. Third appendix presents the graphics obtained as result from the simulations of the Hub-and-Spoke Interoperability approach. Forth appendix presents the graphics used to analyse the Hub-and-Spoke interoperability approach performance in small-scale conditions. And the last appendix enfolds the set of graphs used to thoroughly compare the Hub-and-Spoke and the Point-to-Point Interoperability approaches.

2

State-of-the-Art: Point-to-Point Interoperability

"Realise that everything connects to everything else."

– Leonardo da Vinci, Italian Renaissance polymath

In a scenario composed of many heterogeneous systems, situations often occur when two systems want to exchange information but cannot do so because they use different data representations, rendering these systems non-interoperable. Point-to-Point (P2P) Interoperability is the typical approach used to make a set of systems interoperable, where each system exchanges data directly with any other system.

The study of the P2P approach to interoperability, aiming to understand its main characteristics, and identifying its strengths and especially its weaknesses, requires the definition of its underlying concepts and properties. Model is based on a sound mathematical formalism providing a rigorous description and that makes it easier the translation of concepts into related computer representations. Also, a graphical notation in set, especially for providing visual examples of interoperability scenarios.

And then, the mathematical model serves as baseline for the definition of an algorithm to achieve the maximum possible interoperability in a P2P Interoperability scenario. The algorithm defines the behaviour of P2P Interoperability on interoperating altogether systems in a scenario and is provided using matrix theory representations.

2.1 Concept

Data Interoperability is about making sure that systems can share data between each other and that they understand such exchanged data. A system is seen here "simply" as an entity that can provide data and/or consumes data. And considering this, two given systems are then said to be interoperable if one system provides data in such a way that one other system is able to consume. Systems exchange data following representations called data formats. Two systems can only exchange data if both support one same data format. Systems are interoperable only if a provider system presents data in a given data format that a data consumer system can understand.

Point-to-point (P2P) interoperability is an approach to data interoperability that is characterised by direct system-to-system connections to establish interoperations between systems (Morris, Levine, Meyers, Place, & Plakosh, 2004). An interoperation defines the ability of two systems to exchange data via a determined data format. The P2P interoperability approach considers that two given system that are supposed to exchange data are directly data-linked via some data format, i.e. via an interoperation.

To exemplify the P2P interoperability approach, let's consider the basic interoperability scenario of Figure 12, comprised of five systems that want to share data with all the others and where it exists five data formats but each system provides data in only one format (different from all the others). The symbols (arrows) pointing out of each system represents the ability to provide data in a given data format (*'A'*, *'B'*, *'C'*, *'D'* or *'E'*).

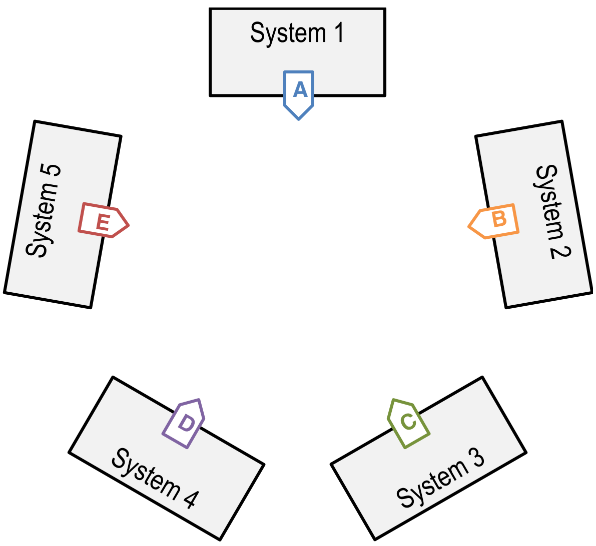


Figure 12: Five-systems five-formats interoperability scenario example

Turning the scenario presented in Figure 12 fully interoperable using the Point-to-Point Interoperability approach implies then to establish direct system-to-system data links. That is, establishing interoperations between pairs of systems, which in practice means to have a systems able to take data in the format of the provide system. And, this considering all the five systems present in the interoperability scenario.

Let's consider first the P2P Interoperability approach following a local (interoperability) optimisation strategy. In the local P2P Interoperability approach, optimisation of interoperability is viewed locally only, at each system, irrespective of any other systems' interoperability optimisation. In this case, considering the example of Figure 12 and observing from any given system in the scenario, one can see that it is needed to establish (in that system) the ability to consume data from all four other systems. Figure 13 depicts the fully interoperable scenario example using the local P2P Interoperability approach. The representation exhibits data systems, their ability to provide/consume data in given data formats and the interoperations between them. Note the filled coloured shapes that represent newly added interoperability supports.

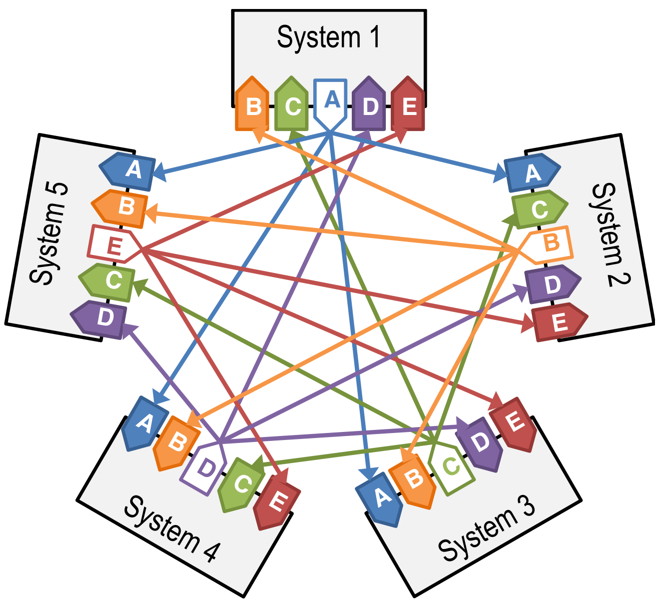


Figure 13: Fully-Interoperable five-systems five-formats interoperability scenario example using the local Point-to-Point Interoperability approach

By following the local Point-to-Point Interoperability approach it would be needed some 20 new supports – let's called them interoperability artefacts from here on – for establishing full interoperability in the scenario. Considering that one is looking at a scenario with only a few systems, this already looks like a considerable number of new artefacts needed. Moreover, a larger number of systems imply an even greater number of interoperability artefacts to establish full interoperability in the scenario.

Now, let's consider a different way: the Point-to-Point Interoperability approach using a global (interoperability) optimisation strategy. In this scheme, the optimisation of interoperability considers the whole of the scenario, i.e. optimisation considers a global view of the interoperability scenario. Let us then take a step-by-step method in order to fully understand the behaviour of the global P2P Interoperability approach. That is, let's consider that we add interoperability artefacts, one a time, aiming to maximise interoperability with the fewest resources (artefacts) needed.

As such, and looking at the entire scenario, one can see that starting off by implementing interoperability artefacts for consuming either *'A'*, *'B'*, *'C'*, *'D'* or *'E'* on any system (other than the one that would represents a self-interoperability) returns the same in respect to interoperability maximisation: it turns two systems interoperable. So, let's select consider that we add a data importing interoperability artefact of format *'A'* to *'System 1'*. Next step: we see now that we have more presence of artefacts of format *'A'* in the scenario than all other and, if we keep selecting artefacts of format *'A'*, this might mean (in potential and in future steps) further options for maximising interoperability. And so, if we keep on with this rational, we get to the intermediate state of scenario interoperability optimisation presented in Figure 14.

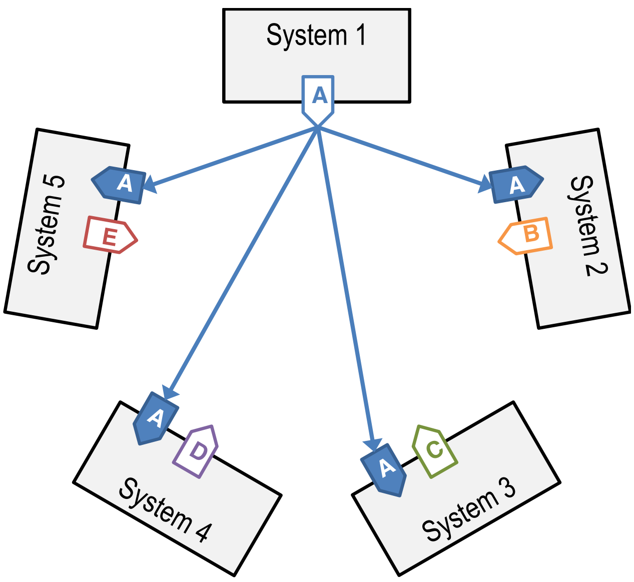


Figure 14: Intermediate state of the exemplary scenario optimisation following the global Point-to-Point Interoperability approach

Now, and considering this state, we again see that the best decision for further optimisation of interoperability is to continue and add a new interoperability artefact to export data in format *'A'* in either *'System 2'*, *'System 3'*, *'System 4'* or *'System 5'*, as this will make interoperable three system-to-system pairs. That is, adding to one of this systems a new interoperability artefact that exports data in format 'A' will make such a system interoperable with the other three systems that can already consume data following data format *'A'*.

Moving on, we reach the final state of the optimisation process (depicted in Figure 15) where no more optimisation steps are possible. In this final state, the scenario is fully interoperable, i.e. all systems can exchange data with any other system in the environment, as required initially.

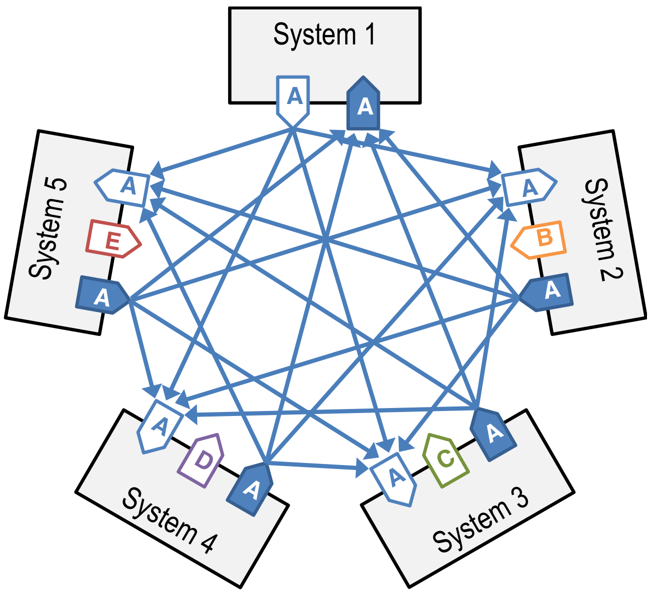


Figure 15: Final state of the exemplary scenario optimisation following the global Point-to-Point Interoperability approach

As noticeable, it was needed a total of nine interoperability artefacts to turn the scenario fully interoperable by using a Point-to-Point Interoperability approach following a global optimisation scheme. Obviously, a global view of the scenario establishes a better performance of the approach and consequently fewer interoperability resources are needed to accomplish interoperability in the scenario. And as such, we will focus onwards analysis on the Point-to-Point Interoperability approach using a global optimisation strategy.

2.2 *Definition and Properties*

2.2.1 System Placeholder and Interoperability Scenario

Interoperability is defined between systems. A System is defined by (2.2.1); n is the number that identifies a system in an interoperability scenario, i.e. the system identifier. Each system has a unique identifier within a given interoperability scenario.

$$\mathrm{S_n} = \mathrm{System}\,, \qquad \mathrm{n} \in \mathrm{Z}^+ \tag{2.2.1}$$

Take notice that at this point, a system represents simply a placeholder and will be comprehensively defined further ahead, still within this section.

Graphically, a system is represented by a light rectangle with borders, as illustrated in Figure 16. In the graphical notation, a system is identified by a text label using the format "S_n" – capital italic 'S' for system and subscript ordinal 'n' representing the system identifier. Example, label "S_1" corresponds to a system whose identifier is '1'.

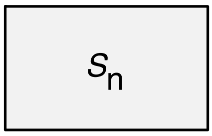


Figure 16: Graphical notation of the system 'n'

Then, an Interoperability Scenario consists of a collection of distinct systems – i.e. a set of systems – where each disparate system is required to exchange information with other systems, in order to be able to achieve its goal. This is given by (2.2.2),

$$\begin{aligned}\mathrm{Z} &= \mathrm{Interoperability\ Scenario} \\ &= \{\mathrm{S_1}, \mathrm{S_2}, \ldots\}\end{aligned} \tag{2.2.2}$$

The Number of Systems in an Interoperability Scenario is given by (2.2.3). The Number of Systems ($\mathrm{N_S}$) is denoted by the cardinality of the Interoperability Scenario set.

$$\begin{aligned}\mathrm{N_S} &= \mathrm{Number\ of\ Systems} \\ &= |\mathrm{Z}|\end{aligned} \tag{2.2.3}$$

2.2.2 Data Format

In an interoperability scenario, systems exchange information following data representations. Two arbitrary systems can only exchange information if, and only if, both support the same data representations. Practically speaking, this means that systems are required to follow matching data representations to be able to share data among themselves.

In this context, these data representations are called data formats. A Data Format specifies how the data (to be exchanged) is organised; basically, a data format defines how the (shared) data is represented and structured. A data format is formalised as presented in (2.2.4), where ‘m’ is the format identifier.

$$F_m = \text{data Format}, \qquad m \in Z^+ \tag{2.2.4}$$

Several data formats can exist in an interoperability scenario, but only those used by at least one system will be considered. From the interoperability perspective, formats that are not used by a minimum of one system are not considered relevant, and therefore are not contemplated. Data formats that are not used within the interoperability scenario do not present, so to say, interoperability value, as they are not adopted by and system. Those data Formats that are of interest in an interoperability scenario are represented in (2.2.5),

$$\begin{aligned} F &= \text{data Formats} \\ &= \{F_1, F_2, \dots\} \end{aligned} \tag{2.2.5}$$

And the Number of Data Formats is straightforwardly expressed by (2.2.6). The number of data formats (N_F) represents the total number of data formats existent in an interoperability scenario and is defined as the cardinality of the Data Formats set.

$$\begin{aligned} N_F &= \text{Number of data Formats} \\ &= |F| \end{aligned} \tag{2.2.6}$$

2.2.3 Exporter and Importer

To achieve data exchange interoperability between two given systems – a data provider and a data consumer system – the data format to be used needs to be supported at both ends. The source system must have the capability to send information in a data format that the target system is able to accept. This condition of systems is realised by two (interoperability) artefacts: data exporters and importers.

An Exporter is as an (interoperability) artefact that represents the capability of a system to make information available in a given data format. The ability of a system to provide data following a given data format is qualified by the existence of an exporter. Contrariwise, the absence of a certain data exporter implies that the system cannot deliver data using such data format.

An exporter is an association of a system (where it is implemented to perform the data export function) and a data format (the exact format in which data is made available). Therefore, an exporter is identified by a system-format pair, which provides the notion that a given system has the capability to offer its information in a particular data format. The nomenclature of (2.2.7) is used to represent an exporter, characterised by the system ($\mathrm{S_n}$) that implements the exporter and the data format ($\mathrm{F_m}$) that it uses to represent the information.

$$\mathrm{e}_{(\mathrm{S_n},\mathrm{F_m})} = \mathrm{Exporter} \qquad (2.2.7)$$

Graphically, an exporter is represented as an irregular pentagon – shaped like a kindergarten-drawn house or some kind of arrow – placed over a system's boundary, pointing towards the outside of the rectangle, symbolising the flow of data from the system to the outside (see Figure 17). The exact data format, which the exporter uses to represent the exported data, is indicated in a text label inside the exporter shape.

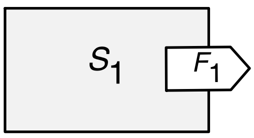


Figure 17: Example of a representation of a system (S_1) with an exporter of data format F_1

A system can have more than one exporter, being capable of representing its information in many different data formats. All exporters corresponding to the same system can be grouped together in a set, as represented in (2.2.8).

$$\begin{aligned} E_{S_n} &= \text{Exporters of system } S_n \\ &= \{e_{S_n,F_1}, e_{S_n,F_2}, \dots\} \end{aligned} \quad (2.2.8)$$

An Importer is an artefact that performs the reverse/opposite operation of an exporter: it represents the ability of a system to interpret information in a given data format. Like an exporter, an importer is identified by a system-format pair as it enables a system to take information expressed in a given data format. An importer is represented in (2.2.9), correlating the system that implements it (S_n) and the data format that it can interpret (F_m).

$$i_{(S_n,F_m)} = \text{Importer} \quad (2.2.9)$$

Graphically, an importer is also represented as an irregular pentagon (or arrow), drawn over a system's boundary, pretty much in the same way as an exporter, but pointing towards the inside of the system, representing the data flow from the outside-in. This notation is illustrated in Figure 18. And, like the exporter representation, the data format supported by the importer is written as text label inside the shape.

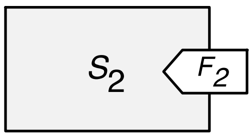


Figure 18: Example of a representation of a system (S_2) with an importer to data format F_2

Following a similar rationale to the one used in the exporter representation, all the importers that a system possesses are grouped together in a set, representing all the data formats that a given system (S_n) can accept, as represented in (2.2.10).

$$\begin{aligned} I_{S_n} &= \text{Importers of system } S_n \\ &= \{i_{(S_n,F_1)}, i_{(S_n,F_2)}, \dots\} \end{aligned} \quad (2.2.10)$$

Now, a system can include an exporter of a given data format and also an importer of that data format, meaning that such system can both provide and take data using one same format. In such a case, and with the objective of simplifying the graphical representation of systems and consequently of whole interoperability scenarios, there exists two alternate representations. The first, shown in Figure 19 left, includes both exporter and importer symbols; the second, presented in Figure 19 right, is one where the exporter and importer symbols have been merged together into a new symbol.

Figure 19: Alternative graphical representations of a system (S_3) having both an importer and exporter to data format F_1

2.2.4 System

In the context of an interoperability scenario, a system is characterised by all the import and export capabilities it has. Therefore, the <u>union of the sets of Importers and Exporters represents a System from an interoperability perspective</u>, as in (2.2.11).

$$S_n = E_{S_n} \cup I_{S_n}, \qquad S_n \neq \emptyset, n \in Z^+ \qquad (2.2.11)$$

In theory, a system can support an indefinite amount of importers and exporters. A system can have either the Importer set or the Exporter set empty; however the two sets cannot be empty at the same time, since that situation would correspond to a system unable to communicate at all. Theoretically, this condition might exist in an environment but is not relevant from a data sharing (thus interoperability) perspective.

The fact that a system can have one of the two sets empty – either the importer or exporter set – comes from the design intent of such a system which is linked to its nature. Some systems are designed to have only the capability to export data, whereas other systems are design only for data importing.

Systems that only come with exporters have the nature of data sources, only; systems that come just with importers have the nature of data consumers, solely. Alternatively, systems can include both exporters and importers, thus possessing a mixed/dual nature of a source of data and a data consumer.

2.2.5 Modifiable and Non-Modifiable System

Despite the fact that the whole of this characterisation represents an interoperability scenario in a certain state, it can and will evolve – new systems will enter and others may leave. To use this characterisation as a baseline study for the evolution of a system, one also needs to know if it is possible to make changes on systems.

A given system is said to be Modifiable if it includes mechanisms that enable the development and deployment of new exporters and/or importers into that system. When a system does not meet this requirement, it is known as Non-Modifiable. In this way, the data exchange capabilities of a non-modifiable system will never change.

Modifiability is therefore a property (of systems) that qualifies – true or false – the ability of a system to be altered in order to accommodate new importers and/or exporters. This is represented by (2.2.12), a two-elements set: a checkmark that represents the ability to be modified; and a cross, representing non-modifiability.

$$\begin{aligned} \mathrm{M} &= \mathrm{Modifiability} \\ &= \{\surd, \times\} \end{aligned} \qquad \textit{(2.2.12)}$$

Then, in a mathematical notation, a non-modifiable system is represented here as a system marked with the cross symbol in superscript, defined by (2.2.13). This representation establishes that a given system has the modifiability property set to false and therefore cannot be altered at all.

$$\mathrm{S}_\mathrm{n}^{\times} = \text{Non-Modifiable System}, \qquad \mathrm{n} \in \mathrm{Z}^{+} \qquad \textit{(2.2.13)}$$

Inversely, the representation of a system with the checkmark symbol in superscript means that the system can indeed be modified. The absence of any symbol in superscript, in the definition of a system, is semantically equivalent to having a checkmark, i.e. such a system is modifiable. This is defined by (2.2.14),

$$\mathrm{S}_\mathrm{n}^{\surd} = \mathrm{S}_\mathrm{n} = \text{Modifiable System}, \qquad \mathrm{n} \in \mathrm{Z}^{+} \qquad \textit{(2.2.14)}$$

Graphically, a non-modifiable system is represented by a rectangle (logically, since it is still a system) with a dark filling, symbolising a “closed box” to denote that it cannot be changed, as illustrated in Figure 20.

Figure 20: Graphical representation of a Non-Modifiable system (S_4)

And, for the sake of simplicity of representation and understanding, a Modifiable system is drawn in the very same way as a (plain) system, i.e. a light rectangle with borders, as depicted in Figure 21.

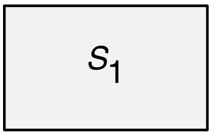


Figure 21: Graphical representation of a Modifiable system (S_1)

So, based on all these definition, the set of all non-modifiable systems of an interoperability scenario – denoted by $\mathrm{Z}^{\times}$, zed with a cross in superscript for non-modifiability – is defined as (2.2.15),

$$\begin{aligned} \mathrm{Z}^{\times} &= \text{Non-Modifiable systems of an Interoperability Scenario} \\ &= \{\mathrm{S}_1^{\mathrm{x}}, \mathrm{S}_2^{\mathrm{x}}, \dots\}, \qquad \mathrm{x} \in \mathrm{M}, \mathrm{x} = \times \end{aligned} \tag{2.2.15}$$

And also, the set of all modifiable systems inside an interoperability scenario – denoted by $\mathrm{Z}^{\surd}$, zed with a checkmark in superscript – is defined as (2.2.16).

$$\begin{aligned} \mathrm{Z}^{\surd} &= \text{Modifiable systems of an Interoperability Scenario} \\ &= \mathrm{Z} - \mathrm{Z}^{\times} \end{aligned} \tag{2.2.16}$$

Take notice that Z – zed without any symbol whatsoever in superscript – still qualifies the Interoperability Scenario, i.e. the set of all systems, and not only set of those systems that are modifiable.

2.2.6 Interoperability

The flow in the data exchange between two given systems is seen from the data exporter system (i.e. the system providing the data) to the importer system (i.e. the system consuming the data). When both the consumer system and the provider system have a compatible importer and exporter (i.e. both artefacts of one same data format), an interoperation between those systems is said to exist. Thus, an Interoperation defines the ability of two systems to exchange data by following a specific data format.

Mathematically, an Interoperation is then formalised as a match of an exporter of a provider system to an importer of a consumer system, by a data format, as duly defined in (2.2.17).

$$\begin{aligned} t_{(S_p,S_c,F_m)} &= \text{Interoperation of provider } \left(S_p\right) \text{ to consumer } (S_c) \text{ via format } F_m \\ &= \left(e_{(S_p,F_m)}, i_{(S_p,F_m)}\right) : \left(e_{(S_p,F_m)} \in S_p \wedge i_{(S_c,F_m)} \in S_c\right) \text{ , } S_p, S_c \in Z \end{aligned} \quad (2.2.17)$$

And, an interoperation is graphically represented as a directed edge (or arrow) between matching exporter and importer of provider and consumer systems, respectively, going then from the exporter (of the source system) and pointing to the importer (of the target system), as exemplified in Figure 22.

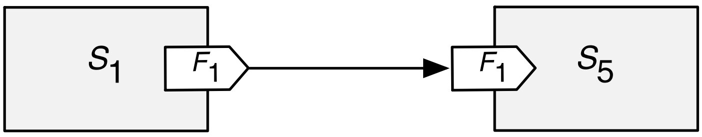


Figure 22: Graphical representation of the interoperation between S_1 and S_5 using format F_1

One system may be interoperable with one other system via several different formats, i.e. there can be more than one interoperation (in one same direction) between both systems. Do, the set of all interoperations from a provider to a consumer is as (2.2.18),

$$\begin{aligned} T_{(S_p,S_C)} &= \text{Interoperations from provider } (S_p) \text{ to consumer } (S_c) \text{ systems} \\ &= \left\{t_{(S_p,S_C,F_m)}\right\}, \forall F_m \in F \end{aligned} \quad (2.2.18)$$

Consequently, the number of interoperations between two systems – from provider system to consumer system – is (2.2.19),

$$\begin{aligned} C_{(S_p,S_c)} &= \text{Number of interoperations from a provider } (S_p) \text{ to consumer } (S_c) \\ &= \left|T_{(S_p,S_c)}\right| \end{aligned} \tag{2.2.19}$$

Then, from that, the set of all Interoperations that exist in an interoperability scenario is defined by (2.2.20), i.e. the set of all interoperations for all system in the scenario.

$$\begin{aligned} T &= \text{Interoperations in an Interoperabity Scenario} \\ &= \{T_{(S_n,S_m)}\}, \qquad \forall S_n \in Z, \forall S_m \in Z \end{aligned} \tag{2.2.20}$$

Finally, Interoperability is said to exist from a provider system to a consumer system if, and only if, there is at least one interoperation between them; conversely, systems are not interoperable if the number of interoperations between them is zero. The existence of Interoperability from a provider to a consumer system is the formally set by (2.2.21).

$$\begin{aligned} X_{(S_p,S_c)} &= \text{Interoperability between a provider } (S_p) \text{ to consumer } (S_c) \\ &= \begin{cases} 1, & C_{(S_p,S_c)} > 0 \\ 0, & C_{(S_p,S_c)} = 0 \end{cases} \end{aligned} \tag{2.2.21}$$

Building on this, and if required for any simple exemplification of an interoperability scenario, it is possible to express the existence of Interoperability between two systems simply by using the graphical notation depicted in Figure 23. That is, drawing a directed edge (or arrow) from the border of the source system to the border of the target system, stating thus interoperability from source to target, only.

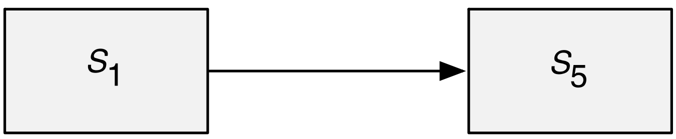


Figure 23: Graphical notation for describing Interoperability between S_1 and S_5

2.3 Algorithm to Maximum Possible P2P Interoperability

One of the reasons to formalise an interoperability scenario – definitely the main one regarding the goal of this research work – is to subsequently use those models to achieve (determine) the maximum possible interoperability between every system present in the scenario. The interoperability models thus serve as the formal support for a process of interoperability maximisation in a given interoperability scenario. Particularly, the Point-to-Point (P2P) Interoperability model, as defined in the previous section, serves as the baseline for an algorithm to achieve the maximum possible interoperability in a P2P Interoperability scenario.

The algorithm operative principle is the modification of some (to all) of the systems that compose the interoperability scenario. The modifications of the systems are performed via the introduction of new or exporters that, in this way, expand the capabilities of systems. And these changes are performed through a step-by-step (i.e. system-by-system) approach, where each system is modified in turn, in order to support the data representations needed to maximise interoperability in the scenario.

The starting point for this algorithm is the representation of both the exporting and importing capability of systems that take part in the interoperability scenario. This is done here using a matrix representation, where two matrices are defined: one, representing the Importers – matrix I; and the other – matrix E – representing the Exporters (per system, per format). Both matrices are presented in (2.3.1) and (2.3.2).

$$\begin{aligned} \mathrm{I} &= \text{Importers per System per Format} \\ &= [\mathrm{i_{nm}}], \qquad \mathrm{n} = 1,2,\dots,\mathrm{N_S};\, \mathrm{m} = 1,2,\dots,\mathrm{N_F};\, \mathrm{i_{nm}} \in \{0,1\} \end{aligned} \tag{2.3.1}$$

$$\begin{aligned} \mathrm{E} &= \text{Exporters per System per Format} \\ &= [\mathrm{e_{nm}}], \qquad \mathrm{n} = 1,2,\dots,\mathrm{N_S};\, \mathrm{m} = 1,2,\dots,\mathrm{N_F};\, \mathrm{e_{nm}} \in \{0,1\} \end{aligned} \tag{2.3.2}$$

These two matrices establish the core understanding of the interoperability scenario as a whole and at a given stage. At any selected moment, this pair of matrices defines the (interoperability) state of an interoperability scenario. Thus means, that it is possible to assess the interoperability status of an interoperability scenario by confronting the 'Exporters' and 'Importers' matrices.

As shown in Figure 24, each matrix – I and E – has the same structure: data formats represented in columns and systems represented in lines. Thus, both matrices have as many rows as there are systems in the interoperability scenario, and as many columns as the total of data formats. The matrices are populated using a binary representation – 0's or 1's – where '0' states the inability of a system (in rows) to export/import a data format (in columns) while a '1' means that it can export/import data using that format.

$$\mathrm{E} = \begin{matrix} \\ \mathrm{S_1} \\ \mathrm{S_2} \\ \mathrm{S_3} \end{matrix} \begin{matrix} \mathrm{F_1} \quad \mathrm{F_2} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix} \qquad \mathrm{I} = \begin{matrix} \\ \mathrm{S_1} \\ \mathrm{S_2} \\ \mathrm{S_3} \end{matrix} \begin{matrix} \mathrm{F_1} \quad \mathrm{F_2} \\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

Figure 24: Example of 'Exporters' and 'Importers' Matrices

Maximising possible interoperability between systems depends on the nature of each particular system: a system can be designed to export only, to import only or to both import and export. This means that if a system is designed from the start without any exporter – so having only importer(s) – then it is not supposed to have one (ever), because is not part of its nature: and the same thing goes for export-only systems.

Another limiting factor is the modifiability of systems: if a system cannot be modified, it certain situations can occur where the data exchange between that system and others cannot be established. The modifiability characteristic of systems of an interoperability scenario is represented by a matrix of systems vs. modifiability capability, as in (2.3.3).

$$\begin{aligned} \mathrm{M} &= \text{Modifiable Systems} \\ &= \left[\mathrm{m_{ij}}\right], \qquad \mathrm{i} = 1,2,\dots,\mathrm{N_S}; \mathrm{j} = 1; \mathrm{m_{ij}} \in \{0,1\} \end{aligned} \qquad (2.3.3)$$

Matrix M is a one-dimensional matrix (i.e. it has one single column) where each position in the matrix can also only hold a '1' – confirming a modifiable system – and a '0' – stating the contrary, i.e. that the system is non-modifiable. The example of Figure 25 presents such a matrix, together with the previously defined E and I matrices, so it could be used to explain the algorithm throughout the whole of this section.

$$\mathrm{M} = \begin{matrix} \mathrm{S_1} \\ \mathrm{S_2} \\ \mathrm{S_3} \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Figure 25: Example of a Modifiable Systems Matrix

Now, to improve interoperability, it is first necessary to have a view of the state of interoperability in an interoperability scenario. This is given by the interoperations between systems, obtained by matching compatible systems' exporters and systems' importers, as duly defined in the previous section. Here, it is represented through a system-to-system interoperability matrix, where each entry in the matrix states that a system – in rows – is interoperable with another system – in columns.

$$C = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

Figure 26: Example of an Interoperability Matrix

Take Figure 26, keeping in mind the working example of Figure 24 and focus now on the calculation of the matrix entry that represents the connectivity of system S_3 (third row) to system S_1 (first column). With the exporters of S_3, one has the ability to export with format F_2; with the importers of S_1, the ability to import using format F_2; and so, data exchange from S_3 to S_1 is possible, as shown in Figure 26. Bear in mind that the diagonal of the Interoperability matrix is to be ignored, because it represents the ability of a system to interoperate with itself and this is assumed not to be a needed goal.

The chosen matrix representation has a property that with only one operation results the Interoperability matrix: matrix product. The product of matrices A by B – denoted as AB – indicates a row-by-column multiplication, where the entries in the i^{th} row of matrix A are multiplied by the corresponding entries in the j^{th} column of matrix B and the results are added altogether. As such, matrices A and B can only be multiplied if the number of columns in A is equal to the number of rows in B.

The matrix product performs the exact operation that's needed for producing the Interoperability Matrix, with only a small detail missing, which is for the 'Importers' matrix to be transposed; this, for making it possible to perform the matrix product operation. And so, the Interoperability (between Systems) Matrix is defined by (2.3.4):

$$\begin{aligned} C &= \text{Interoperability (between Systems)} \\ &= EI^{T} \end{aligned} \qquad (2.3.4)$$

Take notice that, when using the matrix product, each entry in the Interoperability matrix is not only giving '1' or '0' – representing, respectively, yes-interoperability or non-interoperability between systems –, but it also provides the number of exporter/importer pairs (i.e. interoperations) that permit interoperability between two systems in an interoperability scenario. Look at the simple example in Figure 27, where the Interoperability Matrix (C_1) entry of S_1 to S_2 (first row to second column) has 2 as result, indicating that S_1 can share data with S_2 by using either formats F_2 or F_3.

$$E_1 = \begin{array}{c} \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{ccc} F_1 & F_2 & F_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \qquad I_1 = \begin{array}{c} \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{ccc} F_1 & F_2 & F_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array} \qquad C_1 = \begin{array}{c} \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{cc} S_1 & S_2 \end{array} \\ \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \end{array}$$

Figure 27: Example of an Interoperability Matrix with an entry over value '1'

This Interoperability matrix gives information about the already interoperable systems. So, it is the starting point for knowing where to establish new interoperations for achieving maximum interoperability in the interoperability scenario. And since the only permitted changes are the creation of a new exporter or of a new importer, the next step is to rank these potential new importers/exporters by the impact they represent in the overall interoperability measure of an interoperability scenario.

To determine the impact that the creation of any importer or exporter has on the overall interoperability, three types of information are needed: (1) the Interoperability Matrix, to be used as baseline to determine the interoperations needed to achieve maximum interoperability; (2) the 'Exporters' and 'Importers' Matrices, representing the existing importing and exporting capabilities of systems (using designated data formats) in an interoperability scenario, to be used for knowing where to improve; and (3) the Modifiability Capacity Matrix, that asserts the modifiability of systems – which ones allow/disallow changes – to be used to understand permission/avoidance to introduce new importer/exporter artefacts into systems. By using this information, it is possible to make a decision on how to improve interoperability in a scenario.

In order to decide which importer/exporter is the one that gives the best result, some operations about the above information need to be carried out. First, one must determine the pairs of systems where interoperability between them needs to be achieved. This will yield the understanding of those systems that need to be made interoperable to achieve maximum possible interoperability.

To determine of the pair of systems that require interoperability within a given interoperability scenario, one can observe a simple fact: if the Interoperability Matrix already gives the existing interoperation pairs between system pairs, then the “inverse” of such a matrix results in all the possible needed interoperations.

In matrix theory, this is represented by the logical negation. In a logical negation of matrix, each element that is non-zero in value becomes a zero, whereas any matrix entry valued with zero becomes a one, as defined in (2.3.5).

$$\neg(\mathrm{A})_{\mathrm{ij}} = \text{Logical Negation} \qquad (2.3.5)$$
$$= \begin{cases} 1, & \mathrm{a_{ij}} = 0 \\ 0, & \mathrm{a_{ij}} \neq 0 \end{cases}$$

Therefore, computing the logical negation of the Interoperability Matrix, results in the All-Possible-Needed-Interoperability Matrix, set in (2.3.6). This matrix establishes all possibilities of interoperations needed to turn a scenario fully interoperable.

$$\mathrm{N_{possible}} = \text{All-Possible-Needed-Interoperability} \qquad (2.3.6)$$
$$= \neg\mathrm{C}$$

This, applied to the running example in this section, results in the matrix presented in Figure 28. In practice, the matrix states that interoperability between systems S_3 to S_1 is not required, as these are already interoperable via data format F_2 – represented by the ‘0’ in the third line, first column entry – and that all other system-to-system interoperations (S_1 to S_1, S_1 to S_2, S_1 to S_3, S_2 to S_1, S_2 to S_2, S_2 to S_3, S_3 to S_2, S_3 to S_3) are possibly needed – represented by the ‘1’s in all other entries of the matrix.

$$\mathrm{N_{possible}} = \begin{matrix} & S_1 & S_2 & S_3 \\ S_1 & 1 & 1 & 1 \\ S_2 & 1 & 1 & 1 \\ S_3 & 0 & 1 & 1 \end{matrix}$$

Figure 28: Example of All-Possible-Needed-Interoperability Matrix

However, there are issues that dictate the impossibility of achieving full possible interoperability, and which must be considered in the calculations. They are:

1. Interoperability between same systems – It is assumed that no interoperations between the same types of system, as if a system is required to exchange data with itself, then this such will be resolved internally (within the system). Thus, one should remove the requirement for interoperability between same systems, which basically means to set to '0' all entries in the matrix main diagonal. For this purpose, one might use a filter, as defined in (2.3.7).

$$(\mathrm{F_{same}})_{ij} = \text{Filter Need for Interoperability of Same Systems} \qquad (2.3.7)$$
$$= \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases}, \qquad i, j = 1, 2, \ldots, N_S$$

   The filter represents a hollow matrix, which is a square matrix whose diagonal elements are all equal to zero, meaning that interoperability is not required between same systems. Figure 29 depicts such a filter for our example.

$$\mathrm{F_{same}} = \begin{matrix} & S_1 & S_2 & S_3 \\ S_1 \\ S_2 \\ S_3 \end{matrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

   Figure 29: Example of Filter Matrix for filtering need for Interoperability of same systems

2. Capability of the system to import and to export information – Systems that do not have at least one exporter (or importer) are so because they don't have that need in their functionalities (i.e. it is not in their nature do to so). And since the 'Needed Interoperability' matrix derives from the interoperability matrix, it loses the notion of whether the systems really want to connect to each other.

   Then, the 'Needed Interoperability' matrix must be filtered, by removing the need for interoperations between systems that do not have an exporter (or an importer). The filter matrix is created by: (1) extracting two new matrices from the 'Exporters' and 'Importers' matrices, showing whether the system has that capability (seeing if each system has an importer/exporter); and (2) establishing a correlation between the two, to understand which systems can and cannot (due to the exporter/importer limitation) exchange information between them.

To achieve (1) – the capability of each system to import/export –, one needs to create a new representation, which is a matrix of systems vs. whether it exports/imports, which is a one-dimensional matrix that derives from the 'Exporters' and 'Importers' matrices. We do so by performing a Logical OR – that mathematically represents the maximum of all elements – of all elements in the row, which represents all the formats of the system, defined as (2.3.8).

$$\begin{aligned}\left(\mathrm{capacity}((\mathrm{A})_{\mathrm{mn}})\right)_{\mathrm{ij}} &= \text{Capacity of Importer/Exporter} \\ &= \bigvee_{k=1}^{n} a_{ik}, \qquad i = 1,2,\dots,m;\ j = 1\end{aligned} \tag{2.3.8}$$

To achieve (2) – the matrix that represents the possible connections between the systems that have the capability to export to those that import –, the matrix product, as stated, is the way forward. So, multiplying the matrix of the capacity to export with the transpose of the import capability matrix, results in the matrix of 'Exporters' vs. 'Importers' that has the very same dimensions of the current needed interoperability matrix, defined in (2.3.9).

$$\begin{aligned}\mathrm{F}_{\mathrm{imp/exp}} &= \text{Filter the Capability of Import/Export} \\ &= [\mathrm{capacity}(\mathrm{E})][\mathrm{capacity}(\mathrm{I})^{\mathrm{T}}]\end{aligned} \tag{2.3.9}$$

Applying it to the run-on example, we find that in terms of export capacity, all the systems have it, but with respect to import capacity, system S_2 does not have an import capacity because it does possess any importer. This, in the filter matrix and following (2.3.9), is represented by having the S_2-related column set to zero, meaning that no information can be sent to that system. See Figure 30.

$$\mathrm{capacity}(\mathrm{E}) = \begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \mathrm{capacity}(\mathrm{I}) = \begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \qquad \mathrm{F}_{\mathrm{imp/exp}} = \begin{matrix} \\ S_1 \\ S_2 \\ S_3 \end{matrix} \begin{matrix} \begin{matrix} S_1 & S_2 & S_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

Figure 30: Example of the filter import/export matrix

The actual required interoperability between systems of an interoperability scenario (considering the above-defined aspects and/or restrictions that impede full possible interoperability) is then characterised by the all-possible needed interoperability filtered out of the interoperations between same systems and filtered out of the capability of systems to import/export.

Thus, the actual needed interoperability results from the application of these two filters to the all-possible-needed-interoperability matrix. Such filters are to be applied entry-to-entry to the matrix. In matrix mathematics, there is an operation that is the entry-wise multiplication (also known as the Hadamard, or Schur, product) that produces such outcome. The Hadamard product takes two matrices of equal dimensions, and produces another matrix where each entry ij is the product of entries ij of the two input matrices. I.e. the Hadamard product of two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ with the same dimensions (not necessarily square) with entries in a given ring is the entry-wise product $A \circ B = [a_{ij} b_{ij}]$, which has the same dimensions as A and B (Horn, 1989).

Thus, by doing the entry-wise multiplication of the 'All Possible Needed Interoperability' matrix, one can filter out the possibilities that don't apply to this scenario, as in (2.3.10).

$$\begin{aligned} N &= \text{Actual Needed Interoperability} \\ &= N_{possible} \circ F_{same} \circ F_{imp/exp} \end{aligned} \qquad (2.3.10)$$

This, applied to the previous working example and its matrices, produces the following:

$$N = \begin{matrix} & S_1 & S_2 & S_3 \\ S_1 & 0 & 0 & 1 \\ S_2 & 1 & 0 & 1 \\ S_3 & 0 & 0 & 0 \end{matrix}$$

Figure 31: Example of Actual Needed Interoperability Matrix

Following this, and in order to achieve interoperability between two systems, a match between the exporter and the importer that support the same data format is obligatory. So, the next logical step is to understand which is the needed importer or exporter to maximise interoperability. Starting with the actual needed interoperations, one can know which systems need to be interoperated.

For extracting which exporters need to be added to the environment, one starts with the importers. At first, it might be tricky to grasp the logic behind this, but, if one knows the actual needed connections and have the importers that the systems to interoperate support, achieving interoperability is as “simple” as creating an exporter for the already existing importer. This is performed via a matrix multiplication of the actual needed interoperability matrix with the ‘Importers’ matrix.

However, not all of the suggested – possible – importers and exporters can be implemented. This is due to the limitation that some systems cannot be modified and therefore only importers and exporters in modifiable systems can be taken into account. The suggested ‘Importers’ (and suggested ‘Exporters’) matrix filtered by the modifiable status of the systems, is directly obtained by a matrix multiplication of the Modifiable Systems Matrix with the ‘Importers’ (and ‘Exporters’) Matrix, which affects each system, by removing the importers/exporters that are in a non-modifiable system. This is defined as (2.3.11).

$$\begin{aligned} P_{exp} &= \text{Possible Exporters} \\ &= NI \circ (M[a_{mn}]), \qquad m = 1; n = 1,2,\dots,N_f; a_{mn} \in \{1\} \end{aligned} \qquad \textit{(2.3.11)}$$

For the sake of clarity and comprehension, again consider the running example. This time, take the computation of the ‘Possible Exporters’ matrix, as presented here in Figure 32, and following (2.3.11).

$$\begin{aligned} P_{exp} &= \left(\begin{matrix} & S_1 & S_2 & S_3 \\ S_1 & 0 & 0 & 1 \\ S_2 & 1 & 0 & 1 \\ S_3 & 0 & 0 & 0 \end{matrix} \; \begin{matrix} & F_1 & F_2 \\ S_1 & 0 & 1 \\ S_2 & 0 & 0 \\ S_3 & 0 & 1 \end{matrix} \right) \circ \left(\begin{matrix} S_1 & 0 \\ S_2 & 1 \\ S_3 & 1 \end{matrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \\ &= \begin{matrix} & F_1 & F_2 \\ S_1 & 0 & 1 \\ S_2 & 0 & 2 \\ S_3 & 0 & 0 \end{matrix} \circ \left(\begin{matrix} S_1 & 0 \\ S_2 & 1 \\ S_3 & 1 \end{matrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \\ &= \begin{matrix} & F_1 & F_2 \\ S_1 & 0 & 1 \\ S_2 & 0 & 2 \\ S_3 & 0 & 0 \end{matrix} \circ \begin{matrix} S_1 & 0 & 0 \\ S_2 & 1 & 1 \\ S_3 & 1 & 1 \end{matrix} \\ &= \begin{matrix} & F_1 & F_2 \\ S_1 & 0 & 0 \\ S_2 & 0 & 2 \\ S_3 & 0 & 0 \end{matrix} \end{aligned}$$

Figure 32: Example of ‘Possible Exporters’ Matrix

Take the first row of the 'Needed Interoperability' Matrix that dictates the need for interoperability of system S_1 to system S_3. As one knowns, the matrix multiplication is rows times columns, and so the first row (of the 'Needed Interoperability' Matrix) times the first column (of the 'Importers'' Matrix) gives '0', meaning that for such an interoperation, no exporter of the F_1 is required. This is due to two reasons: (1) system S_1 already has an exporter of F_1; but also (2) system S_3 does not have an importer of F_1; and so interoperability cannot be achieved through this.

The same operation, but now considering the first row (of the Needed Interoperability Matrix) times the second column (of the 'Importers' Matrix, the one for format F_2) results in a '1' which means that an exporter in system S_1 for F_2 connects to the importer of F_2 in system S_3.

After gathering all such possibilities, one has to filter them too in order to remove all the exporters that can not be implemented due to the non-modifiability characteristic of the system, which in this case results in the fact that the only exporting possibility is to include an exporter on system S_2 for data format F_2.

For finding the possible importers, the process is pretty much the same. The general idea is to get the possibilities, and so the starting point, once again, is the needed interoperability and now the already existing exporters. In the previous case, the correlation was between the rows and the importers columns; now, for finding the possible importers, the correlation is between the columns of the needed interoperability and the exporters, as the column view represents the import ability.

So, since the matrix multiplication is done following a line times column method, and the goal is to carry out the operation between two columns, the solution is to transpose the 'Needed Interoperability' Matrix so that the multiplication operation comes direct.

And so, the 'Possible Importers' Matrix is defined by (2.3.12).

$$\begin{aligned} \mathrm{P_{imp}} &= \text{Possible Importers} \qquad (2.3.12) \\ &= \mathrm{N^T E} \circ (\mathrm{M}[\mathrm{a_{mn}}]), \qquad \mathrm{m} = 1; \mathrm{n} = 1,2,\dots,\mathrm{N_f}; \mathrm{a_{mn}} \in \{1\} \end{aligned}$$

Again, consider the computation of the 'Possible Importers' Matrix in Figure 33 that once more relates to the working example in this section.

$$
\begin{aligned}
\mathrm{P_{imp}} &= \left(\begin{matrix} & \mathrm{S_1} & \mathrm{S_2} & \mathrm{S_3} \\ \mathrm{S_1} & 0 & 1 & 0 \\ \mathrm{S_2} & 0 & 0 & 0 \\ \mathrm{S_3} & 1 & 1 & 0 \end{matrix} \begin{matrix} & \mathrm{F_1} & \mathrm{F_2} \\ \mathrm{S_1} & 1 & 0 \\ \mathrm{S_2} & 1 & 0 \\ \mathrm{S_3} & 0 & 1 \end{matrix} \right) \circ \left(\begin{matrix} \mathrm{S_1} & 0 \\ \mathrm{S_2} & 1 \\ \mathrm{S_3} & 1 \end{matrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \\
&= \begin{matrix} & \mathrm{F_1} & \mathrm{F_2} \\ \mathrm{S_1} & 1 & 0 \\ \mathrm{S_2} & 0 & 0 \\ \mathrm{S_3} & 2 & 0 \end{matrix} \circ \left(\begin{matrix} \mathrm{S_1} & 0 \\ \mathrm{S_2} & 1 \\ \mathrm{S_3} & 1 \end{matrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \\
&= \begin{matrix} & \mathrm{F_1} & \mathrm{F_2} \\ \mathrm{S_1} & 1 & 0 \\ \mathrm{S_2} & 0 & 0 \\ \mathrm{S_3} & 2 & 0 \end{matrix} \circ \begin{matrix} \mathrm{S_1} & 0 & 0 \\ \mathrm{S_2} & 1 & 1 \\ \mathrm{S_3} & 1 & 1 \end{matrix} \\
&= \begin{matrix} & \mathrm{F_1} & \mathrm{F_2} \\ \mathrm{S_1} & 0 & 0 \\ \mathrm{S_2} & 0 & 0 \\ \mathrm{S_3} & 2 & 0 \end{matrix}
\end{aligned}
$$

Figure 33: Example of 'Possible Importers' Matrix

One very important thing to retain from the resulting 'Possible Importer' (and also from 'Possible Exporters') matrix is that the result is not only '0' or '1'. In fact each entry in the matrix represents the number of systems that gain interoperability by implementing such an importer (or exporter). This is a key element to consider when deciding which importer or exporter should be implemented in order to maximise interoperability.

The final step is to choose the best artefact – whether exporter or importer – that improves interoperability most at a given state of an interoperability scenario. To choose the interoperability artefact to be implemented, the first step is therefore to rank all the possibilities at hand. A list is thus created, by merging possible importers and possible exporters, ordered by the number of interoperations gained if that artefact is used. Table 1 represents such a list for the example set in this section.

Rank	*Imp/Exp*	*# Interoperations Gained*
1st	$\mathrm{e_{S_2,F_2}}$	2
1st	$\mathrm{i_{S_3,F_1}}$	2

Table 1: Ranking of 'Possible Importers' and 'Possible Exporters' by # Interoperations Gained

Then, out of the ordered list, the artefact that maximises interoperability is chosen, i.e. the one that establishes the most interoperations in the interoperability scenario. However, a tie between various options can occur and, in such a case, there has to be tiebreak based on the following strategy:

1. The first condition to choose the best option is to select the artefact that uses a data format that is already most present in the interoperability scenario. In the long term, this can provide better interoperability possibilities, since that format will be used more times in the future, due to this choice. So, in order to choose one artefact, the list of tied options must again be sorted by the number of importer/exporters of the same data format existing in the scenario.

   The list of existing artefacts of one same format is achieved by counting every importer and exporter of a specific data format. This is done by using (2.3.13), which basically sums up both importers and exporters of a given data format X for all the systems within the interoperability scenario.

$$\mathrm{count}(\mathrm{F_x}) = \mathrm{Count\ Importer/Exporter\ per\ Format\ X} \qquad \textit{(2.3.13)}$$
$$= \sum_{k=1}^{N_S} (i_{kx} + e_{kx})$$

   And then, by applying (2.3.13) to those data formats that are present in the artefacts that are tied in the list, one obtains the new rank of artefacts for improving interoperability in an interoperability scenario. Table 2 presents this new rank for our working example.

Rank	*Imp/Exp*	*# Imp/Exp Same Format*
1st	e_{S_2,F_2}	$\mathrm{count}(F_2) = 3$
2nd	i_{S_3,F_1}	$\mathrm{count}(F_1) = 2$

Table 2: Ranking of Importers/Exporters by # Imp/Exp Same Format

2. Being the case that all the options give exactly the same improvements on interoperability, any one of them can be chosen, since the impact is the same. So, in this situation, a random choice is made to select the interoperability artefact to be implemented.

Finally, and once the interoperability artefact (importer or exporter) to implement has been selected, such a decision is reflected in the interoperability scenario. That is to say, that the selected data exporter or data importer is now implemented in the designated system as determined by the algorithm for reaching out for the maximum possible interoperability in the interoperability scenario.

After making the changes to the interoperability scenario, a new decision cycle then takes place – of what importer/exporter to implement next. This means that the algorithm is started again but now considering the new (now current) state of the interoperability scenario.

The interoperability scenario achieves maximum interoperability when no suggestion of importer/exporter is presented, that is, when the suggestion matrices are empty.

3

Problem: Large-Scale P2P Interoperability

"A problem well stated is a problem half-solved"

– Charles Kettering, inventor and engineer

The understanding that the number of information systems that have interest in communicating with others has a natural tendency to increase is not new. It can concern, for instance, systems that want to access some resource, or are interested in a particular kind of information inside a specific interoperability scenario; or systems that are interested in the opposite, i.e. in providing the systems that compose an interoperability scenario access to new resources and/or information. This tendency of information systems results in an increase in the size of interoperability scenarios, imposing a scalability problem. This issue assumes yet another level, when the systems within a scenario present a heterogeneous nature, becoming large-scale interoperability scenarios.

This chapter is focused on studying the behaviour of the Point-to-Point Interoperability approach in large-scale interoperability scenarios. This study is performed by resorting to simulations of the Algorithm to Maximum Possible P2P Interoperability in different scenarios. To analyse these simulations, some Interoperability Metrics need to be defined, allowing the comparison of results, and focusing on the scope of the analysis.

3.1 Interoperability Metrics

The interoperability metrics must address two aspects of interoperability systems: (1) the amount of interoperability artefacts used in the scenario in order to enable interoperability between systems participating in the scenario; and (2) the ratio between the number of systems that are interoperable, and the theoretical maximum number of systems that could be interoperable given the conditions that characterise the interoperability scenario.

3.1.1 Interoperability Coverage

The Number of Interoperable Systems (NIS) is a relevant parameter of a specific scenario because it represents the current interoperability state of that scenario. It corresponds to the number of systems that are interoperable, as represented in *(3.1.1)*.

$$\begin{aligned} \mathrm{NIS} &= \text{Number of Interoperable Systems} \\ &= \sum_{p=1}^{N_s} \sum_{c=1}^{N_s} X_{(S_p, S_c)} \end{aligned} \qquad (3.1.1)$$

However, the NIS by itself is not enough to draw conclusions about the interoperability state of a scenario, as the same NIS can be considered high or low depending, e.g., on the number of systems that compose the scenario. Therefore, each NIS value must be normalized by using a scenario-dependent value, in order to allow conclusions to be drawn. The parameter chosen to assume this role is the Maximum Number of Interoperable Systems (MNIS). MNIS corresponds to the theoretical maximum number of interoperable systems that a scenario can have.

The MNIS value depends on the purpose of the systems, i.e. on the amount of systems that are only data providers, the amount of systems that are only data consumers, and the amount of systems that consume and provide data. This dependence on the systems' purpose is due to the fact that interoperability is directed.

$$\begin{aligned} \mathrm{COV} &= \text{Interoperability Coverage} \\ &= \frac{\mathrm{NIS}}{\mathrm{MNIS}} \end{aligned} \qquad (3.1.2)$$

The ratio between these two parameters – NIS and MNIS – corresponds to the Interoperability Coverage, as represented in (3.1.2). This coverage value represents the interoperability state of an interoperability scenario, as it indicates the portion of the total number of systems that are currently interoperable.

Despite disclosing the interoperability state of a scenario based on the systems' purpose, Interoperability Coverage does not give the notion of the percentage of systems in the scenario that are interoperable. To achieve that, the NIS must be normalised by using a value that corresponds to the state where all systems are interoperable – $MNIS_{abs}$. The Interoperability Absolute Coverage value can be calculated through the application of the formula used to determine the number of edges in a directed and fully connected graph from the Graph Theory, as represented in (3.1.3). The Interoperability Absolute Coverage is computed as indicated in (3.1.4).

$$\mathrm{MNIS_{abs}} = \mathrm{N_s} \times (\mathrm{N_s} - 1) \qquad \textit{(3.1.3)}$$

$$\begin{aligned}\mathrm{COV_{abs}} &= \text{Interoperability Absolute Coverage} \\ &= \frac{\mathrm{NIS}}{\mathrm{MNIS_{abs}}}\end{aligned} \qquad \textit{(3.1.4)}$$

3.1.2 Number of Interoperability Artefacts

By themselves, the coverage values are not enough to assess the interoperability in a scenario, because although they indicate the current state of the interoperability in that scenario, they do not reflect the amount of modifications introduced in the scenario to achieve that interoperability state. These modifications consist in the deployment of new exporters and/or importers, which are Interoperability Artefacts (IA). Hence, the Number of Interoperability Artefacts (NIA) deployed in a scenario, in order to obtain a specific Interoperability Coverage, is a good metric to allow the analysis and comparison of different scenarios. The NIA is determined by counting the number of Importers and Exporters in each system, as represented in (3.1.5).

$$\begin{aligned}\mathrm{NIA} &= \text{Number of Interoperability Artefacts} \\ &= \sum_{\mathrm{i}=1}^{\mathrm{N_s}} |\mathrm{S_i}|\end{aligned} \qquad \textit{(3.1.5)}$$

3.2 Study of P2P Interoperability

The study of P2P Interoperability consists of the generation of the possible interoperability scenarios that can occur in a real environment and the study of the metrics that characterise the interoperability of the environment in such conditions.

For this simulation to take place, a set of inputs is needed and must be cycled, so that the metrics are obtained. First and foremost, the interoperability scenario consists of a set of systems, so the first simulation parameter is the number of systems in the environment – N_S.

Continuing to explain P2P Interoperability, the next definition is the data format. To determine the number of different data formats existing in the environment, the percentage of data formats is defined – $\%_F$. This percentage means that the $\%_F$ of the existing systems have the same data format. For example, if the percentage is 25%, it means that – at maximum – 25% of the systems have the same data format, which roughly means that, in a total number of 8 systems, there are 4 different data formats, each one distributed by every 2 systems.

The next step concerns the capabilities of each system. Each system is described by its capability of exporting, importing or both. So, the next definition is the amount of systems in the environment that have that capability. For that, two definitions are presented: the percentage of systems that Export – $\%_E$, and the percentage of systems that Import - $\%_I$. The definition of systems that have both the capability of exporting and importing is achieved by the overlapping of the percentages, as, for example, in the case of both above 50%.

Next, and last, there is the modifiability of systems. Like the previous definitions, modifiability is also a percentage – $\%_M$ – and it represents the percentage of all systems that are modifiable, allowing their interoperability capabilities to change.

The same combination of these five interoperability scenario characteristics can result in very different conjugations of systems with different capabilities. For example, if only two data formats are used in a 10-system environment, with 50% for importers and exporters, it may happen that all the exporter systems have one format, and all the importer systems have the other, resulting in 0% of coverage. Or it may happen that the two data formats are equally distributed over the importers and exporters, resulting, at the start, in 50% of coverage.

Based on this, random distribution is needed, in which all the possible scenarios are contemplated, so that the simulation reflects a real scenario. The method for this is the Monte Carlo Simulation (Mooney, 1997). In its pure mathematical form, the method consists in finding the definite integral of a function by choosing a large number of independent-variable samples at random from within an interval or region, averaging the resulting dependent-variable values, and then dividing by the span of the interval or the size of the region over which the random samples were chosen. This differs from the classical method of approximating a definite integral, in which independent-variable samples are selected at equally spaced points within an interval or region.

The application of the Monte Carlo method is, as following. The number of systems (N_S) indicates the size of the interoperability scenario, specifying the number of systems that compose the scenario. Deeply associated with this parameter is the exporter and importer rates ($\%_E$ and $\%_I$), specifying the amount of systems that have export and/or import capabilities. The number of systems times each of these rates determines the number of systems with export and import capabilities, respectively. It is important to note, as previously described, that each system can have exclusively export, exclusively import, or both capabilities at the same time.

Each exporter and importer uses a data format, so the sum of the number of different importers and exporters in the scenario gives the initial amount of data formats in the scenario. For the effect of scenario building, all the data formats are represented in an array, as shown in Figure 34. The percentage of equal data formats ($\%_F$) times the sum of importers and exporters, expresses the maximum number of repetitions of each data format, and there must be at least one data format with that number of repetitions.

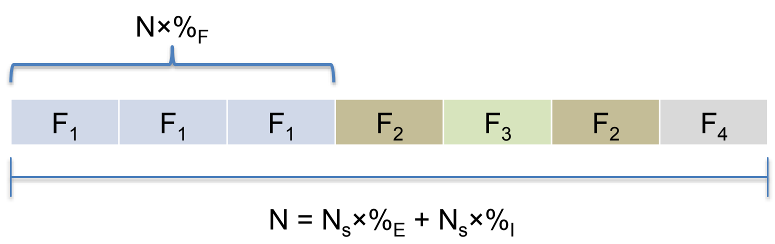


Figure 34: Data Format array used in the scenario building process

In the example displayed in Figure 34, the array of data formats has seven slots, meaning that, in this scenario, there are seven importers and exporters. In this case, it is assumed that $\%_F$ = 40%, which results in $7 \times 0.4 = 2.8 \approx 3$. This result means that the maximum number of repetitions for each data format is 3. As this number must be achieved, in order to ensure it the introduction of three representations of a data format into the array is forced. In this example, F_1 was the data format forced to achieve the 40%. The remaining slots are filled with random selected data formats, in such a way that no data format fills more than three slots.

The array of data formats is filled to aid in the construction of two matrices: (1) an Exporters Matrix, representing all the export capabilities of the scenario; and (2) an Importers Matrix, to represent all the import capabilities of the scenario. The combination of these two matrices describes the entire interoperability capabilities in an interoperability scenario, being an essential input for the algorithm used to simulate the P2P Interoperability approach.

In each matrix, systems are represented as rows, and data formats are represented as columns. Each matrix is initialized with zeros, with the zero replaced for one in the cases in which the system supports the specific data format. The data formats are randomly selected from within the array of data formats. The first matrix to be filled is the Exporters matrix. $\%_E \times N_S$ systems are, one by one, randomly selected to be associated to a data format. Each time a data format is selected, it is removed from the array of data formats.

For simplification reasons, in this phase each importer or exporter only supports one data format. Once the export matrix is filled, an import capability is assigned to all systems without exporters. This is accomplished by associating randomly selected data formats to the systems in the import matrix. The number of systems in this condition can is determined by $N_S - \%_E \times N_S$. The remaining data formats in the array are then randomly associated to any system without an import capability.

The identification of modifiable systems is another essential simulation input. This information is represented in an array where the indexes correspond to the systems represented as rows in the export and import matrices. This array is initialized with zeros and subsequently randomly populated with ones. The number of ones introduced in the array is given by $N_S \times \%_M$; values at '1' represents the modifiable systems.

With these input vectors, the simulation can now be executed, by applying the P2P algorithm to the maximum possible interoperability, so that a common base for all simulations is achieved.

The outputs of this execution are already defined in the previous sub-chapter of the metrics, and they are: (1) NIA, which is the number of interoperability artefacts needed in each scenario to achieve the maximum possible interoperability; (2) COV, which represents, for a specific scenario, the percentage to the maximum possible coverage (disregarding the modifiability of the systems, counting only with the exporters and importers) and, (3) $\mathrm{COV_{abs}}$, which is the coverage normalized to the number of existing systems, so that the change in the parameters is understood.

The variation of these outputs in relation to the inputs is studied by plotting graphics of the outputs versus the variation of the number of systems. The number of systems was chosen for the horizontal axis of the graphs, so that it depicts the impact of increase in the number of systems. And such, one graph would be needed for each variation of the other four parameters. Now, if one would consider a resolution of analysis at 1% for all the four variables, this would raises a major problem since 100 iterations of each parameter needs to be done, resulting in 100x100x100x100 = 100,000,000 graphics!

For making the number of graphics manageable, the parameters can only assume values of 25%, 50% or 75% each. This applies to all parameters: modifiability, data formats, exporters and importers. However, as every system in the environment is defined to have a purpose, either by exporting, importing or both, there is the need for all systems to have at least one importer or exporter, resulting in a restriction in the combined values of exporters and importers. The conjugation of the two needs to be at least 100%, resulting in a smaller set of valid combinations: {(E:25%,I:75%); (E:50%,I:50%); (E:50%,I:75%); (E:75%,I:25%); (E:75%,I:50%); (E:75%,I:75%)}.

By removing the set of invalid combination parameters, a much manageable number of graphics is achieved: 54. Now, the only things missing are the variation of the number of systems, defined as 100 systems, which is already quite far-fetched, because one hundred systems in a scenario already seems to be a very big number to achieve, reflecting perfectly the long-scale analysis being done. However, the number of systems was adjusted to 115, due only to the improvement of graphic presentation.

The Monte Carlo simulation requires that the number of iterations of the same execution be defined, so that the randomness of the construction of the test vectors does not influence the results. So a study was carried out, to understand the number of needed iterations to plot the graphics. This study consisted in repeating the simulations of the same scenario (with the exact same conditions) for 1000 times, and plotting the results of the NIA cumulative mean, in order to retrieve the iteration number where the NIA cumulative mean stabilizes.

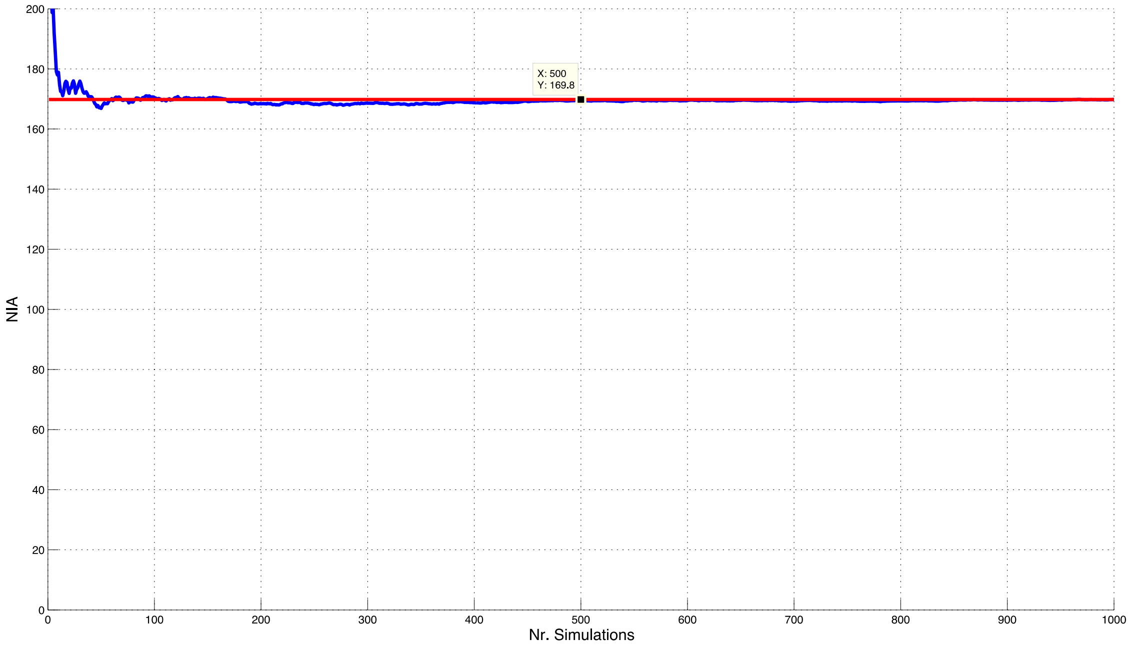


Figure 35: NIA Cumulative mean

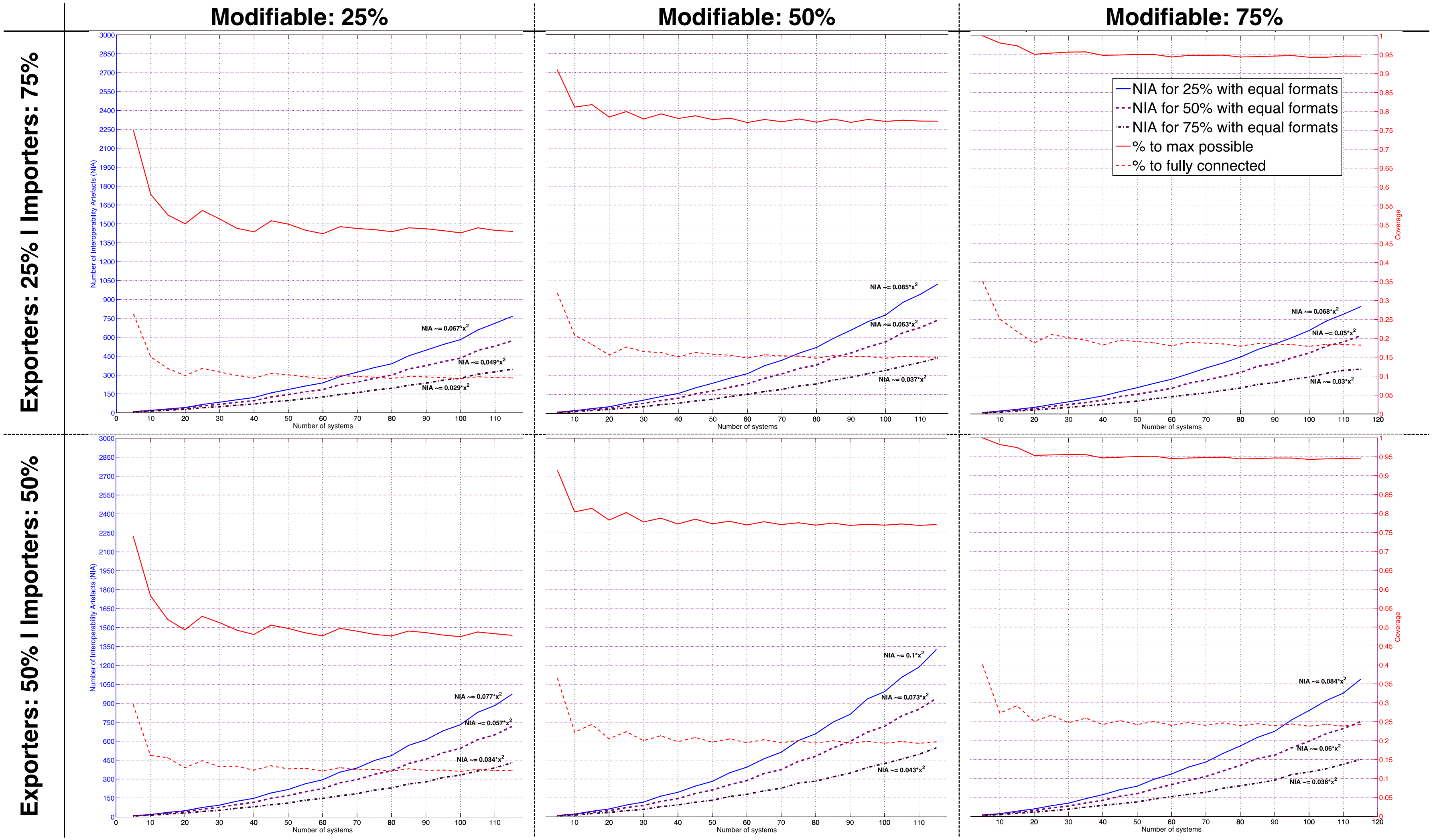
Analysis of this graphic, present in Figure 35, shows that every scenario is iterated 500 times, so that the randomness of the simulations is withdrawn from the result, thus representing a real-world environment example, so to get mean full values from the simulations, each scenario will be executed 500 times, getting 500 different distribution of properties, ensuring a real simulation scenario.

But, even so, these 54 graphics are still a lot of graphics to analyse. Each graph is presented in the appendices, and they show the maximum, the minimum and the average values of the 500 iterations of the simulation, for each simulated scenario.

Looking at the plotted graphics, the first reduction that can be done on the number of graphics is to eliminate two sets of graphics regarding the combination of exporters and importers. In the simulations, the pair of graphics {(E:25%,I:75%);(E:75%,I:25%)} and the pair {(E:50%,I:75%);(E:75%,I:50%)} are seen to have exactly the same results. This is because P2P interoperability only relies on the number of connections between exporter-to-importer pairs, and in these cases they are the same. This means that only one of each pair is needed, resulting thus in a total of 36 graphics.

In order to compare them all, the 36 graphics would need to be placed on one same surface, side by side, at the same time (like the big walls existing in UNINOVA with printed graphics), and even then it is still difficult to analyse the "full picture". However there is still one possible aggregation. In terms of behaviour, it can be noticed that the coverage does not depend on the number of different systems in the scenarios.

So, a final aggregation of the graphs can be accomplished by plotting the three different variations of the different formats on the same graph with the respective coverage. Thus, it was possible to aggregate the graphs that had the same coverage, representing in a single graph the NIA for the three values of different formats (25%, 50% and 75%), along with the coverage that they achieve. This final aggregation resulted in "only" twelve graphics, presented in *Figure 36*.


Modifiable: 25%
Modifiable: 50%
Modifiable: 75%
Exporters: 25% I Importers: 75%
Exporters: 50% I Importers: 50%
Number of Interoperability Artefacts (NIA)
Number of systems
Coverage
NIA for 25% with equal formats
NIA for 50% with equal formats
NIA for 75% with equal formats
% to max possible
% to fully connected
NIA ~= 0.067*x^2
NIA ~= 0.049*x^2
NIA ~= 0.029*x^2
NIA ~= 0.085*x^2
NIA ~= 0.063*x^2
NIA ~= 0.037*x^2
NIA ~= 0.068*x^2
NIA ~= 0.05*x^2
NIA ~= 0.03*x^2
NIA ~= 0.077*x^2
NIA ~= 0.057*x^2
NIA ~= 0.034*x^2
NIA ~= 0.1*x^2
NIA ~= 0.073*x^2
NIA ~= 0.043*x^2
NIA ~= 0.084*x^2
NIA ~= 0.06*x^2
NIA ~= 0.036*x^2

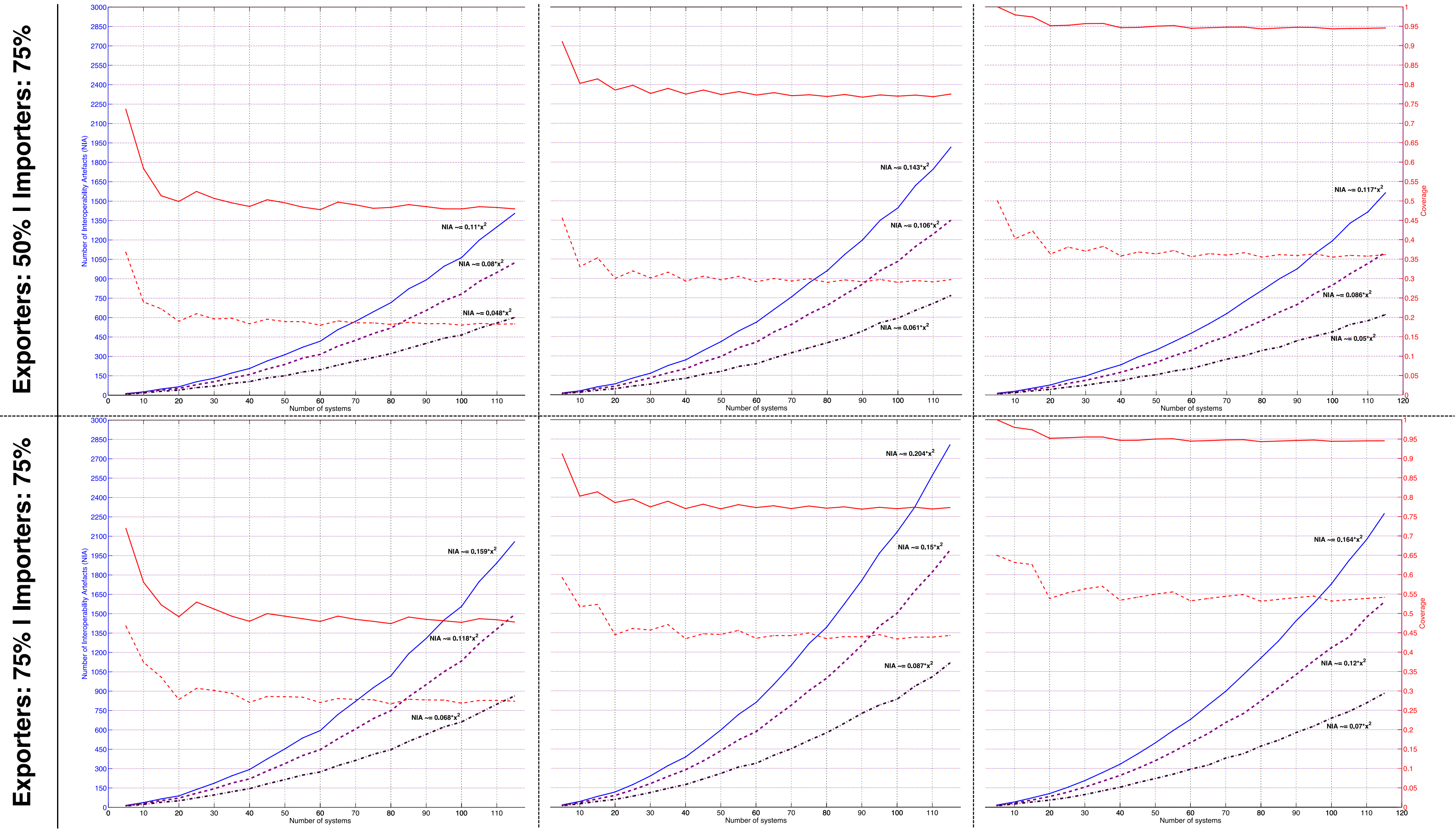


Figure 36: P2P Interoperability Simulations Result Graphs

3.2.1 Coverage vs. Modifiability

By looking at the “% to maximum possible” line in all graphics of Figure 36, which is replicated through the columns, one can observe that it depends only on the modifiability characteristic of the systems. In order to demonstrate its influence, three new graphics were plotted (presented in Figure 37) but this time grouping the simulation output by the modifiability parameter (25%, 50% and 75%).

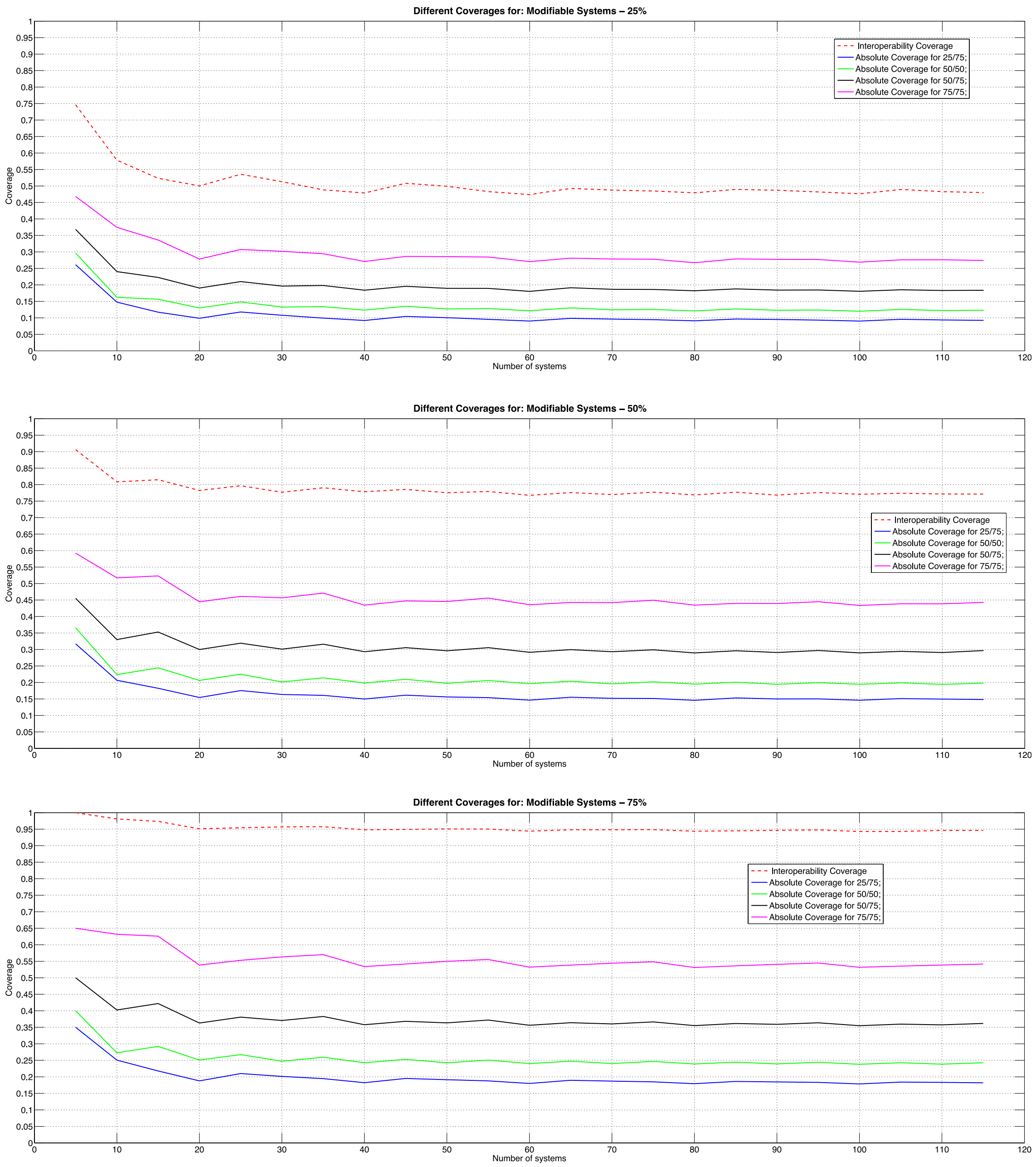


Figure 37: Coverage grouped by modifiability characteristic

The first and most important observation to acquire through the graphics is the dependability of the '% to maximum possible' line on the modifiability characteristics of the systems. The conclusion that can be drawn is that the possibility to achieve full interoperability between every system in the environment is proportionally related to the modifiability of the systems.

It can also be observed that the absolute coverage, which depends on the number of connected exporters and importers, also increases and decreases proportionally to the modifiability characteristic. Proportionality is the relation between the modifiability percentages of the studied systems and the coverage that it achieves. To study this proportionality, new simulations were devised and executed. Simulations comprised analysis of varying '% of Modifiable Systems' in 5% intervals, different 'Number of Systems' and other variables fixed: 50% equal formats, E:25% and I:75%.

Observing the graphics in Figure 37, it is noticeable that the number of systems is not relevant for the behaviour of the modifiability of the systems vs. the coverage, so this new relation is established by using the average of all the different number of systems, all in the "stable" zone of the graph, where all of them tend towards the same value.

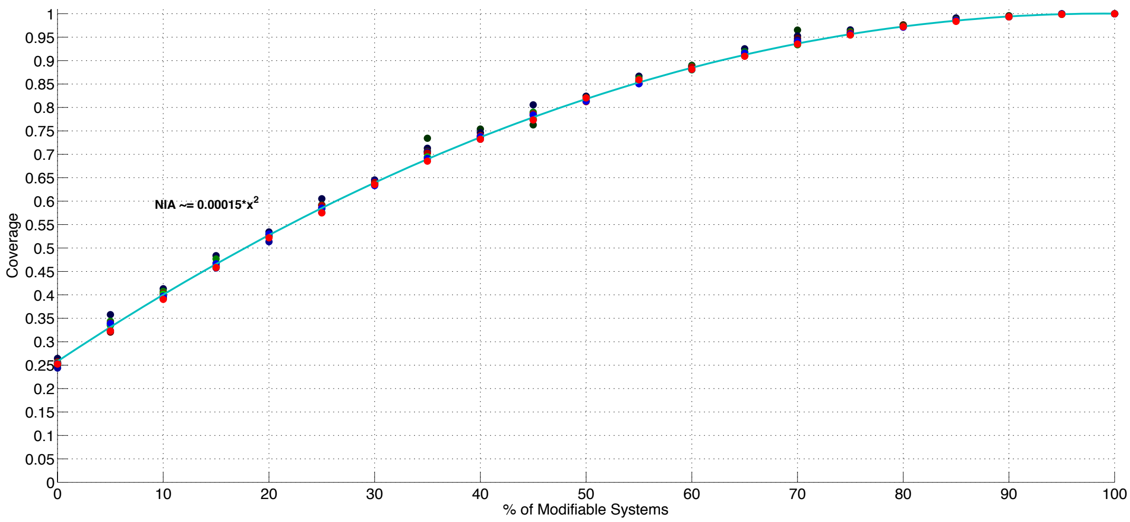


Figure 38: Modifiable Systems (%) vs. Coverage

The graphic presented in Figure 38 shows that, independently from the number of systems analysed, the percentage of the modifiability of the systems has a quadratic relation to the coverage. So, it can be concluded that when the number of modifiable systems rises, the coverage tends to fully connect.

3.2.2 NIA vs. Number Connections

The first conclusion drawn from the NIA on all graphics is about its behaviour: every NIA vs. the number of systems has a quadratic behaviour. Based on that, *Figure 36* shows, for each NIA plot, the approximate quadratic equation that accompanies the results of the simulations. The quadratic equation is defined as follows:

$$\mathrm{a}x^2 + \mathrm{b}x + \mathrm{c} = 0 \qquad \textit{(3.2.1)}$$

For the approximation made over the simulation results, one discarded the constant (c) and the linear coefficient (b), only approximating the quadratic coefficient (a). The reason behind this choice is that it is possible to relate all the simulations to one another, in order to understand the impact of each parameter on the results, so the approximation made is the one presented in the following equation:

$$\mathrm{a}x^2 = 0 \qquad \textit{(3.2.2)}$$

So the starting point for the NIA analysis is all the coefficients, so that relations between them can be understood. For such an analysis, a table of coefficients is presented, where the columns represent the Modifiable variability, the rows the Exporters/Importers relations, and the shades of blue show the variation of the percentage of equal formats in the environment:

		M:25%	*M:50%*	*M:75%*	
E:25%	*I:75%*	**0,067**	**0,085**	**0,068**	*F:25%*
		0,049	**0,063**	**0,050**	*F:50%*
		0,029	**0,037**	**0,030**	*F:75%*
E:50%	*I:50%*	**0,077**	**0,100**	**0,084**	
		0,057	**0,073**	**0,060**	
		0,034	**0,043**	**0,036**	
E:50%	*I:75%*	**0,110**	**0,143**	**0,117**	
		0,080	**0,106**	**0,086**	
		0,048	**0,061**	**0,050**	
E:75%	*I:75%*	**0,159**	**0,204**	**0,164**	
		0,118	**0,150**	**0,120**	
		0,068	**0,087**	**0,070**	

Table 3: P2P Quadratic Coefficients of NIA function approximation

This table of the Quadratic Coefficients is used as the base of the study of the relations between the parameters that comprise the study. By exploring the evolution of these parameters in the simulations, three main courses are studied: (1) Percentage of Equal Formats, (2) Percentage of Modifiable Systems, and (3) Percentage of Exporters and Importers:

1. Percentage of Equal Formats: Each one of the graphics has the plot of three different behaviours by the different application in the same conditions over a different number of formats, so it is the first study to show whether, regardless of the other conditions, the relations between these parameters is the same.

2. Percentage of Modifiable Systems: In the quadratic coefficient table, there are some variations of the parameters, and the columns represent the change that modifiability implies in the interoperability scenario, so this study will see if it is independent of the other parameters.

3. Percentage of Exporters and Importers: The last axis of evolution is the graph by lines, where the combination of Exporters and Importers is increased, so as to study if it is independent of the others. The Exporters and Importers were combined because together they determine the number of systems to be connected, and therefore interoperable.

Using these three variations in the parameters, a study is carried out, so a detailed analysis of each one comes next. All of the studies use the previously presented table of the quadratic coefficients as their baseline, and will present an individual analysis of each one. At the end, a comprehensive analysis of the combination of all these parameters is conducted, so that the behaviour of the Number of Interoperability Artefacts (NIA) can be understood for this type of interoperability solution, such as the one under study here - the Point-to-Point Interoperability.

3.2.2.1 Percentage of Equal Formats

As stated before, the variation of the percentage of equal formats is represented by the different shades of blue. So, to understand if, regardless of the others parameters, the quadratic coefficient always has the same behaviour, meaning it is an independent variable, a relation between the baseline (light blue, where the percentage of equal formats is 75%) and the other conditions (medium blue, percentage equals 50%, and dark blue, 25%) is presented in two shades of green.

		M:25%		*M:50%*		*M:75%*	
E:25%	*I:75%*	**0,067**	2,310	**0,085**	2,297	**0,068**	2,267
		0,049	1,690	**0,063**	1,703	**0,050**	1,667
		0,029		**0,037**		**0,030**	
E:50%	*I:50%*	**0,077**	2,265	**0,100**	2,326	**0,084**	2,333
		0,057	1,676	**0,073**	1,698	**0,060**	1,667
		0,034		**0,043**		**0,036**	
E:50%	*I:75%*	**0,110**	2,292	**0,143**	2,344	**0,117**	2,340
		0,080	1,667	**0,106**	1,738	**0,086**	1,720
		0,048		**0,061**		**0,050**	
E:75%	*I:75%*	**0,159**	2,338	**0,204**	2,345	**0,164**	2,343
		0,118	1,735	**0,150**	1,724	**0,120**	1,714
		0,068		**0,087**		**0,070**	

F:25%
F:50%
F:75%
F: 75% → 25%
F: 75% → 50%

Table 4: Relations of Quadratic Coefficient for Equal Formats %

From the table, all the values for the same relations are seen to have similar values. This is due to the fact that approximations were used, so, to understand if all of them converge to the same values, a statistical analysis is performed:

Relation	*Average*	*Minimum*		*Maximum*	
		Absolute	Δ	Absolute	Δ
F: 75% → F: 50%	1,700	1,667	0,033	1,738	0,038
F: 75% → F: 25%	2,317	2,265	0,052	2,345	0,028

Table 5: Statistics of Relations of Quadratic Coefficient for Equal Formats %

Table 5 shows that the deviations from the average are minimal, and this is because (as already stated) approximations were made. By considering only the quadratic coefficient, <u>one can conclude that the relation between different percentages of equal formats is always the same, regardless of the other parameters, making it an independent variable</u>.

3.2.2.2 Percentage of Modifiable Systems

Like the previous case, the Modifiability of the system is also represented on the quadratic coefficient table, but in the columns of the table, so the study of the relations between the different percentages of modifiability is done column to column. The first column is the baseline for the comparison, and the other two are compared to the first, that comparison being represented in shades of orange.

		M:25%	*M:50%*	*M:75%*
E:25%	*I:75%*	**0,067**	**0,085**	**0,068**
			1,269	1,015
		0,049	**0,063**	**0,050**
			1,286	1,020
		0,029	**0,037**	**0,030**
			1,276	1,034
E:50%	*I:50%*	**0,077**	**0,100**	**0,084**
			1,299	1,091
		0,057	**0,073**	**0,060**
			1,281	1,053
		0,034	**0,043**	**0,036**
			1,265	1,059
E:50%	*I:75%*	**0,110**	**0,143**	**0,117**
			1,300	1,064
		0,080	**0,106**	**0,086**
			1,325	1,075
		0,048	**0,061**	**0,050**
			1,271	1,042
E:75%	*I:75%*	**0,159**	**0,204**	**0,164**
			1,283	1,031
		0,118	**0,150**	**0,120**
			1,271	1,017
		0,068	**0,087**	**0,070**
			1,279	1,029

F:25%
F:50%
F:75%
M: 25% → 50%
M: 25% → 75%

Table 6: Relations of Quadratic Coefficient for Modifiability %

As in the previous case, Table 6 shows that, regardless of the other parameters, the modifiability parameters always have the same relation to the baseline quadratic coefficient, so statistical analyses are carried out (Table 7):

Relation	*Average*	*Minimum*		*Maximum*	
		Absolute	Δ	Absolute	Δ
M: 25% → M: 50%	1,284	1,265	0,019	1,325	0,041
M: 25% → M: 75%	1,044	1,015	0,029	1,091	0,047

Table 7: Statistics of Relations of Quadratic Coefficient for Modifiability %

From this analysis, and looking specifically at the deviations (Δ), all the behaviours are seen to be coherent, because they have a minimum deviation. So, it can be concluded that they have a variation that is independent of the other parameters.

Something else can be seen from this analysis, which is the fact that the multiply factor from the baseline does not increase; in fact, it goes up on the 50% and then decreases. This is because the P2P algorithm always tries to maximise the maximum number of possible interoperability, for each NIA, by using the more prominent format on the environment. The problem is that when there are not many systems to modify (like the case of 50%), more changes on the modifiable systems could be needed, in order to adapt to the non-modifiable systems. In the cases close to the full modifiable environment, such as the 75%, there are less non-modifiable systems to ensure interoperability, so the environment will tend faster to a more prominent format, thus giving a lower number of interoperability artefacts.

3.2.2.3 Percentage of Exporters and Importers

The lines show the evolution of the combination of exporters and importers, so, as in the other analysis, the relation to the baseline will be made, to study the independence of this parameter. This analysis is done in a combined manner, because these two parameters are related. The number of pairs of Exporters and Importers is the same, regardless of whether there are 2 systems to export to 1 system or 1 to 2 systems – in both cases two connections are needed.

		M:25%		*M:50%*		*M:75%*		
E:25%	*I:75%*	**0,067**		**0,085**		**0,068**		*F:25%*
		0,049		**0,063**		**0,050**		*F:50%*
		0,029		**0,037**		**0,030**		*F:75%*
E:50%	*I:50%*	**0,077**	1,149	**0,100**	1,176	**0,084**	1,235	E:25% I:75% → E:50% I:50%
		0,057	1,163	**0,073**	1,159	**0,060**	1,200	
		0,034	1,172	**0,043**	1,162	**0,036**	1,200	
E:50%	*I:75%*	**0,110**	1,642	**0,143**	1,682	**0,117**	1,721	E:25% I:75% → E:50% I:75%
		0,080	1,633	**0,106**	1,683	**0,086**	1,720	
		0,048	1,655	**0,061**	1,649	**0,050**	1,667	
E:75%	*I:75%*	**0,159**	2,373	**0,204**	2,400	**0,164**	2,412	E:25% I:75% → E:75% I:75%
		0,118	2,408	**0,150**	2,381	**0,120**	2,400	
		0,068	2,345	**0,087**	2,351	**0,070**	2,333	

Table 8: Relations of Quadratic Coefficient for Exporters/Importers %

On the table, it is possible to see that the analysis (as stated before) is done row by row, aggregated by the Modifiability and the Number of Formats. The relations extracted from the initial table show that the relation is consistent for all the values. To understand the deviations, and how accurate the relation is regarding all the cases, a statistical analysis is conducted and provided in the following table.

Relation			*Average*	*Minimum*		*Maximum*	
				Absolute	Δ	Absolute	Δ
E:25% I:75%	→	E:50% I:50%	1,180	1,149	0,031	1,235	0,055
E:25% I:75%	→	E:50% I:75%	1,672	1,633	0,039	1,721	0,049
E:25% I:75%	→	E:75% I:75%	2,378	2,333	0,045	2,412	0,034

Table 9: Statistics of Relations of Quadratic Coefficient for Exporters/Importers %

Table 9 shows that all the calculated relations are within an average with a small deviation, therefore, the conclusion that this parameter is also independent from all the others can be drawn.

3.2.2.4 Conclusion

All of the Point-to-Point simulation results were approximated by a quadratic function, more specifically, approximated to the coefficient that controls the speed of increase. This quadratic behaviour shows that when the number of systems in an interoperability scenario increases, the number of needed interoperability artefacts (NIA) also increases at a fast rate. In the following picture, all of the approximations are shown together, where the range of NIAs can be viewed.

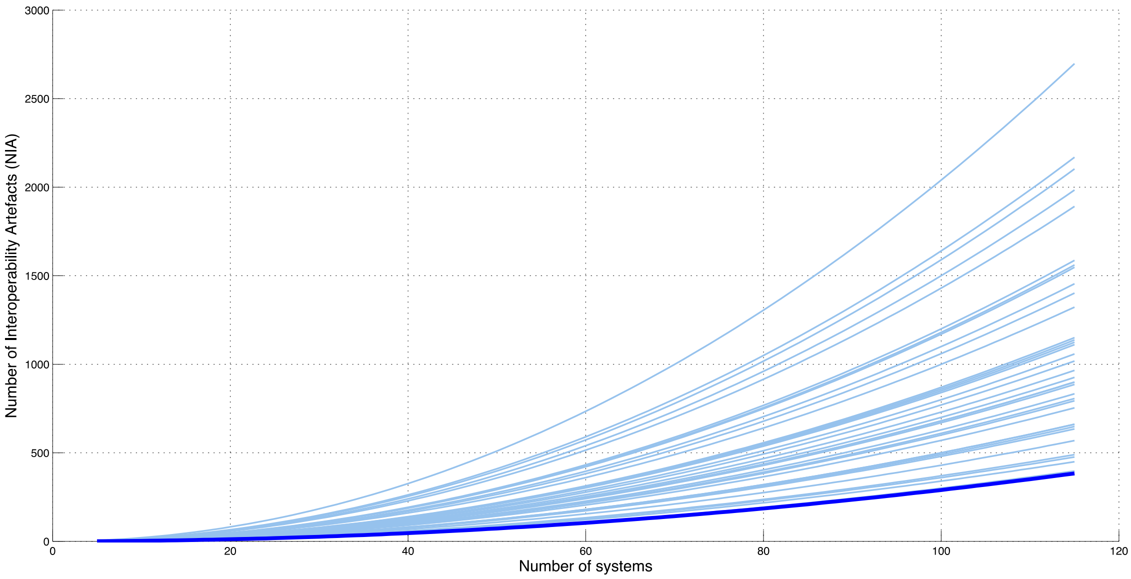


Figure 39: P2P NIA-on-a-graphic based on quadratic approximations

In the graphic, the quadratic baseline approximation is shown in a brighter blue, which is the scenario where the combination of parameters result in the smaller NIA number (M:25%, F:75%, E:25%, I:75%) of the simulations made. And all the plots of the other combinations are represented, showing the increment of the speed of increase through the evolution of the scenarios, revealing that there is a wide range of NIA through the use of the P2P Interoperability approach.

The individual analysis of each parameter variation proved that each one is independent of the others in relation to the baseline. The following table shows the evolution of each parameter regarding the baseline.

Relation	*Average*	*Minimum*		*Maximum*	
		Absolute	Δ	Absolute	Δ
F: 75% → F: 50%	1,700	1,667	0,033	1,738	0,038
F: 75% → F: 25%	2,317	2,265	0,052	2,345	0,028
M: 25% → M: 50%	1,284	1,265	0,019	1,325	0,041
M: 25% → M: 75%	1,044	1,015	0,029	1,091	0,047
E:25% I:75% → E:50% I:50%	1,180	1,149	0,030	1,235	0,056
E:25% I:75% → E:50% I:75%	1,672	1,633	0,040	1,721	0,048
E:25% I:75% → E:75% I:75%	2,378	2,333	0,045	2,412	0,034

Table 10: Statistics of Relations of Quadratic Coefficient

Table 10 shows the impact that each one of the parameters has on the speed of increase of the NIA. Looking at the table, and starting with the analysis of the Modifiable characteristic, it shows a small increase in relation to the baseline, which, as already discussed, stems from the number of modifications and, more importantly, from the impossibility of achieving full connections when there are few modifiable systems, and from the fact that it is possible to achieve a good optimisation when there are a lot of modifiable systems. So, one can conclude that the Modifiable parameter does not cause a big impact in terms of NIA (but more in respect to Coverage as seen before).

Regarding the Exporter and Importers combined parameter and the equal Formats parameter, they have a huge impact on the baseline, which is directly related to the fact that these parameters control the number of needed connections in the scenario. This is the big problem – related to NIA – of the Point-to-Point Interoperability.

3.3 Research Problem

The Point-to-Point Interoperability approach encloses two key problems: the Coverage-Modifiability dependency problem and the NIA-Connections dependency problem.

3.3.1 The Coverage-Modifiability dependency problem

The Point-to-Point Interoperability approach exhibits then a fundamental problem: the number of non-modifiable systems limits the maximum possible interoperability coverage that is achievable in an interoperability scenario. The more pairs of non-modifiable systems with disjoint sets of exporting to importing capabilities exists the least maximum possible interoperability it is possible to realise. That is to say, the bigger is the number of non-modifiable systems that are required to be interoperable (from data source system to data target system) but that do not have compatible interoperability artefacts (matching exporter in the source to an importer in the target), then more distant we will be of attaining full (possible) interoperability in the scenario.

And as such, there is a direct relation of the Modifiability variable (percentage of modifiable and non-modifiable systems) with the (interoperability) Coverage that is possible to achieve within an interoperability scenario. In fact, and as duly proven in the study hereby presented, the Interoperability Coverage decays in a quadratic proportionality as the number (percentage) of non-modifiable systems increases. This is in fact a considerable drawback for interoperability as the maximum possible data exchange possibilities within the environment is highly hindered by the presence of non-modifiable systems. Even considering interoperability scenarios with small portions of non-modifiable systems present in the environment, this present a non-neglected impact on the maximum interoperability coverage that can be made possible.

This is indeed a fundamental limitation of the Point-to-Point Interoperability approach. In the P2P Interoperability approach it is simply not possible to interoperate two non-modifiable systems that do not posses already compatible export-to-importer facilities. And the more of this exist in the scenario, the least it is possible to reach the maximum possible interoperability. So, how to go about making the interoperability Coverage not dependent of the number of non-modifiable systems present in the interoperability scenario? That is the same to say: <u>how to set the interoperability Coverage variable independent of the (percentage of) modifiable (or non-modifiable) systems variable</u>?

3.3.2 The NIA-Connections dependency problem

And P2P Interoperability approach exhibits one other crucial problem: the greater the number of required interoperations the more interoperability artefacts is required. The Number of Interoperability Artefacts (NIA) grows when it is required do interoperate a growing number of systems in interoperability scenarios. And worst, the NIA grows fast – in quadratic degree – as the number of systems increases. A very noticeable case is that of interoperating for instance just one non-modifiable system that basically implies to change all other systems to become interoperable with it, i.e. needing to make many connections from systems to the non-modifiable system. This means that there is a direct implication of the number of connections (interoperations) that need to be established, to the NIA that will make such interoperability possible.

The NIA-Connections problem implies that it might be unfeasible to interoperate an interoperability scenario, as the number of required changes is unrealisable due to the high resources required and/or the high number of parties needing to participate in the changes. This means that a given interoperability scenario could, in theory, be made interoperable, but that practically the costs and involvement makes in not possible. So, and despite the fact that it would be possible to interoperate system altogether, interoperability scenarios will not become interoperable.

The NIA-Connections condition is thus also highly restrictive for interoperability. So, how to go about making the Number of Interoperability Artefacts (NIA) not so much dependent of the number of needed connections. That is the same to say: <u>how to minimise the impact of a growing number of needed connections in the NIA variable</u>?

3.3.3 Research Question

And so, summing up, P2P interoperability approach presents these major problems: (1) the trouble of fully interoperating environments in presence of non-modifiable systems; and (2) the high "cost" of fully interoperating systems in an interoperability environment.

<u>Research Question</u>: How to achieve full (maximum possible) interoperability coverage with the least Number of Interoperability Artefacts (NIA), i.e. how to interoperate all data systems with acceptable interoperability resources ?

4

Research: Interoperability out of the Skies

"To most people, the sky is the limit.
To those who love aviation, the sky is home."
– Anonymous

Interoperability qualifies the ability of two or more systems (or components) to exchange and use the information that has been exchanged (IEEE, 1991). It defines the aptitude of entities (people, objects, systems) to communicate (say and hear) and understand (speak and listen) to each other. Data interoperability is about interrelating entities at the data level, so that information can be duly exchanged and shared between constituent entities of a data-related environment. Practically speaking, data interoperability is about establishing translations of data representations and semantics (data mappings) between entities that do not "speak the same languages".

Broadly looking, there are other environments with their own supporting systems that could relate to a data interoperability support system and where similar problems and equivalent challenges exist. After all, data interoperability support systems have to do with enabling heterogeneous entities to communicate and understand one another, and there are systems, from other domains do practice, that exhibit similar characteristics and needs. Inspecting, studying and reflecting about such comparable systems might provide important hints and good inspirations to develop new concepts and solutions to (data) interoperability.

4.1 Data Interoperability and Commercial Air Transportation

Transport & Logistics is (a broad industry sector) responsible for managing the flow of goods, information, and people between a point of origin and a point of consumption (MARICOPA, 2009). The logistics and transportation industry is highly competitive as regards providing tailored logistics and transportation solutions that ensure coordinated goods movement from source to destination through each supply chain network segment. The need for large-scale multi-modal coordination of transports & logistics operations makes it a prolific ground for best-of-breed theories and methods to handle this complexity.

A transport & logistics system connects several origins to many destinations with the objective of (potentially) interconnecting all designated sources to the required ends, doing so in the most efficient possible way. In the very same manner, a data interoperability support system is directed towards linking data sources to data consumers aiming to maximise data linkage in the environment, i.e. to connect the a maximum of data origins and data targets, in some effective and feasible form.

Also, transportation & logistics operations are conducted using many different kinds of transports and logically, not all transports can be handled at every destination; and modifying them all to support full transports is either economically impractical – prohibitive costs – or simply impossible to be done – operational non-sense. Likewise, a data interoperability support system deals with disparate data sources and dissimilar data consumers – different formats, different languages, different semantics – in a data-heterogeneous environment where not all sources can directly talk to all targets; and making them directly interoperable is either impossible – systems cannot be modified –or costly – unreasonable number of changes.

Thus, transportation & logistics systems share many of the basic needs of a system to support data interoperability. Both systems focus on the move of "things" from source to target – data interoperability support systems on the move of data and transport & logistics systems on the move of goods, information, people, etc. And both systems care about the best way to connect many origins to many destinations, optimally all – data interoperability support systems connecting many data consumers to various data sources and transport & logistics systems connecting several points of origin to lots of points of consumption.

Looking especially at commercial air transportation systems – an area in which the author has great interest and knowledge – one can see many similarities to the challenges of data interoperability support systems. Commercial air transportation has become an increasingly important mode of travel in the lives of us all, making it possible to reach out to all points of the globe in acceptable durations and conditions. This just means that the complexities of commercial air transportation systems have increased to such a state where novel operational schemes are deemed a necessity.

In air transportation systems there are many airplanes of numerous kinds and types that require many diverse and very specific conditions to be handled by airports. There are lots of airports in the world: some (main airports) capable of handling all types of airplanes: others (national/regional airports) only able to deal with some of them; and still others (local airports) able to deal with smaller aircrafts. In air transportation systems, heterogeneity is high, scale is huge, and thus well-organised solutions need to be in place for dealing with such complexity (see Figure 40).

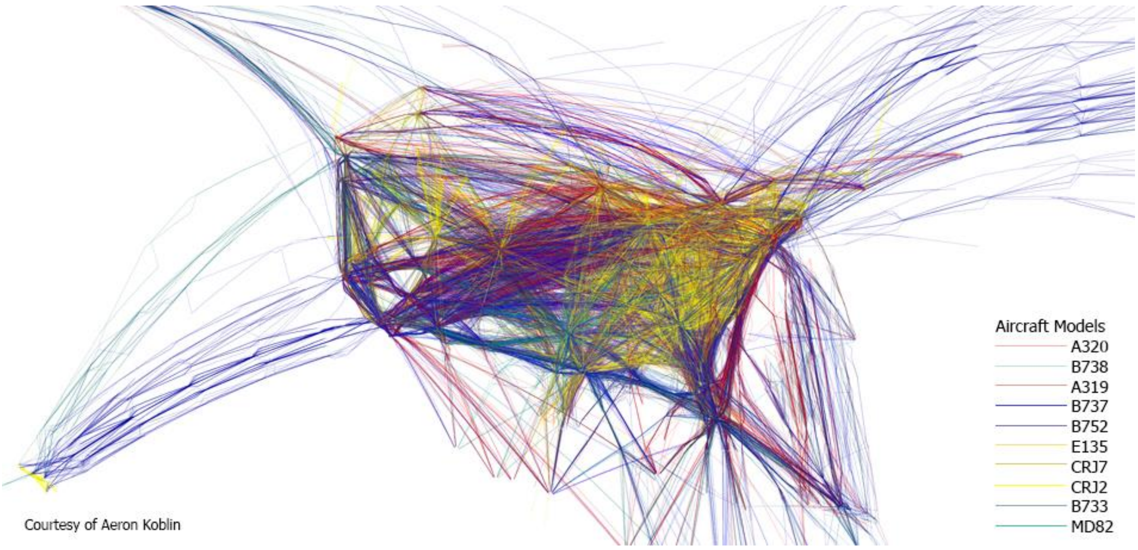


Figure 40: Flight Patterns in the United States by top-10 aircraft models on August 12th, 2008.

The commercial air transportation environment is composed of many entities (airports) whose primary interest is interconnecting with one another in order to transfer people (and cargo). Conversely, and in a given data interoperability environment, entities (systems) are interested in interoperating with each other in order to transfer data. In both – commercial air transportation and interoperability environments – constituent entities have a clear interest in interlinking with others so as to achieve their own (or common) goals – of transferring people/goods and data, respectively.

In the commercial air transportation system, while some airports are able to service all types of commercial airplanes, others are only able to service some types of aircrafts; Airports are thus capable of exporting and importing selected airliners and possibly not others. In the data interoperability domain, systems are able to export and import certain types of data formats. Similarly to the data interoperability support system, airports can be altered to add capabilities of importing/exporting non-supported airplanes, at a cost. Some airports cannot be changed at all – as either the cost is prohibitive or it simply does not make sense from an operational and service standpoint – setting it as a non-modifiable airport for certain types of aircrafts. In the very same way, in a data interoperability environment, non-modifiable systems exist typically because they are proprietary/closed/legacy systems and changing them is impossible to accomplish. Now, these entities – airports and data systems – are integral parts of their respective environments and are (potentially) required to somehow interrelate with all other designated entities – source to destination airports/systems.

In commercial air transportation, air flights, performed by designated airplanes, activate the (transport) interconnection between airports; air flights are directed routes between airports, originating at a source and directed towards a target endpoint; a specific aircraft model is "exported" from the originating airport and (after flight) is then "imported" at the destination airport. Likewise, in data interoperability environments, interoperations make the (interoperability) interconnection between systems. Interoperations represent directed interoperability routes between systems, going from a source to a target system; the interoperability artefact at the originating system exports data in a given format that is then imported by a compatible interoperability artefact at the destination system. Interestingly, not all flights are possible with any given airplane, in the very same way that not all interoperations are possible between systems, because of lacking exporting/importing capabilities in the constituent entities.

And, the goal of commercial air transportation systems is analogous to that of data interoperability systems: to maximise the interrelation of constituent entities (ultimately all designated entities) while efficiently accomplishing such interconnectivity. Via global, regional and local carriers, commercial air transportation connects all airports in the most efficient form; (transportation) efficiency meaning less resources (airplanes), but it could also signify some best operational scheme considering costumer demand, number of flight hops, passenger flow, partnerships, etc. Likewise, data interoperability support systems aim to maximise the data exchange links between constituent systems – optimally, interoperating all designated systems at the data level – in the most efficient form; (interoperability) efficiency meaning less resources (number of interoperability artefacts), but also some best operation interoperability scheme considering data-sharing demand, number of interoperations, etc.

And so obviously, the challenges of commercial air transportation systems are much similar to those of data interoperability support systems, namely:

- How to reach out for a full coverage of worldwide airports considering that airports cannot service all kinds of airplanes and cannot be modified to support them which relates to the equivalent challenge of how to interoperate all systems – especially those that cannot be modified to support (import/export) alternate data formats – on data interoperability environments;

- How to efficiently achieve interconnectivity in between airports of a commercial air transportation system using the least possible resources (airplanes), which relates to the comparable challenge of how to interoperate (at the data level) the many systems of an interoperability environment using optimal use of resources (i.e. interoperability artefacts).

So, let us observe and analyse the solutions that have been put into practice in commercial air transportation systems to address these challenges. Then, let us reflect on the possible parallels with data interoperability support systems in order to try and achieve a better-fitting idea of interoperability that can hopefully perform better than the current approaches.

4.2 *From Scissor-Hubs to Interoperability Mediation*

4.2.1 Scissor-Hub Operations

When by October 2012, Gol Transportes Aéreos (IATA: G3) announced that on 15th December it would start to operate new routes from Rio de Janeiro and São Paulo to Miami and Orlando, Florida, it also highlighted a key aspect on the arrangement of both flights to the US: airplanes would stop halfway, in Santo Domingo, at carefully synchronised arrival/departure schedules (Gol, 2012)! The strategy was to use the base in Las Américas International Airport to exchange passengers between flights.

Gol flight 7710 (G3-7710) departs from Rio de Janeiro Galeão International Airport (IATA: GIG) at 10:36 in the morning, operated by a Boeing 737-800 with registration number PR-GUH, heading to Miami International Airport (IATA: MIA), scheduled to arrive at 20:10 in the afternoon. However, Gol flight 7710 will stop first at Las Américas International Airport (IATA: SDQ) in Santo Domingo, Dominican Republic, with planned arrival at 16:40 and lift-off one hour later, at 17:40, bearing for final destination in US.

At São Paulo, Gol flight 7726 (G3-7726) is listed to depart from Congonhas Airport (IATA: CGH) at 10:40 in the morning, operated by another Boeing 737-800 with number PR-GUJ, routed for Orlando International Airport (IATA: MCO), scheduled to land at 20:39 in the afternoon. Likewise, Gol flight 7726 will first land in Santo Domingo at 16:35 only to be airborne again at 17:35, flying to Orlando (see Figure 41). And the exact same arrangement happens in both return flights.

Daily 10:36 International Rio Janeiro (GIG) 16:40 Santo Domingo (SDQ)
G3 7710 Non-stop Boeing 737-800 (738) 7:04 Valid until 2013-08-04
Daily 17:40 Santo Domingo (SDQ) 20:10 Miami (MIA)
G3 7710 Non-stop Boeing 737-800 (738) 2:30 Valid until 2013-08-04

Daily 10:40 Guarulhos, Sao Paulo (GRU) 2 16:35 Santo Domingo (SDQ)
G3 7726 Non-stop Boeing 737-800 (738) 6:55 Valid until 2013-08-04
Daily 17:35 Santo Domingo (SDQ) 20:30 Orlando (MCO)
G3 7726 Non-stop Boeing 737-800 (738) 2:55 Valid until 2013-08-04

Figure 41: (left) route and schedule of G3-7710 from Rio Janeiro to Miami via Santo Domingo; (right) route and schedule of flight G3-7726 from São Paulo to Orlando via Santo Domingo. Retrieved August 8, 2013, from FlightMapper.net (http://www.flightmapper.net)

Notably, both flights are scheduled to arrive with clockwise precision – G3-7710 at 16:40 and G3-7726 at 16:35 – at Santo Domingo, Dominican Republic. The one-hour stop allows refuelling of the two airplanes but, more importantly, passengers who embarked in Rio can also now go to Orlando, and conversely passengers who entered in São Paulo can also travel to Miami (Maia, 2012). Only passengers who are not in the “correct” airplane are moved out to the other airplane, as others remain seated while some refreshments are served. The interconnection is made in a separate terminal at Santo Domingo, avoiding customs and baggage handling, making the whole operation possible in one hour, or even less.

This clever operational procedure – of two airliners stopping halfway along their routes to interexchange passengers – makes it possible to link more destinations with the same number of airplanes. If Gol was to link Rio-Miami and São Paulo-Orlando with direct non-stop flights, it meant connecting two city-pairs with two aircrafts; By stopping both flights first at Santo Domingo, Gol manages to additionally link Rio to Orlando and São Paulo to Miami, meaning that is able to serve four city-pairs with the same two airplanes. In aviation lexicon, Santo Domingo is called a scissor-hub airport, in the sense that it hubs two inbound to two outbound flights in an open-scissor-like manner, from handles (bows) to blades interchanging at the pivot (hub airport). See Figure 42.

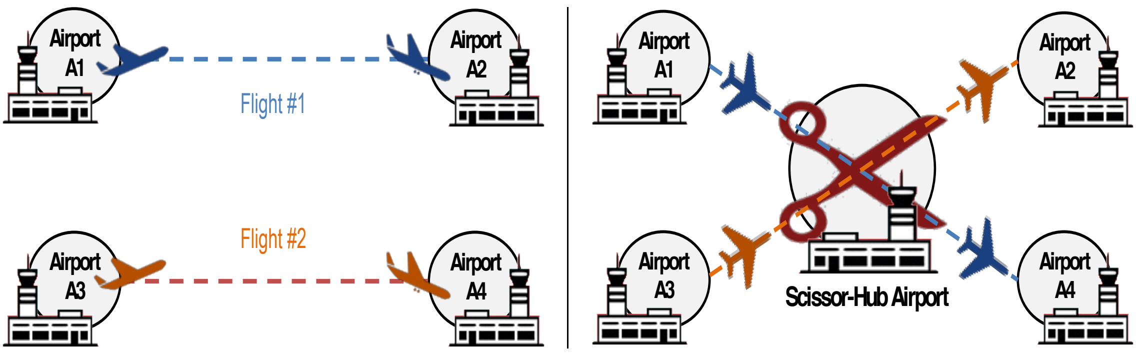


Figure 42: point-to-point connection in between two city-pairs (A-B) and (C-D) using two flights #1 and #2 (left) and (right) scissor-hub operational model able to link four city-pairs (A-B), (A-D), (C-D) and (C-B) using same two flights #1(#1.1 and #1.2) and flight #2 (#2.1 and #2.2)

Practically speaking, <u>scissor-hubs make it possible to do more (destinations/cities) with the same resources (airplanes) at the expense of one stop along the way for transhipment of self-loading cargo (air transport jargon for airplane passengers)</u>. The switching operation at a scissor-hub airport is usually very efficient and is made extremely fast in order for passengers not to consider it a drawback in their travels.

Many other air carriers use (or have used) a scissor-hub operational model, in order to achieve economical and operational competitive advantages. Jet Airways (IATA: 9W) scissor-hubs at Brussels Airport (IATA: BRU) on flights from Mumbai International Airport (IATA: BOM) and New Delhi Indira Gandhi International Airport (IATA: DEL) in India, airborne in the direction of Newark Liberty International Airport (IATA: EWR) in USA and Toronto Pearson International Airport (IATA: YYZ) in Canada, respectively. In this case, the operating jet aircraft – Airbus A330-300 with a maximum range of 6,100 nautical miles – cannot make such long distance non-stop journeys from Mumbai to Newark (6,774 nautical miles) or New Delhi to Toronto (7,240 nautical miles). This is because the aircraft has been selected as to optimise such route and not the opposite.

Notably, there are other carriers providing direct air transportation services between those city-pairs. United Airlines (IATA: UA) flies daily non-stop from Delhi to Newark – flight UA 49 – using a Boeing 777-200ER airplane that can travel 7,725 nautical miles; and Air India (IATA: AI) used to fly non-stop six times a week between New Delhi and Toronto – flight AI 187 – using a Boeing 777-300ER (Indo-Asian News Service, 2010) airliner which has a maximum range of 7,930 nautical miles. Jet Airways deliberately decided to stop halfway, at Brussels, to also give passengers coming from New Delhi the possibility of flying to Newark, and those originating from Mumbai to also travel to Toronto. And all this with no need for any additional flights between the four cities!

Moreover, a scissor-hub operation can involve more than two flights! Take Delta Airlines (IATA: DL) that drives a scissors-hub at Narita International Airport (IATA: NRT), in the Greater Tokyo Area of Japan. Delta inherited a majority of its Tokyo Narita operations in 2008 following its merger with Northwest Airlines (IATA: NW) (Bouquet, 2013). Delta links many Asian-Pacific destinations to United States airports via Tokyo Narita, using a scissor-hub operational model, i.e. inbound flights arriving within a very close timeframe, short turnaround periods basically allowing passengers to be exchanged between aircrafts, flights outbound to their destinations, and all of this under the same flight numbers while passing through Tokyo Narita; And the same thing for return flights.

Nowadays, Delta Airlines operates as many as eight flights in a scissor-hub scheme: from United States Atlanta's Hartsfield-Jackson International Airport (IATA: ATL), Detroit's Metropolitan Wayne County Airport (IATA: DTW), Los Angeles International

Airport (IATA: LAX), Minneapolis-St. Paul International Airport (IATA: MSP), John F. Kennedy International Airport (IATA: JFK), Portland International Airport (IATA: PDX), San Francisco International Airport (IATA: SFO), Seattle-Tacoma International Airport (IATA: SEA), to Asian-Pacific Bangkok's Suvarnabhumi Airport (IATA: BKK), Beijing Capital International Airport (IATA: PEK), Guam's Antonio B. Won Pat International Airport (IATA: GUM), Hong Kong International Airport (IATA: HKG), Manila's Ninoy Aquino International Airport (IATA: MNL), Shanghai's Pudong International Airport (IATA: PVG), Singapore's Changi Airport (IATA: SIN) and Taiwan Taoyuan International Airport (IATA: TPE).

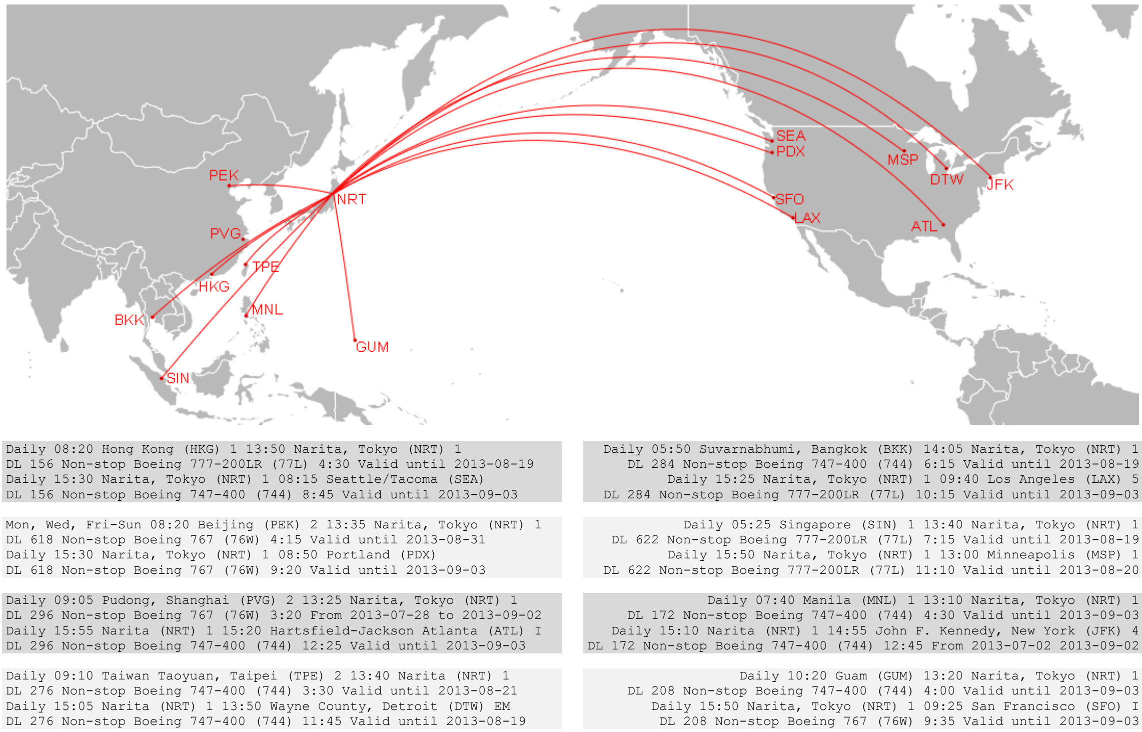


Figure 43: Overview of Delta Airlines scissor-hub at Tokyo Narita International Airport (NRT) linking together eight airports in the USA to eight airports in the Asian-Pacific region

A general definition of a scissor-hub operation is that of a closely aligned schedule of several inbound flights that arrive at one (scissor-hub) airport, transfer passengers between themselves, and get back in the air (with the same flight numbers) outbound to their final destinations. The "scissor" metaphor is used to show that the operation resembles the shape of an open scissors, with flights going from handles (bows) to blades, interchanging at the pivot (hub airport). The "scissors" image is better understood for a two-city-pairs operation but it can also be some imaginary scissors with many handles and the same number of blades, all of them coming together at the pivot. These scissors would hardly cut anything but they would describe the concept.

4.2.2 Towards Interoperability Mediation

The scissor-hub underlying principle is about bringing all airplanes to one common place (scissor-hub airport) and, from there, controlling, managing and executing the inter-crossing between flights with the purpose of offering many destinations to passengers originating from all the sources of the scissor-hub. Scissor-hub operations represent therefore a clever functional scheme that makes it possible to do more (airports) with the same resources (aircrafts). And scissor-hubs also succeed in serving different kinds of airports that can handle dissimilar classes of aircrafts. Thus, scissor-hubs act as a mediator between origin and terminus airports by settling/reconciling their differences – of varying passenger loads and aircraft support infrastructures.

Now, it might be that this mediation approach of commercial air transportation systems can also open up some opportunities to address the initially stated data interoperability problems. In the very same way as in commercial air transportation, data interoperability might see operational gains if a mediation scheme is established between data sources and data consumers. It might be that, if data interoperability could be controlled, managed and executed at one and the same place – a mediator – then it might be that, just like in commercial air transportation, one could address some of the identified challenges of data interoperability. See Figure 44 regarding this idea.

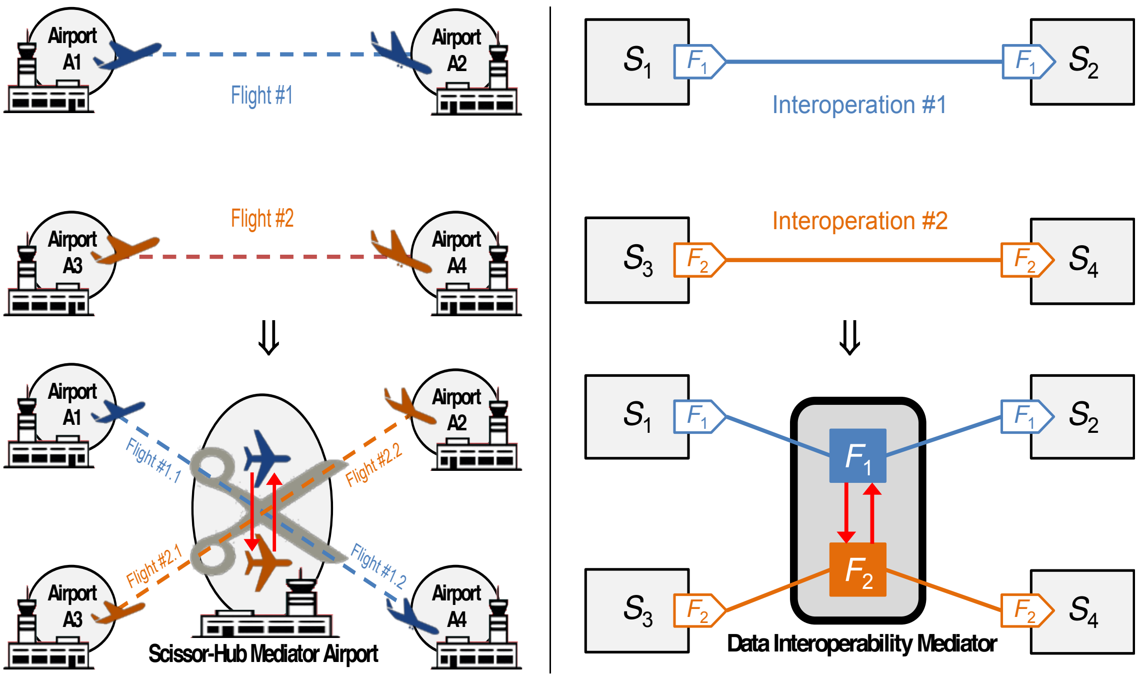


Figure 44: From (left) mediated approach (scissor-hub) in commercial air transportation to (right) mediated approach to data interoperability

This mediation mechanism would then perform as the data interoperability facilitator in the environment acting as the interoperability authority/services in the data exchange. Both, source and consumer data systems, would somehow plug into the mediator agent, exchange few information with it – sufficient for the mediation function to execute – and then expect that the mediator makes the needed hocus-pocus (magic) that enables data providers (or consumers) to readily interoperate with designated data targets (or sources). In this way, and like commercial air transportation systems, it could be that a mediated approach could settle/reconcile the differences in between data systems in data interoperability environments.

In systems integration, the Mediator is a design pattern that describes a central connection point that controls the interaction between a set of objects (Eriksson, Feizabadi, & Zamani, 2001). The Gang of Four (Gamma, Helm, Johnson, & Vlissides, 1994) provided the original definition of the Mediator design pattern as "allowing loose coupling by encapsulating the way disparate sets of objects interact and communicate with each other and allowing for the actions of each object set to vary independently of one another". The Mediator establishes a loosely coupled system – where each of its components has, or makes use of, little or no knowledge of the definitions of other separate components (Wikipedia, 2013) – by keeping objects from explicitly referring to each other and thus letting their interaction vary independently. The mediator pattern is defined as a behavioural (design) pattern, as it is used to manage algorithms, relationships and responsibilities between objects in a system.

Generally speaking, the mediator pattern is used to handle complex interactions between related objects, helping with the decoupling of those objects. Mediators are a good choice of pattern when the interaction between objects is "complicated" (e.g. too many relationships between objects in the system) but well defined. A mediator is itself an object (or component) that centralises behaviour management, thus acting as a central hub of a given operation or function. System objects interface the mediator that is then responsible for facilitating/smoothing their interactions. And by letting a central hub (mediator) manage the interaction between objects, one separates the interaction behaviour from the individual objects' behaviour, which already gives a better view of the system as a whole and, from there, makes it possible to put in practice methods and algorithms that boost the global performance of the interactions.

By using a mediator, the interaction pathways decrease in number since there is a move from one-to-one links between every object-pair that needs to interact, to one-to-many interactions between the many objects and the mediator. For example, in a system with many objects, and assuming the extreme case where every object is required to interact with any other object, there will be $\mathrm{N} \times (\mathrm{N}-1)$ dependencies if linking on a point-to-point scheme, whereas there are $2 \times \mathrm{N}$ dependencies if a mediator is used (see Figure 45). In the example with five objects, this means $5 \times (5-1) = 20$ connections in a point-to-point scheme, compared to the $2 \times 5 = 10$ connections by using a mediator. As a direct consequence, the coupling degree of objects shifts from tightly coupled to more loosely coupled when using a mediator, which then eases the inclusion of new objects and/or the replacement or reutilisation of existing objects.

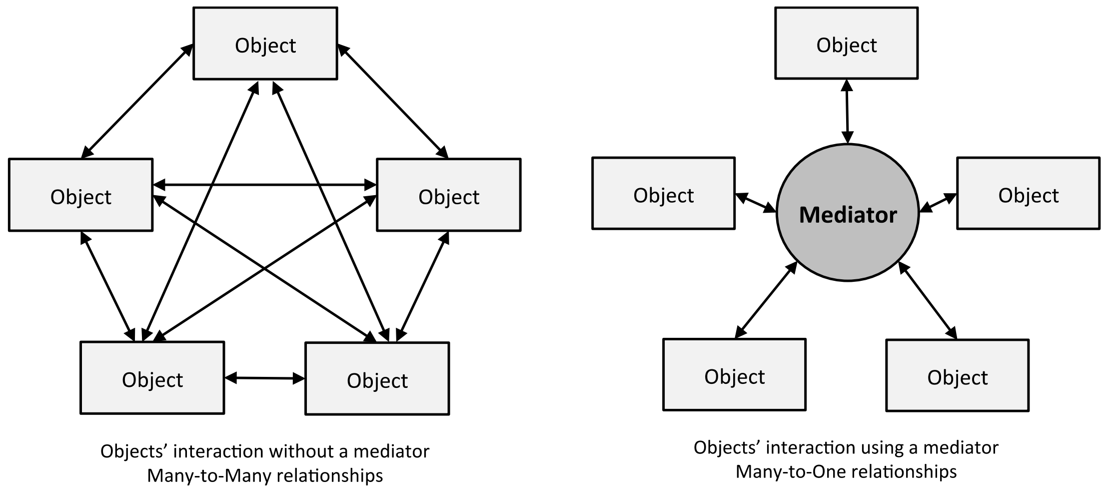


Figure 45: Representation of the Mediator pattern for objects (in systems integration)

And so, maybe by using a mediation approach to data interoperability, it could be possible somehow for data feeds to reach out to every designated data consumer without needing to have many interoperations data-connecting all of them. In practical terms, this would mean that, by using a data interoperability mediator, one might not require to create that many interoperability artefacts – data importers and data exporters – in the respective data systems to accomplish interoperability. And in this way it would be conceivable to achieve the best interoperability in a heterogeneous data environment by using much fewer resources (number of interoperability artefacts), which would in turn imply considerably less effort/cost to connect more of the data systems in the environment.

4.3 *From Hub-and-Spoke to Interoperability Compositions*

4.3.1 Hub-and-Spoke Paradigm

In 1938, passengers who flew with Delta Airlines (IATA: DEL) from Fort Worth, Texas to Atlanta, Georgia had to stop at six intermediate points along the route (Lewis & Newton, 1979) (Figure 46), and a lot more if a person wanted to travel, for instance, from Fort Worth to Miami – thirteen jumps! This was clearly annoying and tedious for passengers, who had to spend a considerable amount of time to accomplish their trips. It so happened that we were in the early ages of commercial air transportation and airplanes had a different use other than moving passengers: mail transport service. At that time, airplanes were called mail planes even though they carried passengers.

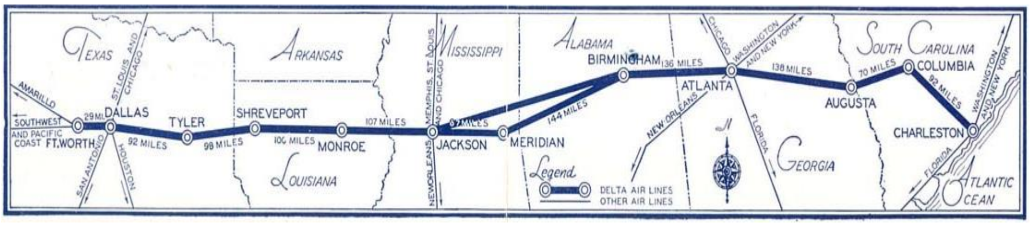


Figure 46: Delta Airlines Trans-Southern Route, March 1938

Back then, the flying routes of these so-called mail planes (Contract Air Mail – CAM – routes) were regulated by the US state and had been organised in the same manner as the rail and horse postal services routes – from city to nearby city in a point-to-point type of connection. When air transport evolved and focused much more on passenger service – even though it also carried mail, and other goods – a new model, one that saved time and used much fewer flight connections, was deemed required.

Airliners have therefore attempted to come up with alternate schemes that provide passengers with greater liberty and fewer hurdles and, perhaps more importantly, that (Wensveen, 2011) provide improved operational models e.g. by maximising the number of passenger seats filled. The hub-and-spoke system establishes a number of routes connected to a central hub airport where passengers are collected from feeder flights, transferred to other flights on the same line, and then carried to the ultimate destination (NTIS, 1988). Passengers land at a hub airport and straightaway transfer to another flight to continue voyage; at major hubs, the percentage of passengers who exchange flights is much higher than that of who gets off. The traffic pattern at a hub airport consists of closely spaced 'banks' of arrivals and departures (Wensveen, 2011).

The hub-and-spoke – also sometimes referred to as spoke-hub – paradigm (or model, or network) is a system of connections arranged like a chariot wheel, in which all traffic moves along spokes connected to the hub at the centre (Mateus, 2012). In the abstract sense, a location is selected to be a hub, and the paths that lead from points of origin and destination are considered spokes (Iseki, Taylor, & Miller, 2006). The terms "hub" and "spoke" create a pretty vivid image of how the hub-and-spoke system works: a hub is a central airport through which flights are routed, and spokes are the routes that planes take flying out of the hub airport (Bonsor, 2001). See Figure 47 showing this parallel of a spoked wheel and the basic hub-and-spoke concept.

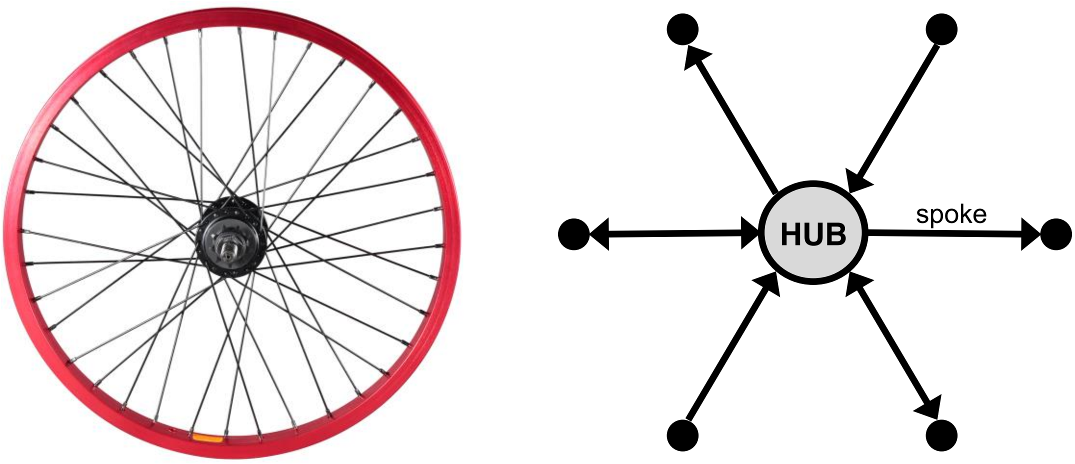


Figure 47: A spoked wheel (left) and the basic hub-and-spoke paradigm (right)

Credit for the hub-and-spoke paradigm is usually, and to some extent correctly, attributed to Delta Airlines in the effort to compete with Eastern Air Lines (IATA: EA). Delta Airlines itself declares that it pioneered the use of the hub-and-spoke system in 1955, when scheduled airplanes brought passengers to a hub airport where travellers connected to other Delta flights (Delta Air Lines, 2013). However, and according to (Hoogerwerf, 2010), Eastern Air transport had already contemplated the concept of a hub-and-spoke model as early as May 1930, at the time when Postmaster General Brown[23] was rearranging the United States aviation map.

---

[23] Walter Folger Brown (May 31, 1869 – January 26, 1961) was Postmaster General of the United States of America from 1929 through 1933 under president Herbert Hoover administration. While in office, Postmaster General Brown secured a reduction of airmail rates and a consolidation of airmail routes-policies that aided the development of commercial aviation. He is sometimes known as the Postmaster General who built the U.S. Airline Industry!

Eastern was competing with Delta, and other airliners, for the route to the west, which would tie into its north/south operation. Eastern felt it would be a splendid operation, as it would hub out of Atlanta and could be handled both to the north, south, and west, making it a good economical endeavour. Regrettably, Postmaster General Brown gave the route to AVCO – the Aviation Corporation, a holding company that later became American Airlines – and the hub-and-spoke would only become a reality much later, in 1955, when Delta obtained approval to operate Atlanta to New York, and later extended scheduled service to Philadelphia, Baltimore and Charlotte.

For Delta, it all started in 1941, when the company moved its headquarters from Monroe to Atlanta, to centre itself along its new route network that stretched to Chicago, Miami, and New Orleans. Atlanta was selected due to its location, reasonable climate, entrepreneurial and economic dynamism, and visionary leadership (Hoogerwerf, 2010). Actually, long before Atlanta became a crossroads of the air, indeed long before it became even a city, it was destined to be a hub. (Braden & Hagan, 1989) state, in their excellent book "A Dream Takes Flight", that "for hundreds of years native Americans, traveling the river valleys and following the Peachtree and Etowah trails, converged on the area, and this accessibility led to Atlanta's founding as a rail head". From moving people by train to using airplanes was just a matter of time.

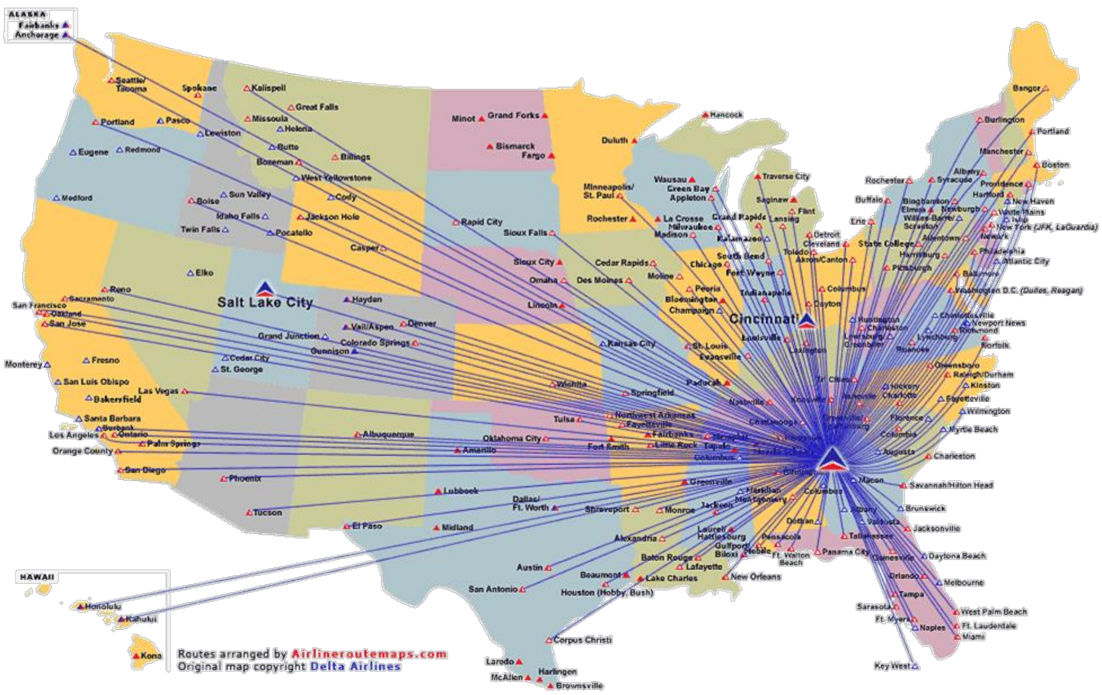


Figure 48: Delta Air Lines USA route network from Atlanta Hub (AirlineRouteMaps.com, 2011)

By far, the Hartsfield-Jackson Atlanta International Airport (IATA: ATL) has been the world's busiest airport (and hub) as regards passenger traffic since 1998 and number of landings and take-offs since 2005. According to Airports Council International 2012 statistics, the Atlanta International Airport ranked #1 as the world's busiest airport with traffic of 95,513,828 passengers and 930,310 aircraft movements! But many other hubs exist in the world of commercial civil aviation, in the six corners of the world: the Beijing Capital International Airport (IATA: PEK) ranked #2 in the world's busiest airports list; the London Heathrow Airport (IATA: LHR) ranked #3; the Dubai International Airport (IATA: DBX) ranked #10; the Sydney Airport (IATA: SYD) ranked #31; the São Paulo-Guarulhos International Airport (IATA: GRU) ranked #43; etc.

Airliners have implemented the hub-and-spoke paradigm because it enables a transport service between more origin and destination points with fewer resources (airplanes). As an example, a carrier needs a minimum of two flights to serve two city-pairs in a point-to-point route system; if operated via a hub, those same two flights can serve as many as four city-pairs. In the very same manner, to serve nine cities using a fully connected point-to-point approach would require a total of 72 flights (see Figure 49 left, and consider an inbound and outbound flight per edge in the graph). If a hub-and-spoke approach is used this can be greatly reduced; depending on the selected structure and organisation we can connect all those cities with much fewer connections. In the example of Figure 49 right, nine cities are connected altogether with a total of 18 flights. This would mean, in this case, that the hub-and-spoke organisation would require only 25% of a point-to-point connectivity!

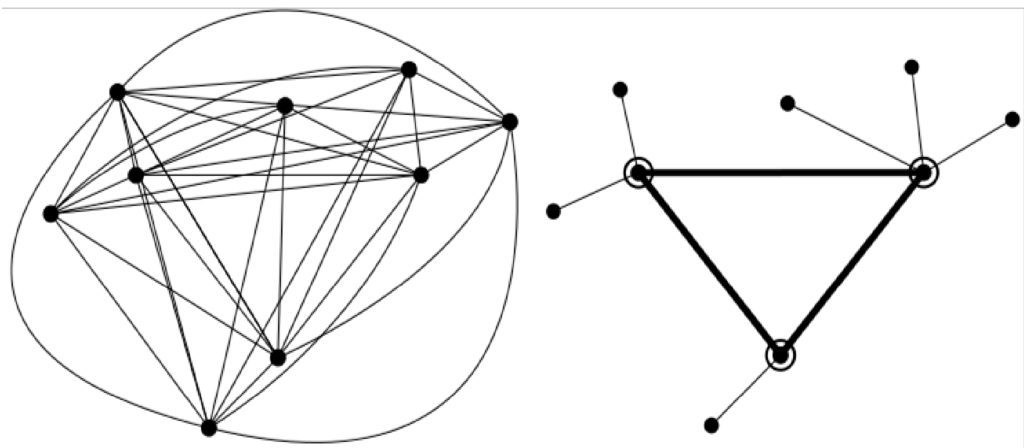


Figure 49: A fully connected point-to-point nine-node network (left) and a hub-and-spoke network organisation of the same nine nodes (right)

In practice this means that, at a cost of hopping in a hub, carriers could serve many more destinations with the same resources – airplanes, which are the most expensive asset of the business. And since there are fewer routes, and considering the same number of airplanes, airlines can schedule more frequent flights along each route and make full use of each aircraft capacity. Moreover, and in this way, travellers could be engaged with the same carrier for longer distances by hub hopping, thus raising the average revenue per passenger. Furthermore, and by concentrating the flow of passengers toward one same central point (hub), it is possible to service those city-pairs that do not have enough demand to justify a direct flight operation. Last but not least, airlines might concentrate services (administration, maintenance, etc.) in one same location leading to economies of scale and operational efficiencies.

Furthermore, adding a new airport to the airport network and making sure that it is properly connected into the air transportation system is a no-brainer. Doing this simply implies the establishment of routes (spokes) to one (or more) relevant airport hub(s) as this duly connects the new airport inside the system. Also, and in this way, it is possible to service airports that cannot admit certain classes of aircrafts – e.g. local airports that typically cannot service "big" airliners like the Airbus A380. Moreover, it would never make sense to alter these airports to support such aircrafts, both due to unaffordable costs and low passenger load. Then these non-modifiable airports can then always be used for short- to medium-haul aircrafts originating from low load that scissor-hub at an intermediate airport. This means that the (worldwide) commercial air transportation system can grow, on demand and in a decentralised way, given that global interconnection is assured by (airport) hubs and spokes (routes between airports).

Meanwhile, airlines have extended the concept of the hub-and-spoke in various ways, seizing additional advantages and realising novel business models. One way is for carriers to pair up with other airlines on code-sharing agreements or – more extensively – in airline alliances. Codeshare is an aviation business arrangement where two or more airlines share the same flight, whereas airline alliances are more profound and long-lasting collaboration agreements between carriers – including the sharing of airplanes, operational infrastructures and joint strategy definitions. Codeshare and/or alliances enable a joint exploitation of routes between airlines, mainly built around hub airports, which makes the use of a hub-and-spoke operational model even more efficient and scalable.

One other outstanding business model built-on the hub-and-spoke paradigm is that of hub-to-hub operations. The idea is to create major routes between hubs, especially servicing high load between hubs routes. Probably the best case of this practice today is Emirates airline (IATA: EK), which focuses mainly on passenger air transport from and to major hubs, to all parts of the world via Dubai International Airport (IATA: DBX). Look at Figure 50 that depicts the route map (and destinations) of Emirates.

All of Emirates' fleet consists of wide-body airliners – Airbus A330/A340, Airbus A380 and Boeing 777 – for operating only medium-haul to long-haul flights. Operation in a single home hub and a wide set of aircrafts of only a few models, along with a leaner workforce, enables Emirates to have lower operating costs and higher efficiency, comparable to a low-cost carrier. Emirates has been accused of unfair competition by established carriers, but, it is the author's view, that it approached the market with a superior business model – the low-cost of long distance travel.

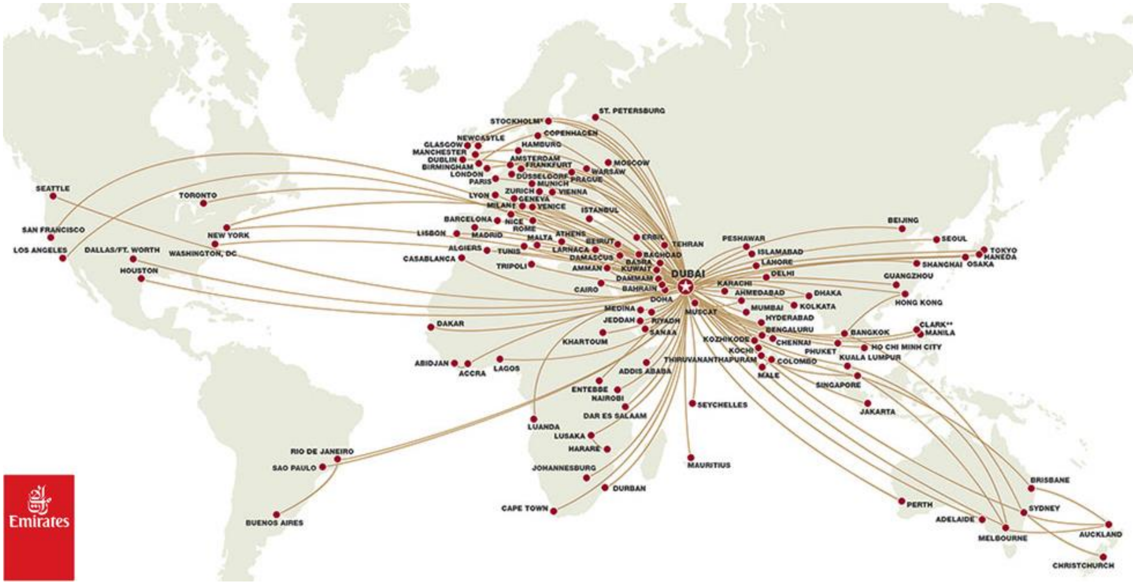


Figure 50: Emirates Airline Route Map

The hub-and-spoke paradigm made a global commercial air transportation network possible, facilitated by major hubs distributed around the globe. This network structure of airport hubs makes it possible for a passenger to traverse the whole world, from a given source to any destination of choice, in a few flight hops. And also, as a result of the extensive networks made possible by hub-hops, carriers are able to attract passengers and, with tight scheduling, to meet passengers' preference for single-carrier service (Wensveen, 2011).

4.3.2 Towards Interoperability Compositions

The hub-and-spoke paradigm allows airliners to service a wide number of city-pairs, without additional resources – particularly airplanes, the key and most costly asset in air transportation – than point-to-point do connections by hopping in multiple (hub) airports. Similarly, and looking from the perspective of a data interoperability support system, probably it might also be possible to efficiently interoperate many (source to target) data systems by "hopping" formats – from source-system format to target-system format via other formats. In this way, and as with the hub-and-spoke approach of commercial air transportation systems, it could be possible to realise (full) data connectivity (i.e. system coverage) in the environment, without the need for lots of interoperations between constituent (source to target) data systems.

The baseline idea of Interoperability Compositions would be then to take advantage of the possibility to traverse in between multiple (interoperability) "hubs" in order to reach out from one system to any other system in the environment. That is, to provide data from one given system, then possibly somehow hop via multiple other formats in order to be able to share data (i.e. be interoperable) with some other system that take data in a much different data format than that of the originating system.

Figure 51 presents this parallel of the hub-and-spoke paradigm and of Interoperability Compositions. The left part of figure presents the move from a point-to-point interconnectivity model to an optimised hub-and-spoke network that minimises the number of routes by promoting airports to hubs; on the right side of figure one can see the parallel to move from point-to-point interoperability to a scheme that would take advantage of interoperability compositions to make interoperable systems in the scenario. Take notice that some systems would need to traverse over one format (aside from the source and target data formats) to be interoperable!

The multiple possibilities (paths) of going from one system to another system, via data format hopping, would then create a comprehensive network that represents all the interoperability options in the environment – the interoperability network. This would then give the opportunity to apply network theory methods and particularly to establish ways as to reduce the global number of connection that might be needed to interoperate all designated systems in an interoperability scenario, i.e. to minimise the number of resources that might be needed to attain interoperability in the environment.

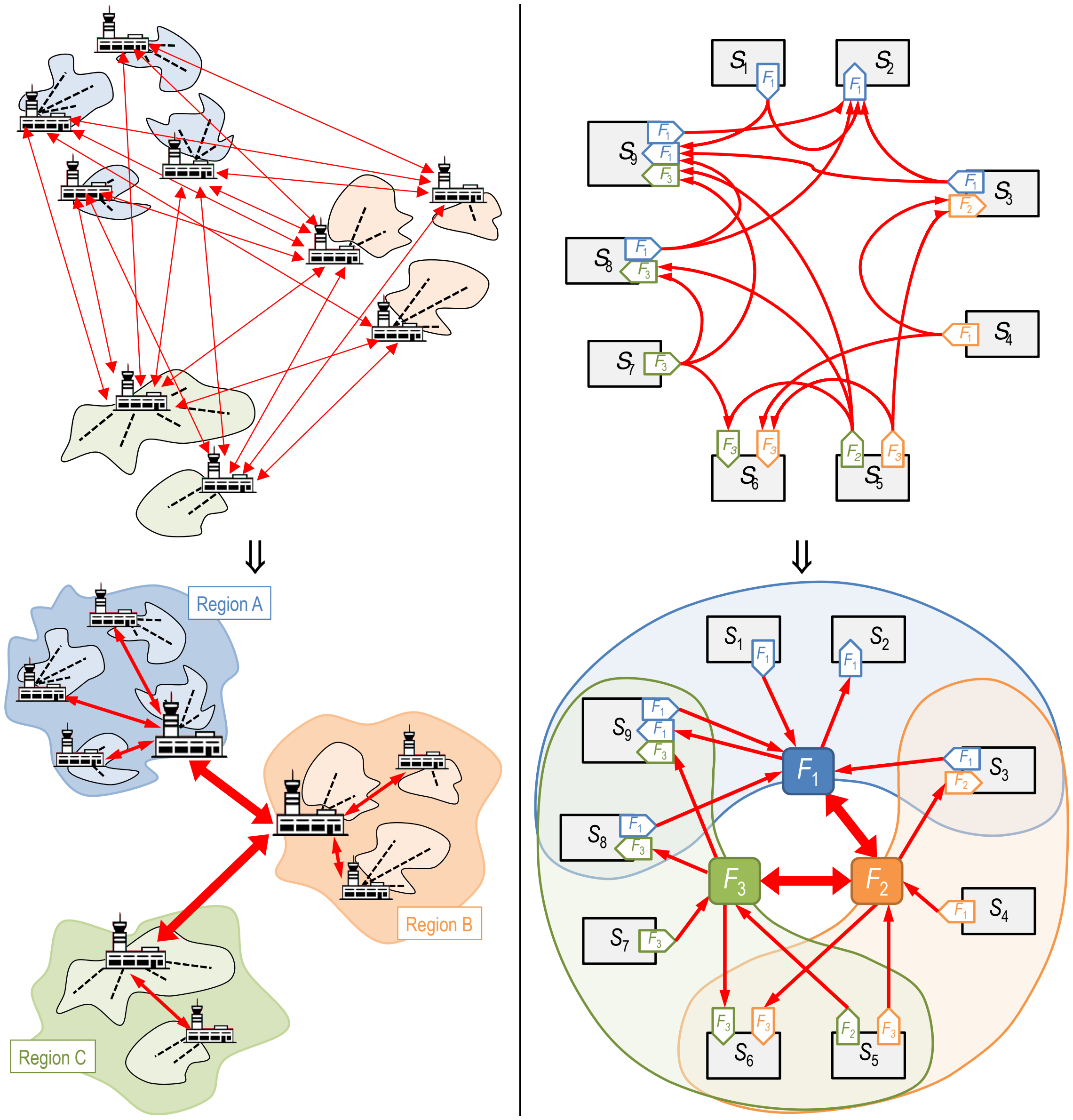


Figure 51: From (left) Hub-and-Spoke paradigm in commercial air transportation to (right) Interoperability Compositions

Even, it might possible to somehow, just like in the hub-and-spoke commercial air transportation networks, to establish interoperability networks where most data formats nodes are not neighbours of one another, but where most data formats can be reached from every other by a small number of (data formats) hops. This would enable to have an optimal interoperability network that much saves on interoperability resources especially in large-scale conditions, qualifying by what is called a small-world (interoperability) network. Furthermore, such interoperability networks might also possess characteristics of scale-free networks for robustness and scalability.

5

Solution: Hub-and-Spoke Interoperability

"No problem can stand the assault of sustained thinking."

Voltaire, writer and philosopher

A novel approach to address the issues highlighted in the Point-to-Point Interoperability approach is presented in this chapter. This approach, called Hub-and-Spoke (H&S) Interoperability, resorts to a series of concepts inspired by the Commercial Air Transportation domain, to handle Interoperability in the domain of Information Systems. By doing so, this approach aims to solve the Coverage-Modifiability problem and to minimize the NIA-Connections Problem. Regarding the first problem, it is solved by resorting to a mediated solution, where no system is subject to modifications. In respect to the second problem, a solution is proposed that reuses interoperability artefacts in order to render interoperable systems that are non-interoperable.

In this chapter, the concepts associated with H&S Interoperability are presented, putting a special focus on the concepts of Interoperability Mediator and Interoperability Compositions. After this presentation, a formalisation, both mathematical and graphical, of all concepts and properties of this approach is performed. An algorithm to improve interoperability of a scenario to its maximum, using the H&S Interoperability approach, is also presented and used in this chapter in scenario simulations. These simulations are used to verify the characteristics of this approach, supporting the definition of a research hypothesis to solve the research problem.

5.1 Concept

Interoperability qualifies the ability of two or more systems (or components) to exchange and use the information that has been exchanged (IEEE, 1991). Interoperability defines the aptitude of entities (people, objects, systems) to interlink ("talk" and "understand") each other; non-interoperability defines the inability of entities to interconnect. Data interoperability relates to the interoperability of (data) systems, at the data level, so that information can be duly exchanged. And two data systems are said to be interoperable if they have a compatible importer and exporter, i.e. if source and target data systems, export and import data in one same data format.

But interoperating multiple systems altogether presents key challenges. First, there is the issue of the non-modifiable systems that hinders interoperability in a data exchange environment. Data systems are said to be non-modifiable due to being of proprietary/closed/legacy/etc. nature or simply because the costs for such changes make modifications prohibitive. The more the number of non-modifiable systems exist in an interoperability scenario the least interoperability coverage exists; interoperability coverage defines the portion of the total number of systems in an interoperability scenario that are interoperable.

Then, to establish interoperability between data systems in a data exchange interoperability scenario it is needed to add new data exporters and/or data importers – called interoperability artefacts – with such formats that makes compatible the data exchange. And, interoperating altogether many systems might require a vast number of changes to systems. As it happens, maximising interoperability in an interoperability scenario might necessitate such a considerable number of changes to systems that makes it prohibitive or simply impractical to be accomplished.

The Hub-and-Spoke Interoperability is an approach to data interoperability that has been conceived using principles as to address these key challenges. The Hub-and-Spoke Interoperability approach deals with the requirement to have interoperability between all data systems that should be interoperable within a data exchange environment. And, the Hub-and-Spoke Interoperability approach addresses the problem of the potential high number of connections (or interoperations) between systems that relates with the need to have many importers and/or exporters in systems to attain interoperability in the environment.

The hub-and-spoke – also sometimes referred as spoke-hub – paradigm (or model, or network) is a system of connections arranged like a chariot wheel, in which all traffic moves along spokes connected to the hub at the centre (Mateus, 2012). In the abstract sense, a location (object) is selected to be a hub, and the paths that lead from points of origin and destination are considered spokes (Iseki, Taylor, & Miller, 2006). The terms "hub" and "spoke" creates a good image of how the Hub-and-Spoke Interoperability approach works: an "interoperability hub" is a format to where interoperability is routed through, and spokes are the links from systems to hub formats. Interoperability is then accomplished by interconnecting "interoperability hubs" (hub-to-hub) via mappings between data formats – termed interoperability functions.

To explain the Hub-and-Spoke Interoperability concept let's consider an interoperability scenario comprised of nine data systems (‘*System 1’* ... ‘*System 9’*) and three data formats (‘*A*,’ ‘*B’* and ‘*C’*), interoperated altogether as shown in Figure 52. The scenario represents a data exchange environment comprised of systems with different natures: data provider only (*e.g. ‘System 1’*), data consumer only (*e.g. ‘System 6’*), and mixed data provider and consumer (*e.g. ‘System 8’*). In the scenario, a set of system-to-systems is already capable to exchanging data (i.e. are interoperable) graphically symbolised by directed arrows, representing interoperations. Also note that there exist two non-modifiable systems (‘*System 2’ and ‘System 4’*) in the scenario and that are duly represented by a filled black box.

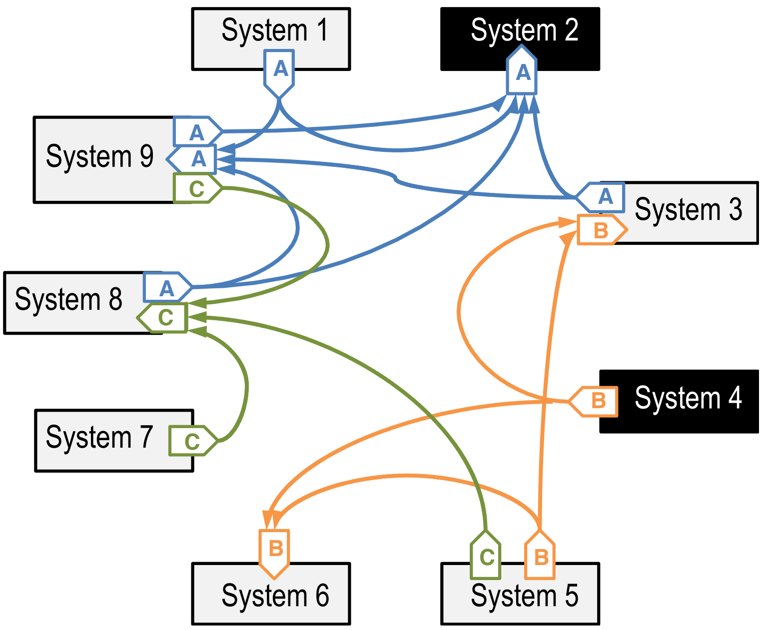


Figure 52: Nine-systems three-formats interoperability scenario example

In this scenario one can observe some key interoperability issues: First, Interoperability is not fullest as there are admissible system-to-systems interoperations that are not established (e.g. '*System 7'* to '*System 6'*, '*System 1'* to '*System 3'*, and many others); and, maximising interoperability in the scenario would require the inclusion of many interoperability artefacts in systems. Note however that interoperability for instance of '*System 2'* to other systems in the scenario is not a requirement as '*System 2'* is considered to always be only a data consumer and is to never provide any data. The same thing happens with '*System 7'* that will always be a data producer and is not ever supposed to become able to consume data.

Moreover, one can observe that data exchange out of *'System 4'* to *'System 2'* will never be possible. This is due to the fact that none of the systems can be modified to include new interoperability artefacts (importers and/or exporters) and therefore data sharing will not be possible. And so, considering this situation, the maximum possible interoperability in between systems within the interoperability scenario – the maximum possible interoperability coverage – will not be attained.

Figure 53 presents then the nine-systems three-formats scenario example interoperated by following the Hub-and-Spoke Interoperability approach.

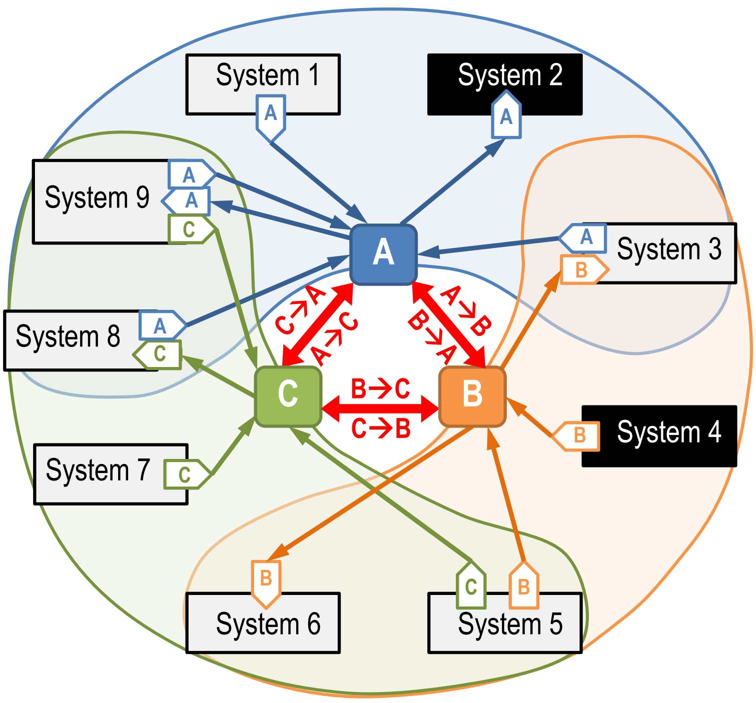


Figure 53: Nine-systems three-formats scenario example interoperated following the Hub-and-Spoke Interoperability approach

In the example of Hub-and-Spoke Interoperability one can see the outstanding features of the approach that relate to the key issues of interoperability: In the example, any system can exchange data with all other systems in the environment – i.e. all systems are interoperable with one another. For instance, interoperability from *'System 1'* to a *'System 3'* – that did not exist in the initial scenario representation – is now possible being facilitated by the interoperability function of data format *'A*' to data format *'B'* (*A→B*). And notably, the scenario has been made fully interoperable by the introduction of few interoperability functions – six in total. Note again that there exist systems that are not required to exchange data; for instance, interoperability from *'System 6'* to any other system in the scenario is inexistent simply because of the immutable nature of *'System 6'* that is a data consumer only.

More, one can also observe that data exchange out of non-modifiable *'System 4'* to non-modifiable *'System 2'* is now possible; interoperability is enabled by the interoperability function of data format *'B'* to data format *'A'* (*B→A*). This means that interoperability between any non-modifiable systems that have disparate data formats is now possible. In this way, in the Hub-and-Spoke Interoperability approach, the maximum possible interoperability in the scenario – the maximum possible interoperability coverage – will attained as all systems will be able to exchange data between each other, irrespective of being modifiable or non-modifiable systems.

And very importantly, the Hub-and-Spoke Interoperability approach accomplishes all this – full interoperability coverage with the use of acceptable interoperability resources – without needing to change any of the systems involved in the data exchange. This is especially relevant for non-modifiable systems, as those simply cannot be altered. And also the approach enables systems to exploit already in-place interoperability resources (i.e. interoperability functions) to reach out to more systems in the scenario.

Furthermore, the Hub-and-Spoke Interoperability approach is built around two key cornerstones: Interoperability Mediation and Interoperability Compositions. Mediated Interoperability relates to the centralisation of interoperability management as to globally organise, optimise and execute interoperability whereas Interoperability Compositions relate to a point-wise application of interoperability functions (one after the other) to "hop" formats in order to go from a source-system format to target-system format via some other(s) format(s).

5.1.1 Interoperability Mediation

The Hub-and-Spoke Interoperability takes advantage of a mediation scheme. A mediator establishes an unique point for managing and controlling the interactions between elements in a system. A mediator is typically used to manage algorithms, relationships and responsibilities between objects/components in a system.

Considering the Data Interoperability system, the (interoperability) mediator is then an object (agent) that centralises interoperability management (and its execution), acting as a central hub for interoperability provision. Data systems interface the mediator, who is then responsible for facilitating interoperability in between them. Systems plug into the mediator agent – i.e. plug into the interoperability support system –, exchange few information with it – sufficient for the mediation function to be performed – and then it is the role of the mediator to enable interoperability between designated systems. In this way, the mediated approach settles/reconciles the differences – of data formats – between data systems taking part of the interoperability scenario.

The Interoperability Mediator is then the unique point for global organisation and execution of interoperability in the data exchange environment. The Interoperability Mediator provides facilities (adaptors) to read and write data following the supported data formats. Systems in the environment then attach to the designated adaptors of the Interoperability Mediator (reader and/or writer of the supported data format(s)) and are able in this way to exchange data with the Mediator. Then, it is the role of the Interoperability Mediator to oversee and execute interoperability between systems via their supported data formats. The Mediator encloses the knowledge of all the data formats that each system supports and is able to know how to interoperate any two data systems via their supported data systems. System exchange data between each other via the interoperability facilitation mechanisms provided by the Mediator.

Figure 54 shows an Interoperability Mediator for the nine-systems three-formats interoperability scenario running example. The Mediator is represented using a circle (in the centre of the image) with the label ‘Mediator’ inside it. And, in the boundary of the Interoperability Mediator are represented three read/write adaptors for data formats *‘A’*, *‘B’* and *‘C’*. These adaptors enable the Mediator to take/give data from/to systems participating in the interoperability service, following the supported data formats. The arrows (in the exterior) represent the data flows from/to systems to/from the mediator.

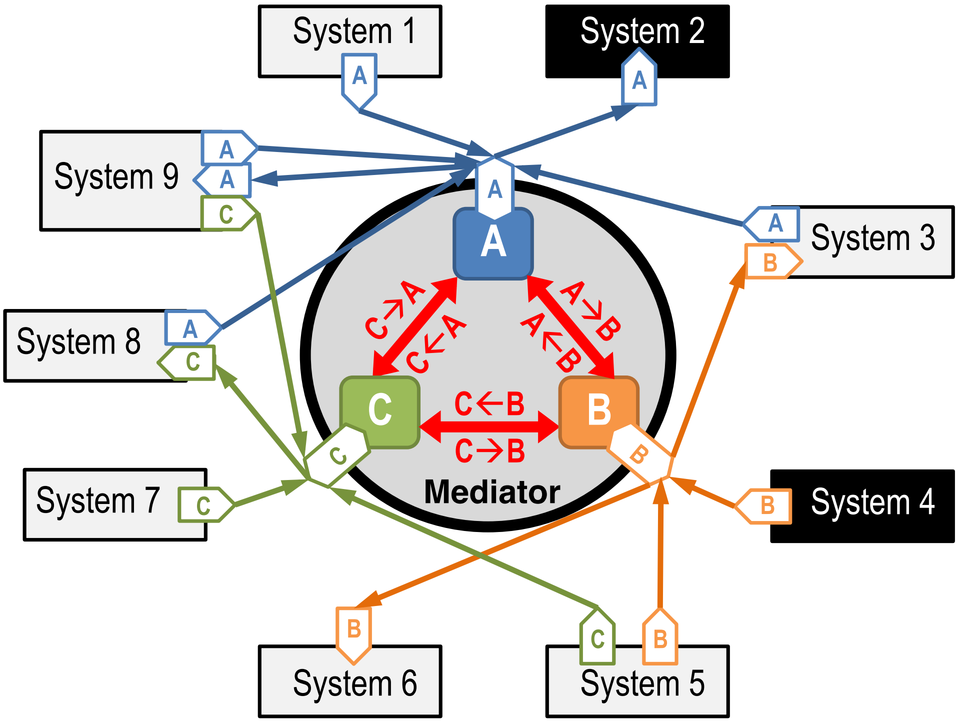


Figure 54: Illustration of Interoperability Mediation in the nine-systems three-formats interoperability scenario example

Furthermore, the Interoperability Mediator makes it possible to globally optimise interoperability in the environment. Particularly, the Mediated Interoperability enables the realisation of Interoperability Compositions so as to establish compositions of interoperability functions making it possible to interoperate a source data format to a target data format via some other intermediary data format(s).

5.1.2 Interoperability Compositions

In the Hub-and-Spoke Interoperability approach, interoperability is managed and performed at the "interoperability hub" level. That is, data interoperability is organised and executed "centrally" by the interoperability mediator, which enables a comprehensive and global management of interoperability. This also makes it possible to implement schemes that optimise interoperability in the environment, i.e. it is possible to inspect further options that enhance and improve overall interoperability.

The mediated approach separates the interoperability behaviour out of the individual objects' behaviour and brings to a unique point the interoperability management and control. This enables a clear view of the interoperability system as a whole and from here to put in practice methods and algorithms that boosts global performance of interoperability. And that is exactly what the Interoperability Compositions is about: still make fully interoperable a given scenario but using a satisfactory number of interoperability resources – in the case, interoperability functions.

The basic idea of the Interoperability Compositions is to operate interoperability functions one after the other (in multi-hop style) going from a source data format to a target data format via other formats. Interoperability Compositions follows after the homonym mathematical operation – compositions of functions, $(f \circ g)(x)$ – denoting the nesting of two or more functions to form a single new function. And, like such, in Interoperability Compositions, the result of each interoperability function is passed as input of the next, and the result of the last one is the result of the whole.

Figure 55 displays an illustration of the Interoperability Compositions using the same nine-systems three-formats interoperability scenario example as presented earlier in this section. As expected, the Interoperability Compositions also enable all data systems – data exporter to data importer – to be interoperable; but however, this technique uses a minimum number of interoperability functions to make the environment fully interoperable.

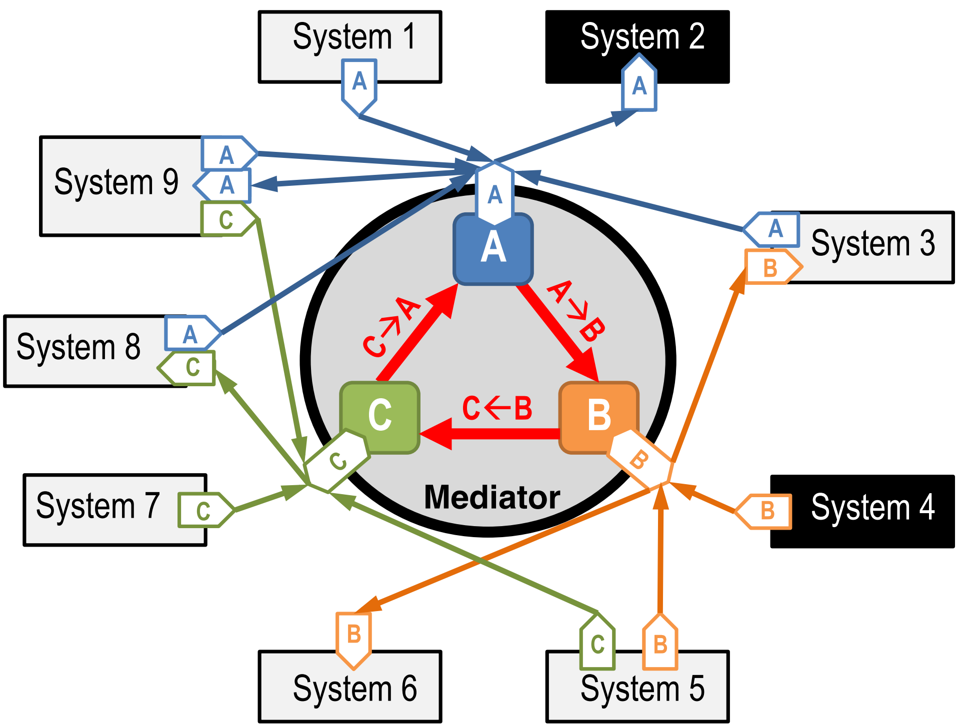


Figure 55: Illustration of Interoperability Compositions in the nine-systems three-formats interoperability scenario example

Let's focus attention on the interior of the Interoperability Mediator in Figure 55. One can observe that the Interoperability Compositions scheme inside the mediator is making it possible to interoperate all systems in the scenario. Some systems are being interoperated simply via a direct interoperability function between two given formats. This is the case for instance of interoperability from *'System 1'* to *'System 3'* that is habilitated by the interoperability function of data format *'A'* to data format *'B'* (*A→B*). Or also for example the situation of data exchange from *'System 4'* to *'System 8'* which is mediated by the interoperability function of format *'B'* to format *'C'* (*B→C*). The same thing for data sharing out of *'System 7'* to *'System 2'* that is enabled by the interoperability function of format *'C'* to format *'A'* (*C→A*).

But however, there are those systems that are only interoperable by traversing multiple formats. That is for instance the circumstance of interoperability from *'System 1'* to *'System 8'* that is only make possible by a composition of interoperability function from data format *'B'* to data format *'C'* (*B→C*) after interoperability function from format *'A'* to format *'B'* (*A→B*), thus accomplishing the composite interoperability function of format *'A'* to format *'C'* through format *'B'* (*A→B→C*). An identical situation take place when needing to interoperate, as an example, *'System 4'* to *'System 2'* that is performed by the composite interoperability function of format *'B'* to format *'A'* through format *'C'* (*B→C→A*). And the same things happens as to interoperate for instance *'System 7'* to *'System 3'* that exploits the composite interoperability function of format *'C'* to format *'A'* through format *'B'* (*C→A→B*).

The baseline strategy of the Interoperability Compositions scheme is to establish full (possible) interoperability in the scenario by using the fewest interoperability resources (interoperability functions). So, Interoperability Compositions makes it possible to efficiently interoperate many – source to target – data systems by "hopping" formats – from source-system format to target-system format via other formats. And, as scale grows – number of systems with disparate data formats – the efficiency of Interoperability Compositions is even more visible as its interoperability networks tend to interconnect all systems (via their supported formats) with minimum interoperability resources employed, i.e. with an optimum number of interoperability function used. As such, Interoperability Compositions provide then a great advantage in terms of interoperability optimisation and its practical realisation.

5.2 *Definition and Properties*

The Hub-and-Spoke Interoperability is characterised by using an entity (mediator) to interoperate systems with each other. This entity uses Interoperability Functions, which provide mappings between heterogeneous Data Formats. There is the need to study the Hub-and-Spoke approach, as P2P Interoperability was studied, and in order to do so a definition of the properties and underlying concepts of this approach are needed.

To provide a rigorous description and definition, mathematical formalisations are presented. These formalisations help translating concepts into their computer representations. A graphical representation is also provided, thus offering a visualisation of the situations and examples.

5.2.1 System

A System is defined by (5.2.1); n is the number that identifies a system in an interoperability scenario, i.e. the system identifier. Each system has a unique identifier within a given interoperability scenario. In the Hub-and-Spoke Interoperability approach, a system is characterised by all the import and export capabilities it has. Therefore, the union of the Importer and Exporter sets represents a System from an interoperability perspective, as shown in (5.2.1).

$$\begin{aligned} \mathrm{S_n} &= \mathrm{System} \quad , \qquad \mathrm{n} \in \mathrm{Z}^+ \\ &= \mathrm{E_{S_n}} \cup \mathrm{I_{S_n}} \, , \qquad \mathrm{S_n} \neq \emptyset, \mathrm{n} \in \mathrm{Z}^+ \end{aligned} \qquad (5.2.1)$$

In theory, a system can support an arbitrary amount of importers and exporters. A system can have either the Importer set or the Exporter set empty; however both sets cannot be empty at the same time, since that situation would correspond to a system unable to communicate at all. The fact that a system can have one of the two sets empty – either the importer or exporter set –, comes from the design intent of such a system which is linked to its nature.

Some systems are designed to only have the capability to export information, whereas other systems are designed to only import it. Systems that only come with exporters have the nature of data sources, only; systems that come only with importers have a nature of data consumers, only. Otherwise, systems can have both exporters and importers, and so possess a mixed nature of data source and data consumer.

An Exporter is like an (interoperability) artefact that represents the capability of a system to make information available in a given data format. An Importer is an artefact that performs the reverse/opposite operation: it represents the ability of a system to interpret information in a given data format.

The ability of a system to provide data in a given data format is qualified by the existence of an exporter, while an importer enables a system to take and interpret information expressed in a given data format. The absence of a given data exporter/importer means that the system cannot deliver/interpret data following such data format.

A Data Format specifies how the data (to be exchanged) is organised; basically, a data format defines how the (shared) data is represented and structured. Two arbitrary systems can only exchange information if, and only if, both support the same data format. Both a Data Format and a set of Data Formats are formalised as presented in (5.2.2) and (5.2.3), where 'm' is the format identifier.

$$\mathrm{F_m = Data\ Format, m \in Z^+} \quad (5.2.2)$$

$$\begin{aligned}\mathrm{F} &= \mathrm{Set\ of\ Data\ Formats} \\ &= \{\mathrm{F_1, F_2, \ldots}\}\end{aligned} \quad (5.2.3)$$

Both exporters and importers are an association of a system (where they are implemented to perform the data export or import function) and a data format (the exact format in which data is delivered or interpreted). Therefore, both exporters and importers are identified by a system-format pair, providing the notion of a given system with the capability to deliver or interpret information using a particular data format.

The nomenclature used to represent an exporter and an importer is similar, with both being characterized by the system ($\mathrm{S_n}$) that implements the exporter/importer and the data format ($\mathrm{F_m}$), which it uses to represent/interpret the information. The exporter is expressed by of (5.2.4), while the importer is expressed by (5.2.5).

$$\mathrm{e}_{(\mathrm{S_n},\mathrm{F_m})} = \text{Exporter} \qquad \textit{(5.2.4)}$$

$$\mathrm{i}_{(\mathrm{S_n},\mathrm{F_m})} = \text{Importer} \qquad \textit{(5.2.5)}$$

A system can have more than one exporter and more than one importer, being capable of representing or interpreting their information in different data formats. All exporters and all importers corresponding to the same system can be grouped together in a set, representing all the data formats that a given system ($\mathrm{S_n}$) can accept, as represented in (5.2.6) for exporters and in (5.2.7) for importers.

$$\begin{aligned}\mathrm{E}_{\mathrm{S_n}} &= \text{Exporters of system } \mathrm{S_n} \\ &= \{\mathrm{e}_{(\mathrm{S_n},\mathrm{F_1})}, \mathrm{e}_{(\mathrm{S_n},\mathrm{F_2})}, \dots\}\end{aligned} \qquad \textit{(5.2.6)}$$

$$\begin{aligned}\mathrm{I}_{\mathrm{S_n}} &= \text{Importers of system } \mathrm{S_n} \\ &= \{\mathrm{i}_{(\mathrm{S_n},\mathrm{F_1})}, \mathrm{i}_{(\mathrm{S_n},\mathrm{F_2})}, \dots\}\end{aligned} \qquad \textit{(5.2.7)}$$

Graphically, a system is represented by a light rectangle with borders, as illustrated in Figure 56. In the graphical notation, a system is identified with a label using the format "S_n" – capital italic 'S' for system and subscript ordinal 'n' –, representing the system identifier. For example, label "S_3" corresponds to a system whose identifier is '3'.

A system can have an exporter of a given data format and also an importer of the same data format, meaning that such a system can both provide and consume data using one same format. In that case, and with the objective of simplifying the graphical representation of systems – and consequently of whole interoperability scenarios –, there are two alternate representations: the first, shown in Figure 56 (left), includes both exporter and importer symbols. Graphically, exporters and importers are both represented as an irregular pentagon – shaped like a kindergarten-drawn house or some kind of arrow – placed over a system's boundary, pointing towards the outside (if it is an exporter) or inside (if it is an importer) of the rectangle, symbolising the flow of data to/from the system.

The second, presented in Figure 56 (right), is one where the exporter and importer symbols have been merged together into a new symbol. The exact data format, which the exporter/importer uses to represent the exported/imported data, is indicated in a text label inside the exporter/importer shape.

Figure 56: Alternative graphical representations of a system (S_1) with both an importer and exporter from/to data format F_1

Another important aspect of each System is its Modifiability, i.e. the possibility of making changes. A given system is said to be Modifiable if it includes mechanisms that enable the development and deployment of new exporters and/or importers into that system. When a system does not meet this requirement, it is defined as Non-Modifiable. This way, the data exchange capabilities of a non-modifiable system will never change.

Modifiability is therefore a property (of systems) that qualifies – true or false – the ability of a system to be altered in order to accommodate new importers and/or exporters. This is represented by (5.2.8), a two-element set: a checkmark that represents the ability to be modified; and a cross, representing non-modifiability.

$$\begin{aligned} \mathrm{M} &= \mathrm{Modifiability} \\ &= \{\surd, \times\} \end{aligned} \tag{5.2.8}$$

Then, in a mathematical notation, a non-modifiable system is represented here as a system marked with the cross symbol in superscript. This representation establishes that a given system has the modifiability property set to false and therefore cannot be altered at all. Inversely, the representation of a system with the checkmark symbol in superscript means that the system can indeed be modified. The absence of any symbol in superscript, in the definition of a system, is semantically equivalent to having a checkmark, i.e. such a system is modifiable. Both Non-Modifiable and Modifiable definitions are presented in (5.2.9) and in (5.2.10). It is also important to refer that all systems must be identified as Modifiable or Non-Modifiable and that property cannot be changed afterwards.

$$\mathrm{S}_\mathrm{n}^{\times} = \text{Non-Modifiable System}, \qquad \mathrm{n} \in \mathrm{Z}^{+} \tag{5.2.9}$$

$$\mathrm{S}_\mathrm{n}^{\surd} = \mathrm{S}_\mathrm{n} = \text{Modifiable System}, \qquad \mathrm{n} \in \mathrm{Z}^{+} \tag{5.2.10}$$

Graphically, a Modifiable system is drawn in the very same way as a (plain) system, i.e. a light rectangle with borders. A non-modifiable system is represented as a rectangle (obviously, since it is still a system) with a dark filling, symbolising a “closed box” to denote that it cannot be changed. Both Modifiable and Non-Modifiable systems are illustrated in Figure 57.

Figure 57: Graphical Representation of a Modifiable (S_1) and a Non-Modifiable System (S_2)

5.2.2 Interoperability Scenario

An <u>Interoperability Scenario</u> consists of a collection of disparate systems, i.e. a set of systems, with their own Exporters and Importers, where each one is required to exchange information with other systems, in order to be able to achieve its goal. Given this assumption, an Interoperability Scenario is defined as a set of systems as shown in (5.2.11).

$$\begin{aligned} \mathrm{Z} &= \text{Interoperability Scenario} \\ &= \{\mathrm{S}_1, \mathrm{S}_2, \dots\} \end{aligned} \tag{5.2.11}$$

The Number of Systems in an Interoperability Scenario is expressed by the equation depicted in (5.2.12). The Number of Systems (N_S) is denoted by the cardinality of the Interoperability Scenario set.

$$\begin{aligned} N_S &= \text{Number of Systems} \\ &= |Z| \end{aligned} \tag{5.2.12}$$

And the Number of Data Formats is straightforwardly expressed by (5.2.13). The number of data formats (N_F) represents the total number of data formats existing in an interoperability scenario and is defined as the cardinality of the Data Formats set.

$$\begin{aligned} N_F &= \text{Number of data Formats} \\ &= |F| \end{aligned} \tag{5.2.13}$$

In an Interoperability Scenario, there is also the need to know the Modifiable and Non-Modifiable Systems. So, the set of all non-modifiable systems of an interoperability scenario is denoted by $Z^{\times}$, i.e. a zed with a cross in superscript for non-modifiability. And the set of all modifiable systems inside an interoperability scenario is denoted by $Z^{\surd}$, i.e. a zed with a checkmark in superscript. The two of them are expressed by (5.2.14) and by (5.2.15).

$$\begin{aligned} Z^{\times} &= \text{Non-Modifiable systems of an Interoperability Scenario} \\ &= \{S_1^x, S_2^x, \dots\}, \qquad x \in M, x = \times \end{aligned} \tag{5.2.14}$$

$$\begin{aligned} Z^{\surd} &= \text{Modifiable systems of an Interoperability Scenario} \\ &= Z - Z^{\times} \end{aligned} \tag{5.2.15}$$

It should be noticed that Z – zed without any symbol whatsoever in superscript – still qualifies the Interoperability Scenario, i.e. the set of all systems, and not only the set of those systems that are modifiable.

5.2.3 Mediator

The Mediator is the agent that acts as the central hub for interoperability support in the Hub-and-Spoke interoperability approach, and it is expressed by (5.2.16). It is important to notice that the Mediator (*H* - from Hub) does not have an identifier number, and this is due to being a central entity within the environment that aids the interaction between systems.

$$\mathrm{H} = \mathrm{Mediator} \tag{5.2.16}$$

The Mediator needs to a have a set of Adapters that enable it to Read or Write the information being exchanged from or to specific Data Formats. The Data Formats, mathematically represented by the equation present in (5.2.2), specify how data is organised, represented and structured.

So, the mediator needs two types of Adapters: a Reader, that enables the mediator to read the information in a specific Data Format; and a Writer, that enables the mediator to write the information in a specific Data Format. Readers and Writers are expressed by (5.2.17) and (5.2.18), respectively.

$$\mathrm{r}_{(\mathrm{F_x})} = \text{ Reader of Data Format } \mathrm{F_x} \tag{5.2.17}$$

$$\mathrm{w}_{(\mathrm{F_y})} = \text{ Writer of Data Format } \mathrm{F_y} \tag{5.2.18}$$

In order to mediate the exchange of information between several systems using heterogeneous data formats, the Mediator must possess several readers and writers, so sets of adapters need also to be formalised. They are being mathematically expressed by (5.2.19) and (5.2.20).

$$\begin{aligned} \mathrm{R} &= \text{ Set of Readers} \\ &= \{\mathrm{r}_{(\mathrm{F_x})}, \mathrm{r}_{(\mathrm{F_y})}, \dots\} \end{aligned} \tag{5.2.19}$$

$$\begin{aligned} \mathrm{W} &= \text{ Set of Writers} \\ &= \{\mathrm{w}_{(\mathrm{F_x})}, \mathrm{w}_{(\mathrm{F_y})}, \dots\} \end{aligned} \tag{5.2.20}$$

Graphically, the mediator is represented as a circle and has its own adapters, as depicted in Figure 58. Readers and writers are both represented as an irregular pentagon, just like an exporter and importer, but placed over the mediator's boundary, pointing towards the inside (if it is a reader) or the outside (if it is a writer) of the circle, symbolising the flow of data to/from the mediator. As each reader and writer refer to a specific data format, each of the referred data formats are graphically represented as rounded rectangles and connected to the reader or writer by a dashed line, as shown in Figure 58.

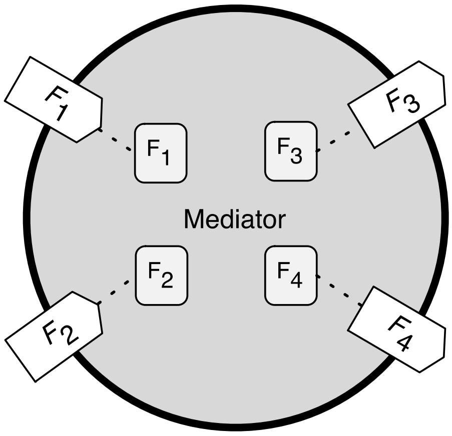


Figure 58: Graphical representation of the mediator with readers, writers and data formats

Besides these adapters, the Mediator needs functions that link different Data Formats, the <u>Interoperability Functions</u>. These Interoperability Functions, expressed by (5.2.21), specify all the actions that must be executed to perform the information transfer from a provider (F_p) to a consumer Data Format (F_c).

$$\mathrm{I}_{(\mathrm{F_p},\mathrm{F_c})} = \text{Interoperability Funtion from } \mathrm{F_p} \text{ to } \mathrm{F_c} \qquad (5.2.21)$$

Graphically, interoperability functions are showed as arrows (representing links) between different data formats, as illustrated in Figure 59.

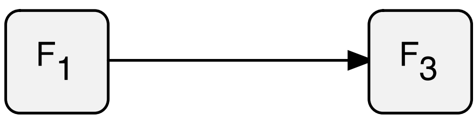


Figure 59: Graphical representation of an interoperability function from F_1 to F_3

As described before, there are several data formats, and therefore multiple interoperability functions exist, making the links between the heterogeneous data formats. So, there is the need for a mathematical definition of a set of interoperability functions, expressed by (5.2.22).

$$\begin{aligned} \mathrm{L} &= \text{Set of Interoperability Functions} \\ &= \left\{ \mathrm{l}_{(\mathrm{F_x},\mathrm{F_y})} \right\} : \forall\, \mathrm{F_x} \in \mathrm{F}, \forall\, \mathrm{F_y} \in \mathrm{F} \end{aligned} \qquad (5.2.22)$$

Therefore, in order to fulfil its purpose, the mediator needs to have readers (to understand data expressed in specific data formats), writers (to express data in specific data formats) and interoperability functions (to know how to exchange data between different data formats).

Knowing this, the mediator can be expressed as the union of the set of readers (R) with the set of writers (W) together with the set of interoperability functions (L). This is expressed by (5.2.23).

$$\begin{aligned} \mathrm{H} &= \text{Mediator} \\ &= \mathrm{R} \cup \mathrm{W} \cup \mathrm{L} \end{aligned} \qquad (5.2.23)$$

5.2.4 Interoperability

In order to interoperate different data formats, the mediator uses interoperability functions that specify how to exchange information between one specific data format and another. However, knowing the set of interoperability functions (L) available, the mediator can infer on how to achieve interoperability between data formats that do not have an interoperability function between them, by "discovering" paths between the data formats and using several different interoperability functions (which can be unlimited in number) in a row, and that way provide an interoperation between two data formats "indirectly".

These paths are mathematically expressed by (5.2.24), where the first case is 1, that happens when the provider data format is equal to the consumer data format, which means that no interoperability function is needed in order to interoperate them. The second case represents a path from F_p to F_c using a "direct" interoperability function ($l_{(F_p,F_c)}$). The third case represents a set of interoperability functions that are used in sequence, to interoperate data format F_p with F_c, using other data formats in between.

$$p_{(F_p,F_c)} = \begin{cases} \{1\}, & F_p = F_c \\ \begin{cases} \{l_{(F_p,F_c)}\}, & l_{(F_p,F_c)} \in L \\ \{l_{(F_p,F_a)}, l_{(F_a,F_b)}, \dots, l_{(F_x,F_y)}, l_{(F_y,F_c)}\}, & l_{(F_p,F_c)} \notin L \end{cases}, & F_p \neq F_c \end{cases} \quad (5.2.24)$$

As described, a path is nothing more than a set of interoperability functions (one or more) enabling the exchange of information between two data formats. So, and in order to simplify the graphical representation, the paths are represented as interoperability functions, i.e. as arrows between data formats. Figure 60 represents two different possibilities for a path between data format F_1 and F_4. The first is a path with only one interoperability function from F_1 to F_4; the second is a path composed by two interoperability functions, the first from F_1 to F_3, followed by one from F_3 to F_4.

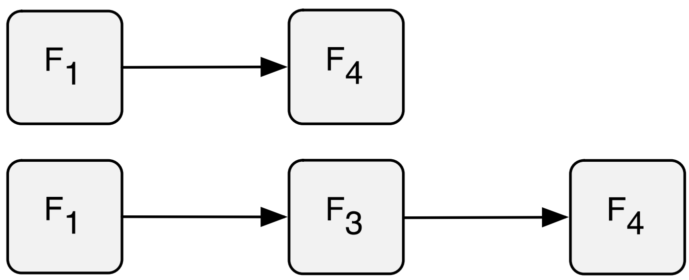


Figure 60: Two possible "paths" between F_1 and F_4

Each path between different data formats is expressed in (5.2.24), and as each interoperability function can be part of many different paths, there is the need to mathematically represent the set of all paths that can be "discovered" by using the available interoperability functions. This set of paths (*P*) is represented in (5.2.25).

$$\begin{aligned} P &= \text{Set of Paths} \\ &= \{p_{(F_x,F_y)}\} : \forall\, F_x \in F, \forall\, F_y \in F \end{aligned} \quad (5.2.25)$$

In order for the mediator to fulfil its purpose, i.e. mediate interoperability between two systems, it must possess a reader enabling it to read the source data format ($r_{(F_p)}$), a writer enabling it to write to the target data format ($w_{(F_c)}$), and a set of interoperability functions (path) that enable the data transfer from the source data format to the target one ($p_{(F_p,F_c)}$). Mediation is expressed by (5.2.26).

$$\begin{aligned} m_{(F_p,F_c)} &= \text{ Mediation between data format } F_p \text{ and } F_c \\ &= \left(r_{(F_p)}, p_{(F_p,F_c)}, w_{(F_c)}\right) : \exists \left(r_{(F_p)} \in H \wedge p_{(F_p,F_c)} \in P \wedge w_{(F_c)} \in H\right) \end{aligned} \quad (5.2.26)$$

Each mediation starts with the reader for a provider data format ($r_{(F_p)}$) and ends in the writer of the consumer data format ($w_{(F_c)}$), using a set of interoperability functions - path - in order to interoperate both data formats ($p_{(F_p,F_c)}$). Figure 61 presents examples of mediations from data format F_1 to F_3 and F_4 and from data format F_2 to F_3 and F_4. These mediations include the readers, paths between data formats, and writers.

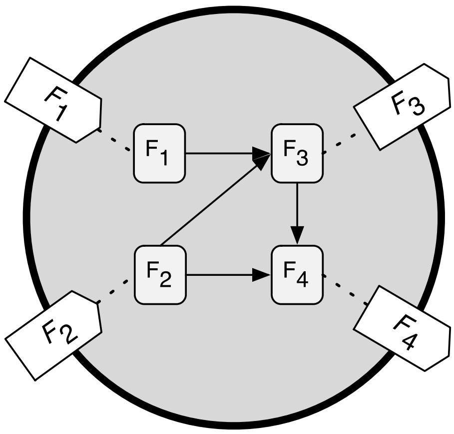


Figure 61: Mediation between data formats

All possible mediations available to the mediator are represented by a set of mediations between all data formats known within the interoperability scenario, and are expressed by (5.2.27).

$$\begin{aligned} M &= \text{ Set of Mediations} \\ &= \left\{m_{(F_x,F_y)}\right\} \forall F_x \in F, \forall F_y \in F \end{aligned} \quad (5.2.27)$$

Also the flow of information exchange between two given systems is seen from the data producer system (the data format used by the exporter of the system providing the data – F_p) to the data consumer system (the data format used by the importer of the system consuming the data – F_c). In this context, an Interoperation is described by the ability of two systems to exchange data using the mediator, and it represents the flow of information from the provider to the consumer system, defined as (5.2.28).

$$\begin{aligned} t_{((S_p,F_p),(S_c,F_c))} &= \text{Interoperation of provider } (S_p, F_p) \text{ to consumer } (S_c, F_c) \\ &= \left(e_{(S_p,F_p)}, m_{(F_p,F_c)}, i_{(S_c,F_c)}\right) : \exists \left(e_{(S_p,F_p)} \in S_p \right. \\ &\quad \left. \wedge\, m_{(F_p,F_c)} \in M \wedge i_{(S_c,F_c)} \in S_c\right), S_p \in Z, S_c \in Z \end{aligned} \qquad (5.2.28)$$

An interoperation is graphically represented as a combination of a directed arrow matching an exporter and a reader within the mediator; a set of arrows establishing a "path" between the provider data format reader and the consumer data format writer; and, finally, an arrow matching the writer within the mediator to the importer in the consumer system, as exemplified in Figure 62.

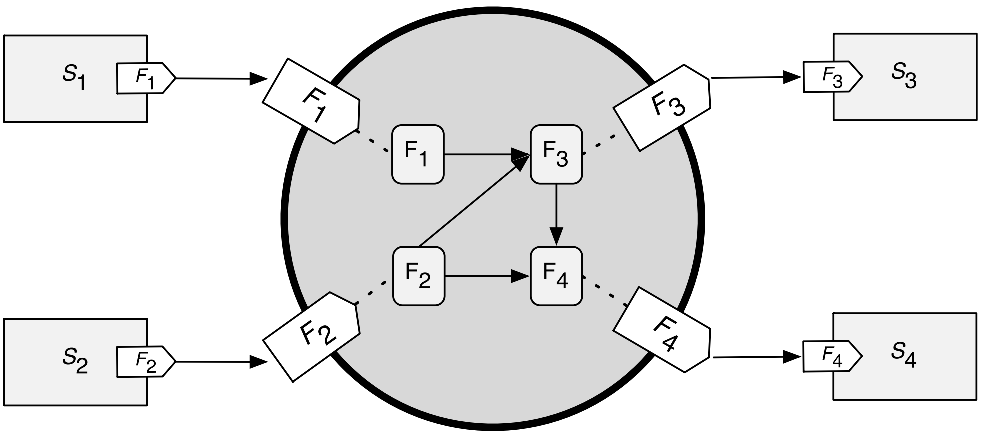


Figure 62: Graphical representation of the interoperation between S_1, S_2, S_3 and S_4

And, the set of all possible interoperations (*T*) in an interoperability scenario, which is the set of all possible flows of information between every system, is given by (5.2.29).

$$\begin{aligned} T &= \text{Set of all Interoperations within the Interoperability Scenario} \\ &= \left\{t_{((S_a,F_a),(S_b,F_b))}\right\} \forall S_a \in Z, \forall S_b \in Z, \forall F_a \in F, \forall F_b \in F \end{aligned} \qquad (5.2.29)$$

One system may be interoperable with another system, by exporting and/or by importing via several different formats, i.e. there can be more than one interoperation (in one same direction) between both systems. So, the set of all interoperations from a provider to a consumer is as represented by (5.2.30).

$$\mathrm{T}_{(\mathrm{S_p},\mathrm{S_c})} = \text{Interoperations from provider } (\mathrm{S_p}) \text{ to consumer } (\mathrm{S_c}) \text{ systems} \qquad (5.2.30)$$
$$= \left\{ \mathrm{t}_{((\mathrm{S_p},\mathrm{F_p}),(\mathrm{S_c},\mathrm{F_c}))} \right\} : \mathrm{F_p} \in \mathrm{F}, \mathrm{F_c} \in \mathrm{F}$$

Consequently, the number of interoperations between two systems – from provider system to consumer system – is (5.2.31).

$$\mathrm{C}_{(\mathrm{S_p},\mathrm{S_c})} = \text{Number of interoperations from provider } (\mathrm{S_p}) \text{ to consumer } (\mathrm{S_c}) \qquad (5.2.31)$$
$$= \left| \mathrm{T}_{(\mathrm{S_p},\mathrm{S_c})} \right|$$

Finally, Interoperability is said to exist from a provider system to a consumer system if, and only if, there is at least one interoperation between them; on the other hand, systems are not interoperable if there is no interoperations between them. The existence of Interoperability from a provider to a consumer system is set by (5.2.32).

$$\mathrm{X}_{(\mathrm{S_p},\mathrm{S_c})} = \text{Interoperability between a provider } (\mathrm{S_p}) \text{ to consumer } (\mathrm{S_c}) \qquad (5.2.32)$$
$$= \begin{cases} 1, & \mathrm{C}_{(\mathrm{S_p},\mathrm{S_c})} > 0 \\ 0, & \mathrm{C}_{(\mathrm{S_p},\mathrm{S_c})} = 0 \end{cases}$$

Building on this, and if required for a simple exemplification of an interoperability scenario, it is possible to express the existence of Interoperability between two systems by simply using the graphical notation depicted in Figure 63. That is, by drawing a directed edge (or arrow) with a circle in the middle (corresponding to the mediator) from the border of the provider system to the border of the consumer system, thus showing mediated interoperability from provider to consumer, only.

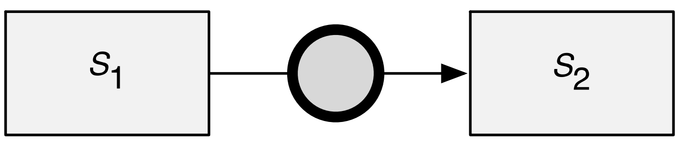


Figure 63: Graphical notation for describing Interoperability between S_1 and S_2

5.3 Algorithm to Maximum Interoperability

The main goal of the Maximum Interoperability algorithm is to allow the assessment of a few metrics from the result of the application of the Hub and Spoke Interoperability approach to an interoperability scenario. So, this algorithm is designed to measure the least Number of Interoperability Artefacts (NIA) required for a specific scenario to reach the full interoperability state, i.e. to reach a state where each system is able to successfully transfer data to all the other systems that compose the scenario. This full interoperability state corresponds to the situation where the value of the interoperability coverage is 1.

In the context of H&S Interoperability approach, the application of an interoperability artefact corresponds to the definition and deployment of an Interoperability Function (IF) between two data formats existing within the interoperability scenario under study. Hence, the goal of the Maximum Interoperability algorithm is to determine the smallest set of Interoperability Functions (IFs) that enables each system to import data from all the systems that export data.

As input, the Maximum Interoperability algorithm requires one representation of the scenario to be analysed. This representation of the scenario must model and distinguish the data formats that each system supports either for export or import data. Such representation is required due to the fact that IFs are unidirectional by concept, requiring the distinction between the data formats used to export data and the ones used to import data in order to optimise the set of IFs to be used.

The execution of the Maximum Interoperability algorithm results in four outputs. The first, and the most important, is the set of IFs applied to the scenario during the execution of the algorithm. This set indicates the IFs that must be developed and applied to the scenario so it reaches the interoperability state determined by the algorithm. Alongside with the set of applied IFs are presented the metrics required for the evaluation of the interoperability state determined by the algorithm. These metrics are: the Number of Interoperability artefacts within the scenario after the application of the Hub–and-Spoke Interoperability approach, the Interoperability Coverage and the Interoperability Absolute Coverage.

The Maximum Interoperability algorithm is represented as a flowchart in Figure 64, starting with the determination of the initial NIA in the scenario, in accordance to the concepts of the Hub-and-Spoke Interoperability approach.

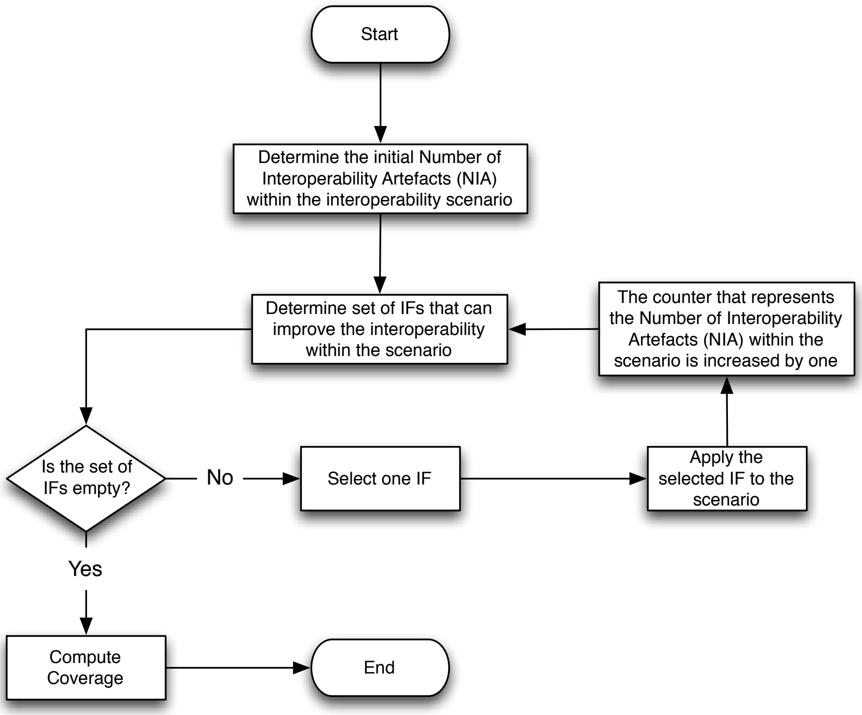


Figure 64: Flowchart representing the Maximum Interoperability Algorithm

The Maximum Interoperability algorithm can be abstracted as a loop, where a set with all the IFs that can improve interoperability within the scenario is continuously determined. This set will then be subject to a selection algorithm in order to select the IF that produces the greatest impact in the scenario's interoperability state, i.e. the IF that greatly improves interoperability coverage. Afterwards, the selected IF will be applied to the current scenario, producing a new scenario that will then be analysed in order to determine a new set of IF that can improve interoperability within this new scenario. Whenever a cycle of the loop is completed, the counter representing the NIA is increased by one, as its cycle corresponds to the deployment of a new IF into the scenario. The loop goes around until the situation where interoperability in a scenario can no longer be improved. Once outside the loop, the algorithm computes the coverage, which is expected to be always equal to 1.

This algorithm can be viewed as a composition of five activities: (1) the determination of the initial NIA, (2) the determination of the set of all possible IFs, (3) the selection of the best IF, (4) the application of the selected IF in the scenario and (5) the computation of the coverage. The first activity (1) is described in Figure 65.

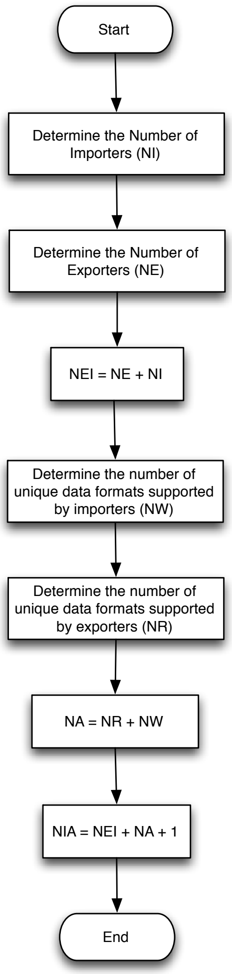


Figure 65: Flowchart describing the procedure to determine the initial number of interoperability artefacts (NIA) of a scenario (2)

This activity performs the count of the Number of Interoperability Artefacts (NIA) within the interoperability scenario. At this stage of the Maximum Interoperability algorithm, the interoperability artefacts considered for the count are: the importers and the exporters of the systems that compose the scenario; the mediator used to enable the interoperability between the systems; and each one of the adaptors that the mediator requires to be able to connect to each system in the scenario.

Activity (1), as represented in Figure 65, begins by determining the Number of Importers (NI) and the Number of Exporters (NE) within the scenario in order to know how many exporters and importers exist, as represented in the flowchart by NEI.

After this step determining the Number of Adaptors (NA) required in the mediator takes place. The number of different data formats supported by importers determines the Number of Writers (NW) needed in the mediator, as it needs to be able to structure data in all the data formats supported by importers. Likewise, the number of different data formats supported by exporters determines the Number of Readers (NR). The sum of these two counts (NR and NW) determines NA.

Once these two parameters are determined (i.e. the Number of Importers and Exporters and the Number of Adaptors), it is now possible to determine the initial NIA through the sum of these parameters and by adding the value ‘1’. This adding of ‘1’ is performed to take into account the existence of the mediator which, being an interoperability artefact in itself, must also be counted.

After determining the initial NIA, the Maximum Interoperability algorithm enters in the next stage, characterized by a loop-like behaviour. The first step of the loop is activity (2), whose flowchart is presented in Figure 66.

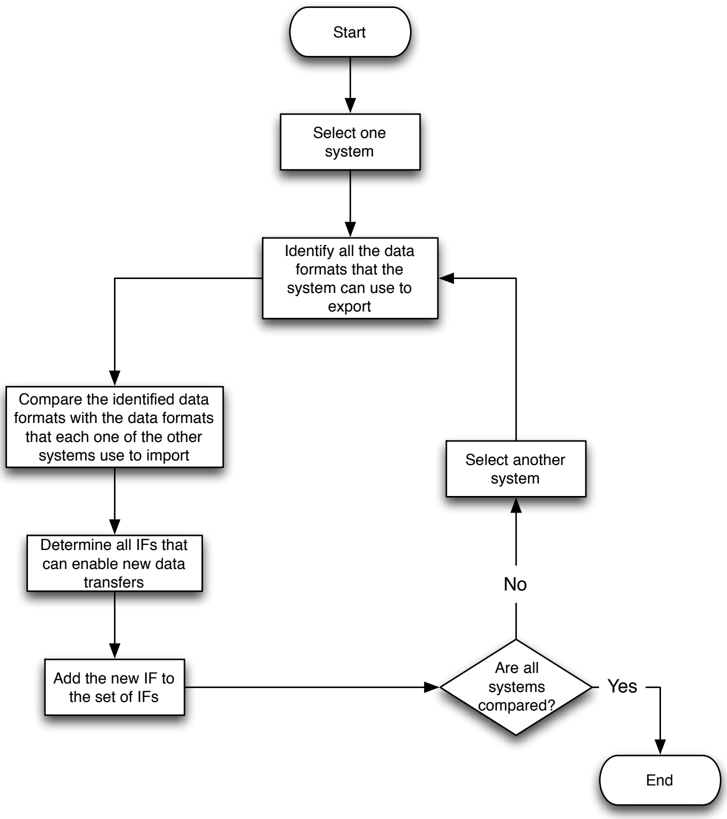


Figure 66: Flowchart describing the procedure of identification of the IF that can improve interoperability (2)

The objective of this activity is to identify all the IFs that can improve interoperability within a scenario. To reach this objective, a representation of the scenario to be analysed is needed as input. The output of this activity is a set composed by all the identified IFs. In order to identify all the IFs that can improve interoperability in a specific scenario, an analysis of all systems based on the data formats that each one supports, is required. This analysis is performed by comparing the data formats that each system uses to import data with the data formats used by the other systems to export data.

For one IF to be able to improve interoperability, it must enable one or more systems to import data from "unreachable" systems, i.e. systems whose exporters support data formats that are unreadable by the importers of other systems. Thus, the data formats supported by the importers of each system must be compared with the data formats supported by the exporters of the other systems, one by one. This comparison allows the identification of the systems that can currently communicate based on the matching of data formats in common. When systems that cannot communicate are identified, the data formats supported by the exporters and importers of the systems are combined, generating IF proposals. This process is exemplified in Figure 67.

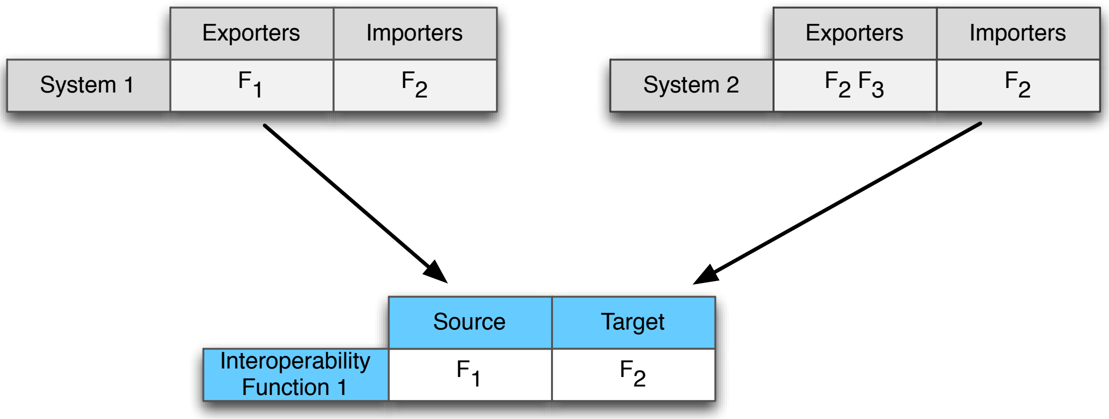


Figure 67: Example of the IF proposal process

In this example, there is a scenario composed by two systems: System 1 and System 2. Each system has two sets of data formats: one to represent the data formats that it can use to export, and another to represent the data formats that it can use to import data. System 1 can export data using the F_1 and can only import data in the F_2, whereas System 2 can export data using F_2 or F_3 and import data written in the data format F_2. As previously defined, the scenario is analysed from the point of view of the importers, inferring from which systems a specific system can export. In the scenario represented, System 1 can import data from System 2 using F_2, but System 2 cannot import data from System 1 because there is no data format in common between the exporters of System 1 and the importers of System 2. In this case, interoperability within the systems would improve with the addition of an IF that would convert data represented in F_1 to data in F_2, as shown in Figure 67.

Once the scenario is analysed and all the IFs that can improve interoperability within the scenario are identified, it is time to describe activity (3), i.e. the selection of the best IF. The flowchart corresponding to this activity is described in Figure 68. This activity has the objective of selecting the IF that mostly improves interoperability from within a set of IFs; therefore these elements are, respectively, its output and input. This selection activity starts by determining the number of systems that each IF in the set will connect if it is applied to the scenario. This number of systems is represented in a counter, one for each IF, where the IF associated with the counter with the highest count is the one that mostly improves interoperability within the scenario.

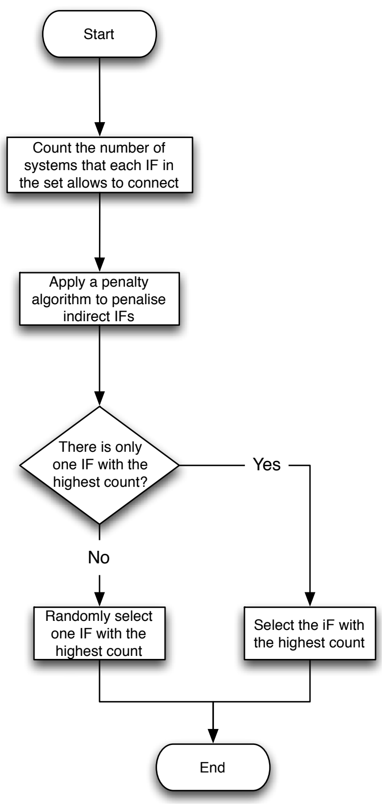


Figure 68: Description of the IF selection procedure (3)

As the best IF is the one with the highest count of new system connections, a question arises: What to do when there is more than one IF with the highest count? In order to break this tie, two rules are applied. The first rule penalizes counters associated to IFs that are based on IFs previously applied to the scenario. This penalization is applied in order to prefer the deployment of IFs that directly improve interoperability in the scenario, instead of a composition of IFs. The preference of such IFs is due to the fact that by depending on more direct IFs reduces the impact in the overall interoperability of the scenario caused by an incorrect maintenance of the IF artefacts.

This penalization consists in subtracting the counter associated to the IFs that use data formats used by IFs previously applied to the initial scenario. This refers to either the source or the target data format. The amount subtracted to the counter, the penalty value, must be less than 1, because it shouldn't have the same weight in the selection algorithm as the communication with a system.

If, after applying the penalty value to the counter of each IF (where it is applicable), there is still more than one IF with the highest counter, then the second rule is applied. This rule consists in a random selection of an IF from the subset of IFs with the highest presence counter.

After the selection of the best IF, it needs to be applied to the scenario, which corresponds to activity (4). The deployment of an IF into a scenario involves the addition of the support of new data formats into importers and exporters, as described in the flowchart represented in Figure 69. This addition of data formats results in a new scenario as some or all systems have virtually gained support of new data formats, other than the initial ones.

To produce this new scenario three things are needed: the current scenario representation, the set of all IFs already applied until that moment, and the IF selected to be applied. The set of previously applied IFs and the IF selected in activity (3) are grouped in one new set, forming the set of IFs to be applied.

The deployment procedure consists in the application of all IFs in this set over the scenario, several times, until no data format support is added to any system that composes the scenario. By repeatedly applying all the IFs in the set, it is possible to identify all the indirect support of data formats and to ensure that the order in which each IF is applied is not relevant.

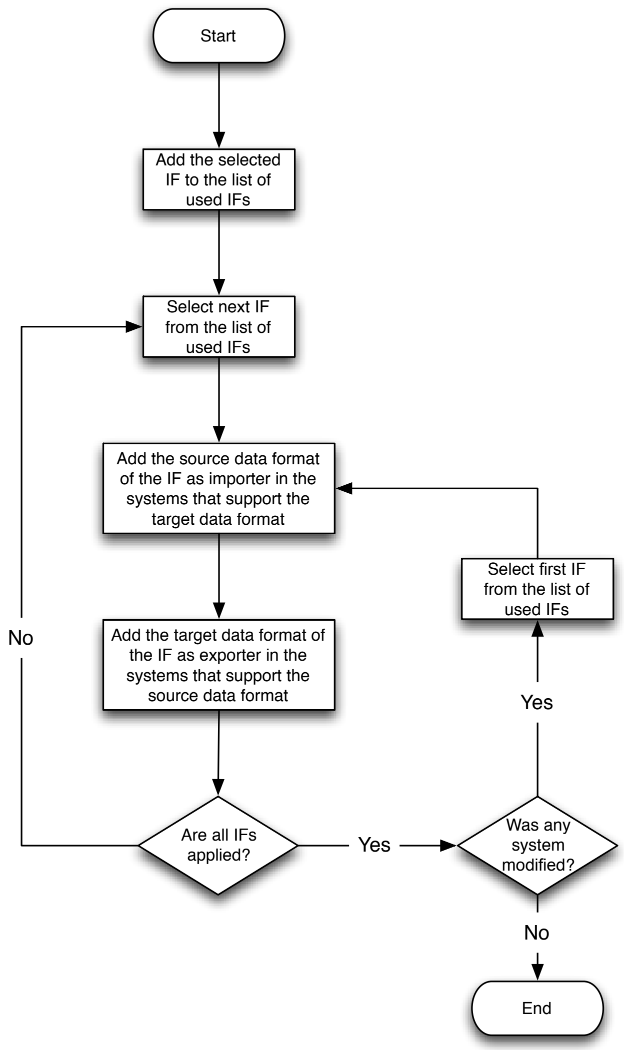


Figure 69: Flowchart describing the IF deployment procedure (4)

The deployment of one IF can potentially affect all data formats supported by the exporters and importers of all systems that compose the scenario. When an IF is applied to a scenario, all systems that use the source data format to export data will also be able to export using the target data format. In a similar way, all systems capable of importing data in the target data format will also support the import of data in the source data format.

An example of the deployment of an IF over a scenario is presented in Figure 70, composed of two systems, where System 1 can import data using F_2 and export data using F_1; and System 2 can export and import data expressed in data format F_1.

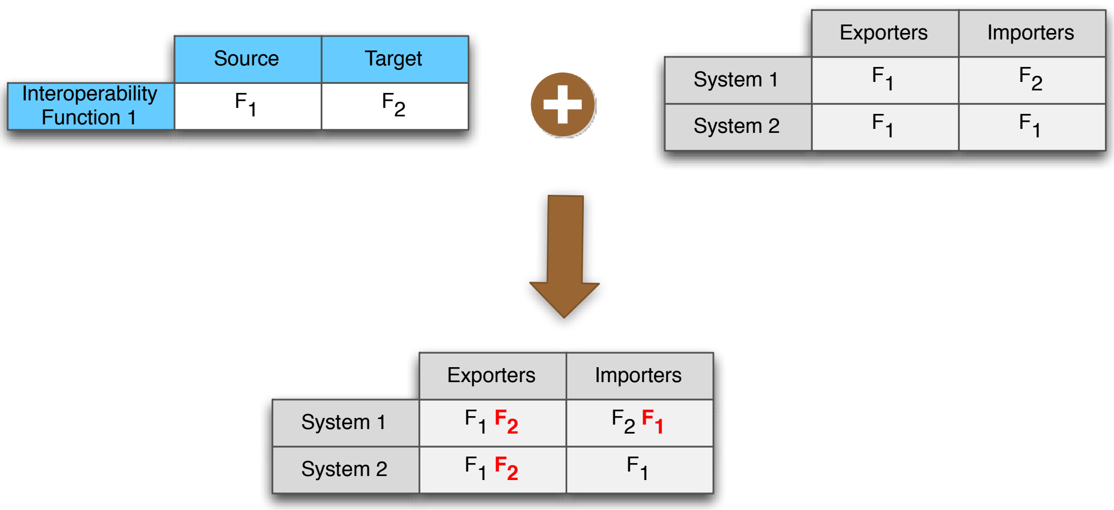


Figure 70: Example of an IF deployment

Analysing interoperability within this scenario it can be concluded that System 2 can import data from System 1 (using F_1) but the opposite flow of data is not possible, because System 2 only supports the export of data using F_1 and System 1 only imports data expressed in F_2. This issue is addressed by deploying Interoperability Function 1 into this scenario.

The deployment process of an interoperability function consists in two steps: the addition of the target data format (F_2) to each set of exporters already supporting the source data format (F_1), and the addition of the source data format to each set of importers supporting the target data format. In this example, the execution of these two steps results in the introduction of the data formats in red. It is important to remember that the new export and import capabilities do not correspond to the addition of new importers and exporters to the systems, but are provided by the mediator.

Activities (2), (3), and (4) are continuously and sequentially executed until the scenario reaches a state where it can no longer be improved, i.e. no IF proposal is suggested. Once the scenario reaches this state, the corresponding interoperability coverage is computed. This step corresponds to activity (5), whose algorithm is described in the flowchart presented in Figure 71. In this activity, the two coverage kinds are computed:

Interoperability Coverage and Interoperability Absolute Coverage. Both kinds of coverage require determining the current number of systems that are interoperable, referred to in Figure 71 as *CurrentConnections*. This parameter is given by the sum of the number of systems that each system can import data from.

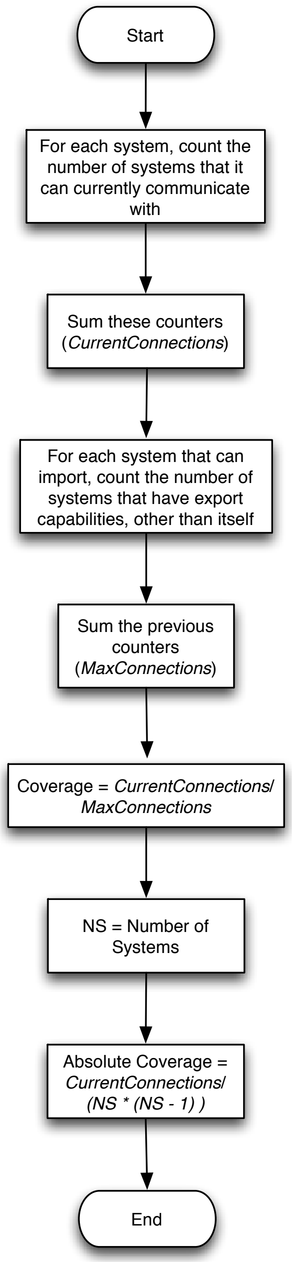


Figure 71: Flowchart describing the algorithm used in activity (5)

Interoperability Coverage consists in the relation between this parameter and the maximum number of connections possible in the scenario, taking into consideration the communication purpose of each system. The communication purpose is relevant because some systems may not be designed in export or import data, and one can expect that a system will do something that it was not designed to do. This maximum number of connections is known as *MaxConnections*, and is determined by counting the number of systems that can export for each system that can import.

The Interoperability Absolute Coverage consists in the relation of *CurrentConnections* with the total number of connections possible in a scenario with Ns number of systems. This total number of possible connections is determined by using the formula to determine the number of links in a fully connected directed network.

5.4 Simulation

To understand the behaviour of Hub-and-Spoke Interoperability, a number of simulations are performed. These simulations will demonstrate the behaviour of an interoperability scenario where H&S approach is used. This behaviour will be demonstrated in the form of the relevant Interoperability Metrics that characterize a good or bad interoperability in a scenario, and they are the Number of Interoperability Artefacts (NIA) and the Interoperability Coverage.

These Metrics are the information collected by the simulations, that will be analysed to understand the impact that H&S approach has on achieving interoperability in a scenario. Firstly, the Interoperability Coverage represents an "already connected state", meaning that the coverage value is the relation between the number of systems already interoperable and the maximum number of systems. Secondly, and while coverage gives the notion of interoperability in a given scenario, it is important to know how many modifications the interoperability scenario needs in order to achieve this coverage, and that is what the NIA represents.

To simulate the behaviour of the H&S approach in interoperability scenarios, there is the need of an algorithm capable of putting into practice the H&S concepts, which was previously defined as Algorithm to Maximum Interoperability, due to its objective to always try to maximise interoperability in a scenario. However, in order to enable the algorithm's execution, a set of inputs is needed, consisting in the characterisation of the interoperability scenario.

Each interoperability scenario (algorithms input) is generated by using the same methodology as used with P2P interoperability, to better understand the differences between the two approaches. The interoperability scenarios are generated based on scenario characteristics such as: (1) the number of different systems present in the environment, (2) the percentage of heterogeneous data formats being used by those systems, and certain system properties, such as (3) the percentage of systems capable of exporting data, (4) the percentage of systems importing data and (5) the percentage of systems capable of being modified, meaning that they can be changed in order to increase their importing or exporting capabilities.

To fully understand the H&S behaviour, it must be tested with several different interoperability scenarios, ideally varying each parameter to observe its evolution while changing each scenario parameter. However, this would reach an enormous amount of iterations and it would be almost impossible to compare and interpret all of them. So, a small set of values was defined as possible, when generating new interoperability scenarios: {25%; 50%; 75%}. All parameters defined as percentages (data formats, exporters, importers and modifiability) are bound to use one of these values.

Another important aspect is that systems that do not import or export are not considered, therefore the percentage of exporters and importers, must be equal to, or higher than, 100%, so each system possesses at least one importer or one exporter. So, there are only 6 possible combinations for exporters and importers, which are: {(E:25%, I:75%); (E:50%, I:50%); (E:50%, I:75%); (E:75%, I:25%); (E:75%, I:50%); (E:75%, I:75%)}.

With this set of possible parameters, the number of simulations needed decreases substantially. For the combinations of importers and exporters there are only 6 possible values, and for then percentage of data formats there are 3 possibilities {25%; 50%; 75%}, resulting in 18 iterations by varying these three parameters. The system modifiability parameter is not used in the H&S Interoperability approach simulation, because this approach uses a mediator to enable interoperations between systems, so it is irrelevant if the system can be modified or not, because the modifications are applied in to the mediator and not to the systems. It is only considered here because it was an input of the interoperability scenario generator. Now, the only thing missing is the number of systems to consider, so it was defined that the range would be between 5 and 115, using only multiples of 5.

Another important thing to be defined is the number of iterations of the same execution (using the exact same parameters), in order to remove the influence of the randomness of the test cases' scenarios. So a study was carried out, consisting in repeating the simulations of the same scenario (using the exact same inputs) 1000 times, in order to understand how many iterations were needed. The output was the NIA cumulative mean.

The NIA cumulative mean graphic, represented in Figure 72, was plotted, using the number of simulations as the X-axis and the Number of Interoperability Artefacts (NIA) as the Y-axis. The graphic in Figure 72 shows that every scenario needs to be iterated 250 times so that the randomness of the simulation is withdrawn from the result, thus representing a real-world environment example.

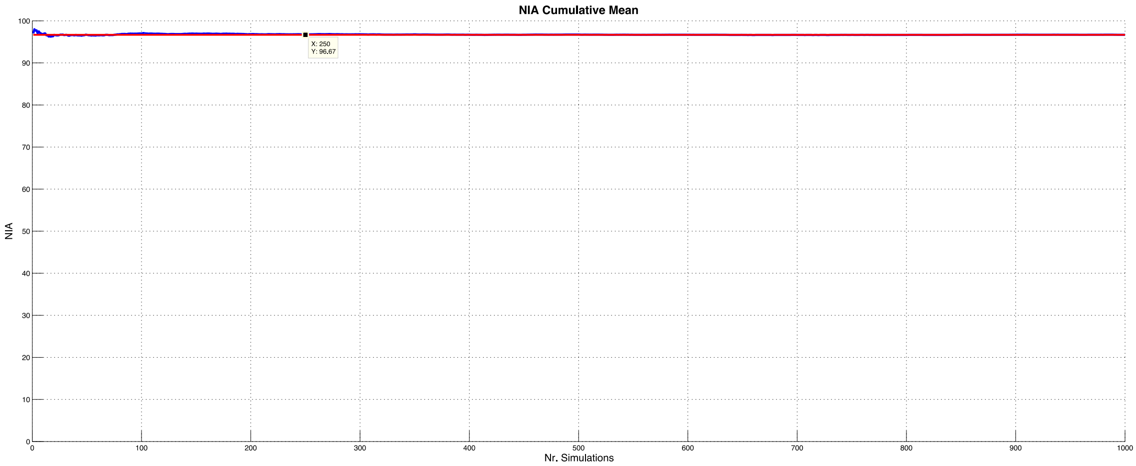


Figure 72: NIA cumulative mean

Having defined the possible values for each parameter, the simulations resulted in 18 different graphics, all present in Appendix. Each graphic has the number of systems in the X-axis, and both the NIA and the coverage in the Y-axis; and, for each value of number of systems, it depicts three values: minimum, average and maximum number of NIA for the 250 iterations of the simulation, using the exact same parameters.

By looking at the 18 resulting graphics side by side, it was noted that the graphics where the exporters and importers had the combinations (25%, 75%) and (75%, 25%) were exactly the same. This is due to the fact that the Hub-and-Spoke Interoperability approach relies on connections to and from the mediator, regardless of whether there are more or less connections to the mediator than from the mediator. This can also be seen when comparing the graphics with the exporters and importers combinations of (50%, 75%) and (75%, 50%). By eliminating these two sets of graphics, the number of graphics fell to 12.

Another aspect noted by the observation of the 18 initial graphics was that the coverage is completely independent of the variation of the data format percentage. This means that, as the coverage is the same for every value of data format percentage, the results of that variation can be grouped in a single graphic, representing the common coverage and the three NIA variations. This allows a reduction of the number of graphics to be presented to only 4, which are shown in the following figures.

The graphics shown in the following figures present the simulation results differentiated according to the Exporter and the Importer values. Figure 73 depicts the results of the H&S approach, using 25% Exporters and 75% Importers, while Figure 74 shows the simulation results, using 50% Exporters and 50% Importers. The graphic in Figure 75 shows the effect of using 50% Exporters and 75% Importers. Finally, Figure 76 depicts the simulations outcome, using as input 75% Exporters and 75% Importers. Each of them shows NIA results for the three values of data formats (25%, 50% and 75%), the coverage to max possible, and the coverage to achieve a fully connected state.

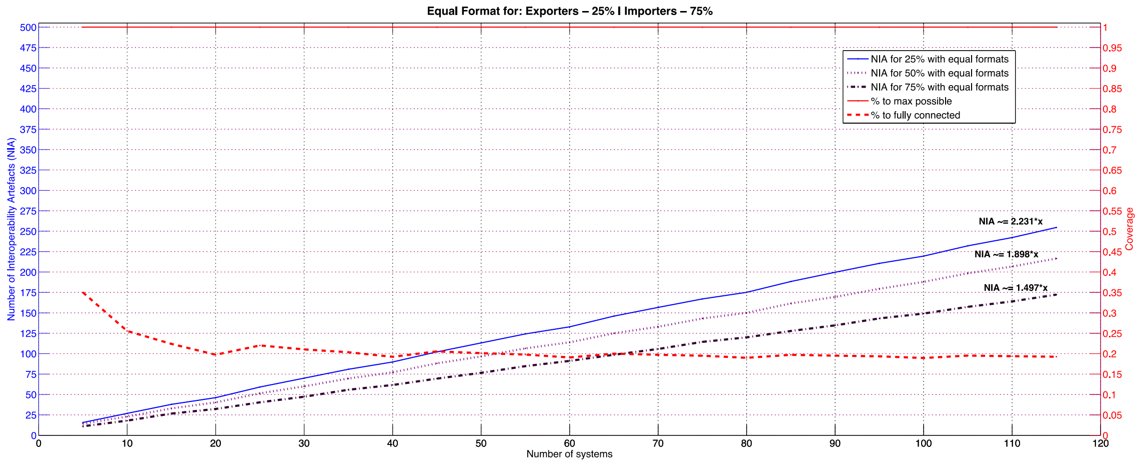


Figure 73: Simulation results grouped by Exporters 25% and Importers 75%

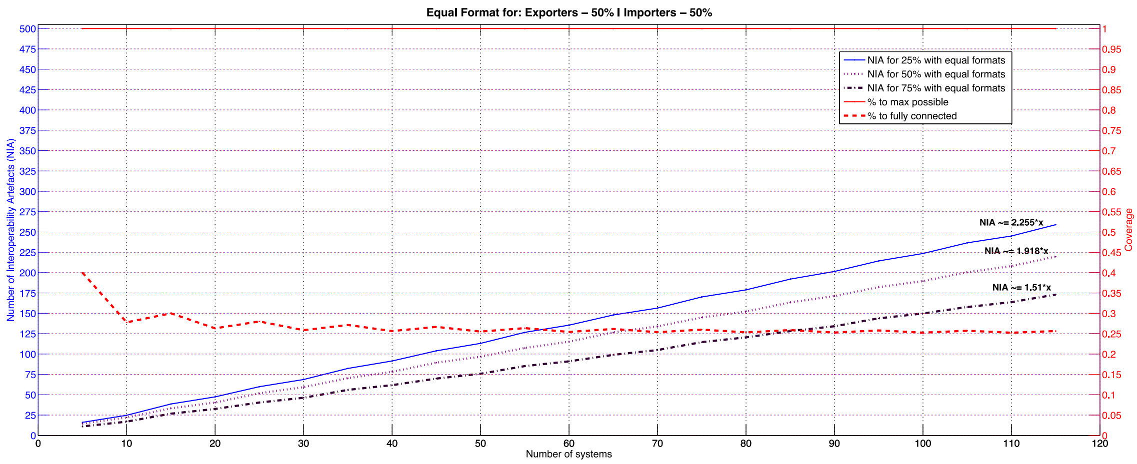


Figure 74: Simulation results grouped by Exporters 50% and Importers 50%

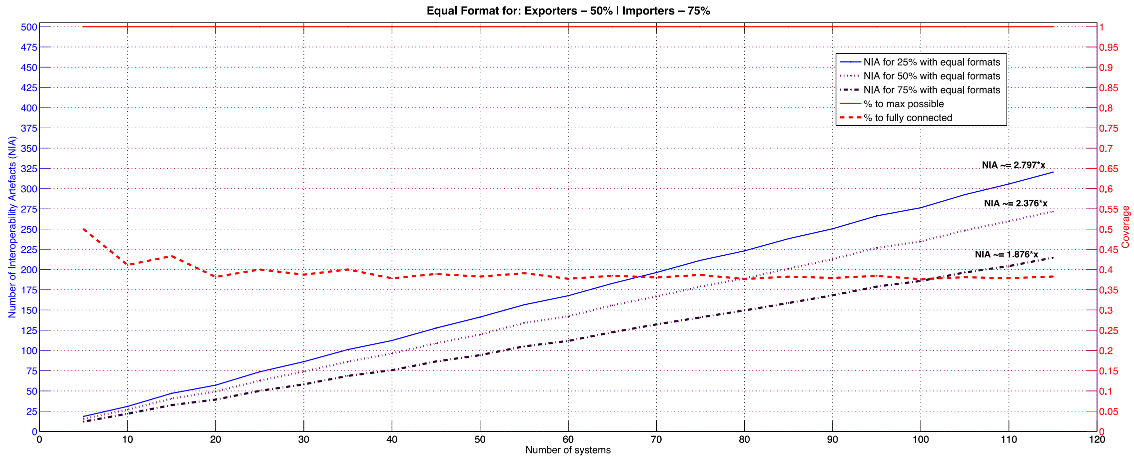


Figure 75: Simulation results grouped by Exporters 50% and Importers 75%

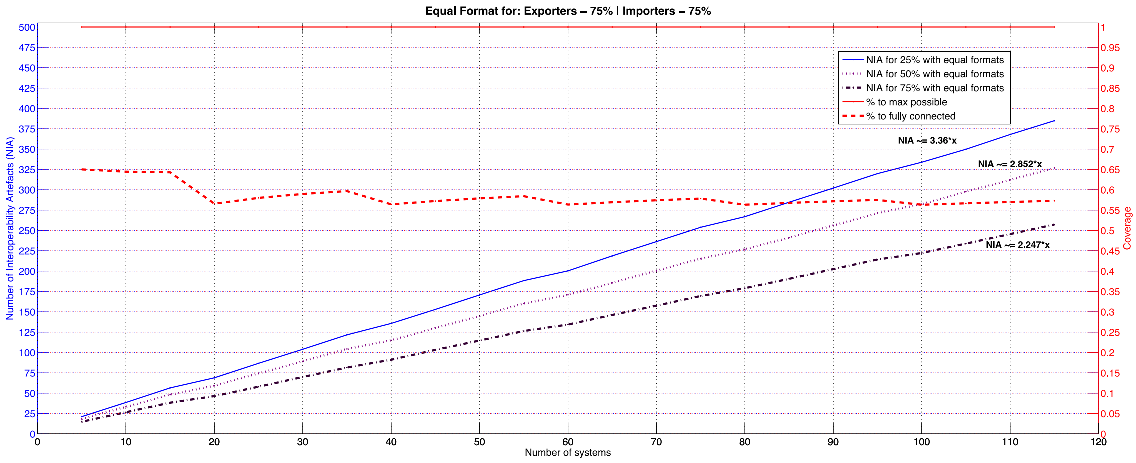


Figure 76: Simulation results grouped by Exporters 75% and Importers 75%

An analysis of the graphics shows that, in all cases, the NIA increases linearly with the increase in the number of systems. It also points out that the number of NIA in each graphic decreases as the percentage of data format increases. The overall Number of Interoperability Artefacts also increases as more systems have exporting and/or importing capabilities, which is due to the fact that more importers and exporters mean more possible connections between systems, therefore requiring more NIA.

As for coverage, the main focus is the fact that the coverage to max possible is always 100%, meaning that at all times all systems can connect to all others (excluding those which possess only importers among themselves, and those who only possess importers). Coverage to fully connected also increases with the increase of the number of systems exporting and/or importing.

5.5 Research Hypothesis

The Hub-and-Spoke (H&S) Interoperability approach as been defined as to address the fundamental challenges of data interoperability, highlighted when using the P2P interoperability approach. The H&S approach takes on the Coverage-Modifiability challenge via a mediated approach that facilitates interoperability between systems, especially of non-interoperable ones. And the H&S approach addresses the NIA-Connections challenge via interoperability compositions that reuse interoperability artefacts to render non-interoperable systems interoperable.

One of key problems of interoperability, as duly identified in the study of the P2P Interoperability approach, is exactly that the interoperability coverage is highly impacted by the number of non-modifiable systems present in an interoperability scenario. The H&S approach makes fully interoperable a given interoperability scenario especially making interoperable those systems that are non-modifiable. The other problem of interoperability is the quadratic progression of the Number of Interoperability Artefacts (NIA) to number of systems. The H&S approach makes it possible to efficiently accomplish interoperability in a scenario by establishing a linear progression of the NIA versus number of systems.

Summing up, the H&S approach – powered by mediated interoperability and interoperability compositions – might provide the leading edge to data interoperability that moves past the challenges posed by the P2P approach to interoperability. First, the proposed interoperability approach might establish the best interoperability, in an arbitrary data-exchange environment, using fewest resources when compared to P2P interoperability. Also, it looks possible to interoperate all systems, especially those that cannot be modified to support (import/export) other data formats, thus putting them all as participants in the data-exchange environment. These two important features might provide a general solution to the data interoperability problem, one that is efficient, effective and that able to scale to large-scale interoperability scenarios.

<u>Hypothesis</u>: The combined use of interoperability mediation and interoperability compositions (e.g. as in the Hub-and-Spoke Interoperability approach) enables to address the fundamental challenges of data interoperability, i.e. to achieve full interoperability coverage with a suitable number of interoperability artefacts.

6

Evaluation: Hub-and-Spoke vs. P2P Interoperability

"No amount of experimentation can ever prove me right;
a single experiment can prove me wrong."
– Albert Einstein

The definition of evaluation can be stated as the judgement on the value of a hypothesis with reference to defined criteria of this judgement. The criterion used in a scientific work is the research question. More general meaning of the term "evaluation", can be seen as a systematic survey of values or features of a given programme, activity or an object, taking into consideration the adopted criteria to enhance, improve or understand them better. Evaluation always is the study with an objective, which in this case is to ensure the proper behaviour of the H&S Interoperability approach.

The criteria used to evaluate the proposed solution, is in this work, the one that goes from the problem study, where a set of drawbacks of the typical interoperability approaches was identified:

- Coverage vs. Modifiability: P2P interoperability has a limitation on the coverage over the number of modifiable systems in the environment.

- NIA vs. Number Connections: P2P interoperability has a quadratic behaviour on the evolution of the NIA vs. the needed connections on a scenario.

The evaluation process on this work is done through benchmarking, which consists in assessing the effects of the solution via their comparison to previous works, such as in the case of this thesis, the typical used interoperability approach, which was identified to be the P2P interoperability. Possessing the comparison of the strengths and weaknesses of the approaches, a verdict of the behaviour of the solution is achieved.

The report of this evaluation process is detailed along four sub-sections. First is made a detailed comparison of the H&S against the P2P Interoperability doing a problem-oriented analysis following the previously identified interoperability problems and focusing on large-scale data interoperability scenarios. Then, a study of small-scale settings is also performed to assess the performance of the H&S approach also in such conditions. At the end, an overview of the two simulations and the verdict of the H&S proposed approach is finally given.

- Problem-oriented Analysis: For the H&S Interoperability to perform better that the P2P Interoperability, it must, at least, to overcome the identified problems in that approach, related to the coverage limitation on the modifiability of the systems, and the quadratic dependability of the NIA on the number of needed connections on the interoperability scenario.

- Small-Scale Performance: The H&S Interoperability approach is especially tailored for large-scale interoperability scenarios; so, this small-scale performance analysis is made, to understand if, even with a concept specifically designed for the large-scale, it can be used to address interoperability in small-scale data interoperability environments.

- Overview: This section provides an overview of the comparison made between the Point-to-Point Interoperability and the Hub-and-Spoke Interoperability, showing together the simulations of each one, so one can see the two approaches side to side.

- Verdict: Last, a verdict on the fitness of the Hub-and-Spoke Interoperability facing the current interoperability approach – the P2P Interoperability – is given, making it the section that states the performance of the H&S over the P2P.

6.1 Problem-oriented Analysis

For H&S Interoperability to perform better than P2P Interoperability, it must, at least, overcome the identified problems in that approach. Those problems are:

- Coverage vs. Modifiability: P2P interoperability has a limitation on the coverage. It depends on the number of modifiable systems - Modifiability. Modifiability has a quadratic relation with the coverage, and only when it is close to 100% is it possible to achieve full interoperability between systems.

- NIA vs. Number Connections: P2P interoperability has a quadratic behaviour regarding the evolution of the NIA vs. the needed connections in a scenario. Three "axes" have a big impact on the behaviour of the NIA, the Number of Formats, Modifiability and the number of Exporters and Importers in the system.

6.1.1 Coverage vs. Modifiability

The Coverage vs. Modifiability Study has already been made for the P2P Interoperability approach, so this subchapter will cover a summary of that study, plus an equivalent one for the proposed solution, the H&S Interoperability. So the goal is to see if the latter – H&S Interoperability – has the same problem, or whether it can overcome that limitation, as desired and designed. Figure 77 shows the summary of the result of that study, showing the behaviour of Coverage vs. Modifiability.

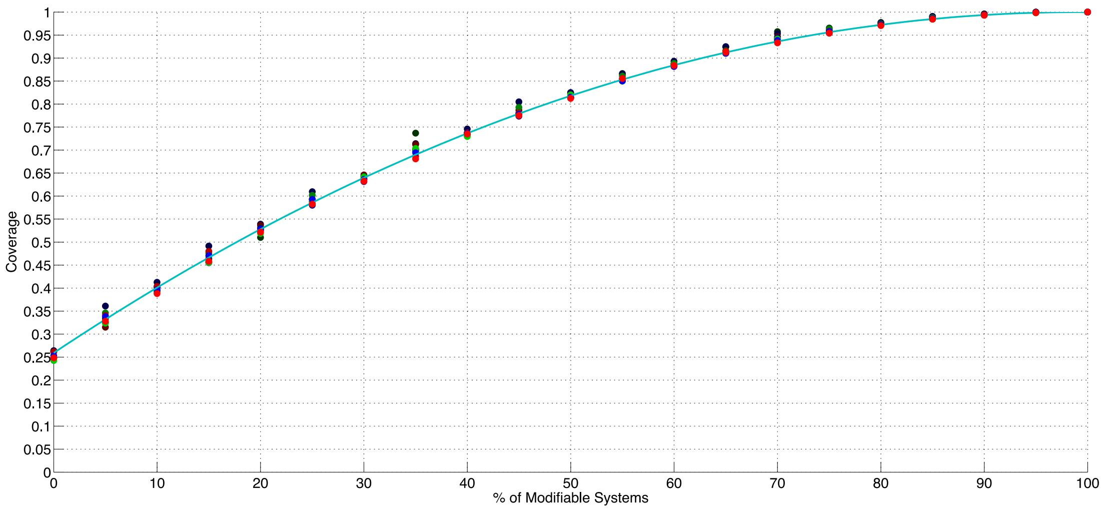


Figure 77: Modifiable Systems (%) vs. Coverage in P2P Interoperability

Therefore, regarding the subject of Coverage vs. Modifiability, and using the previous graphic as reference, the P2P Interoperability study reached the conclusion that the maximum possible coverage in an environment, where not all systems can be modifiable to improve their interoperability is related to that characteristic.

One of the design goals of H&S Interoperability was to overcome this limitation. More specifically, the decision to use a mediator is concretely tailored for this problem, which means that the modifiability of the systems is – no longer – a parameter that is used in H&S Interoperability. Even so, it must be studied if full interoperability between the systems is to be achieved.

From the simulations of P2P Interoperability (previously demonstrated), the conclusion can be drawn that coverage only depends on the number of systems and on the percentage of exporters and importers that exist in the scenario. Combining this information results in a comprehensive graphic of H&S Interoperability behaviour regarding coverage, as presented in Figure 78.

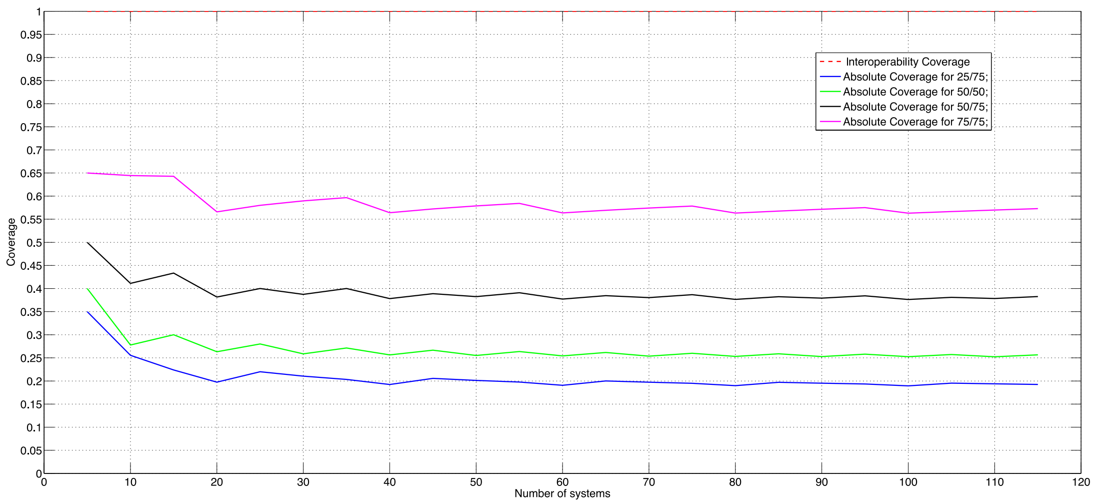


Figure 78: All H&S Interoperability Coverage behaviours on one graphic

From this graphic, it is possible to see that, regarding the simulated interoperability scenarios, the H&S Interoperability always achieves the maximum possible interoperability within the systems in an environment, thus guaranteeing full interoperability.

For the sake of comparison between the two interoperability approaches, the same comparison of the behaviour of Coverage vs. Modifiability of systems is done. The graphic of behaviour of the Modifiability studied vs. the Coverage was plotted, and is presented in Figure 79.

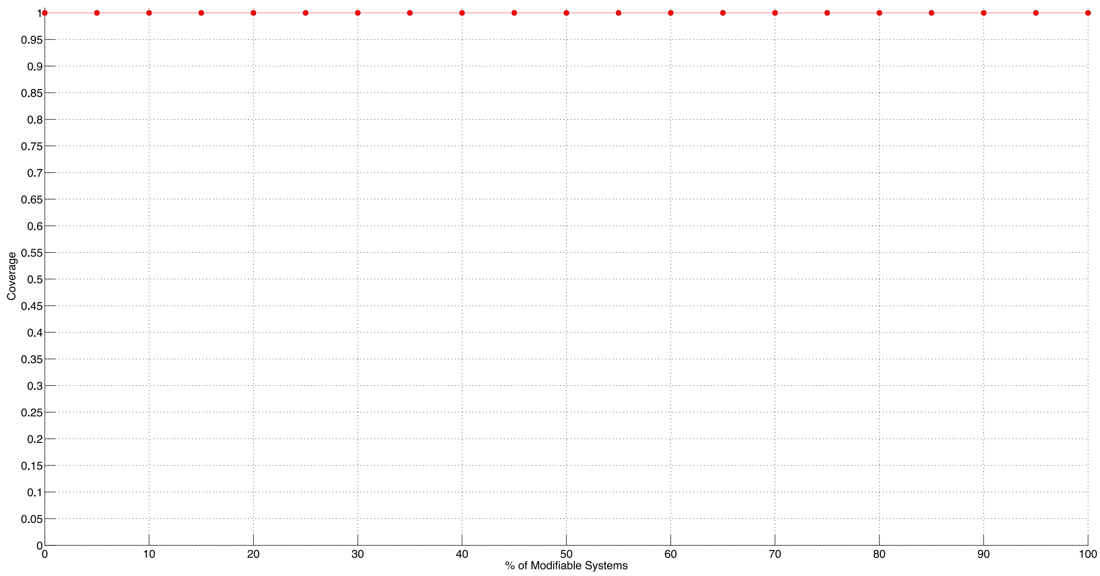


Figure 79: Modifiable Systems (%) vs. Coverage on H&S Interoperability

As expected the H&S Interoperability does not depend on modifiability (as stated before, it does not even contemplate that characteristic of a system), so a combined graphic of the behaviour of P2P and H&S Interoperability is presented in Figure 80.

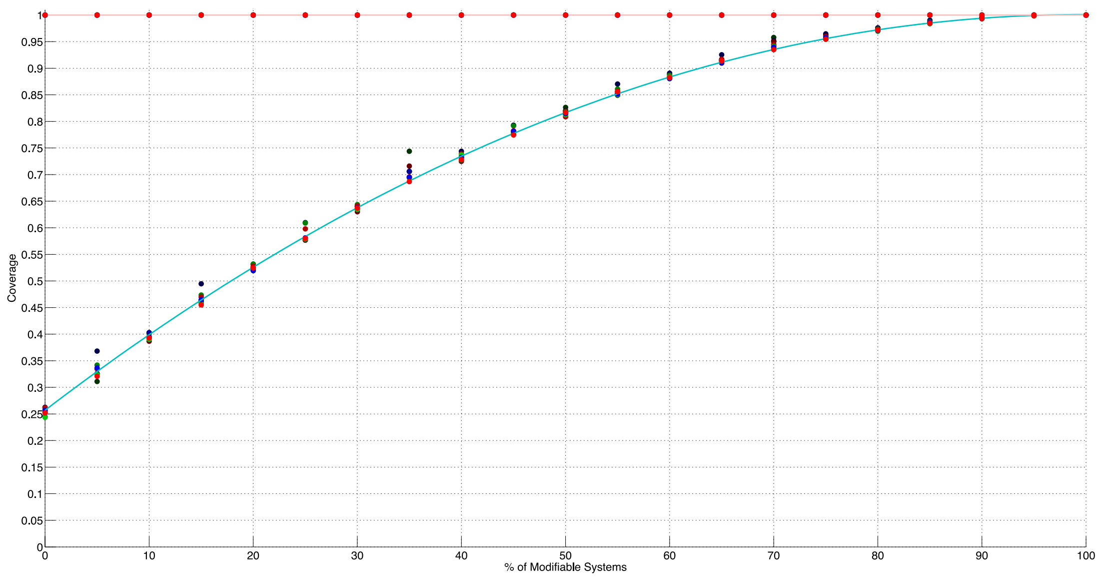


Figure 80: Comparison of Modifiable Systems (%) vs. Coverage of H&S and P2P

This concludes that the H&S solution addresses this problem with complete success, so H&S Interoperability can always achieve the maximum possible interoperability.

6.1.2 NIA vs. Needed Connections

Simulations of the H&S Interoperability are used as the starting point to understand if it can overcome this problem, and more concretely the output graphics that show a linear behaviour. For detailed analysis of the H&S Interoperability behaviour on the NIA, one starts by making an approximation to those plots (shown in the simulations). These approximations were made by using a linear equation, defined in (6.1.1):

$$\mathrm{ax} + \mathrm{b} = 0 \tag{6.1.1}$$

For the approximation made to the simulation results, the constant (b) is discarded, only approximating the linear coefficient (a). The reason behind this choice is that it is possible to relate all the simulations between them in order to understand the impact of each parameter on the results, so the approximation made is presented in (6.1.2):

$$\mathrm{ax} = 0 \tag{6.1.2}$$

Therefore, the starting point for the NIA analysis is all the coefficients, so that relations between them can be understood. For this analysis, a table of coefficients is presented, where the rows represent the Exporters/Importers relations, and the shades of blue are the variation of the percentage of equal formats in the environment:

E:25%	*I:75%*	**2,231**	*F:25%*
		1,898	*F:50%*
		1,497	*F:75%*
E:50%	*I:50%*	**2,255**	
		1,918	
		1,510	
E:50%	*I:75%*	**2,797**	
		2,376	
		1,876	
E:75%	*I:75%*	**3,360**	
		2,852	
		2,247	

Table 11: H&S Linear Coefficients of NIA function approximation

The Table 11, demonstrating the summary of the linear Coefficients, is used as the base of the study of the relations between the parameters that comprise the study. So, when exploring the evolution of these parameters in the simulations, two main courses are studied: (1) the Percentage of Equal Formats and (2) the Percentage of Exporters and Importers:

4. Percentage of Equal Formats: Each one of the graphics that were made in the simulations of the H&S approach, have the plot of three different behaviours of the different application in the same conditions on a different number of formats, so it is the first study where, regardless of the other conditions, the relations between these parameters are always the same.

5. Percentage of Exporters and Importers: The other axis of evolution (the four independent graphics of the H&S Simulations) concerns the combination of Exporters and Importers, so one must study if it is independent of the others. The Exporters and Importers were combined because together they determine the number of systems to be connected, and are thus interoperable in the interoperability scenario.

Using these two parameter variations, a study of each one is carried out, and its detailed analysis is presented next. All the studies use as baseline the previously presented table of the quadratic coefficients, and will present an individual analysis of each one.

At the end, a comprehensive analysis of the combination of all these parameters is carried out, so that the behaviour of the NIA can be understood for this type of interoperability solution, namely Hub-and-Spoke Interoperability.

6.1.2.1 Percentage of Equal Formats

As stated before, the variation of the percentage of equal formats is represented by the different shades of blue, in order to understand if, regardless of the other parameters, the linear coefficient always has the same behaviour, meaning it is an independent variable from the other parameters.

Table 12 shows the relation between the baseline (light blue, where the percentage of equal formats is 75%) and the others conditions (medium blue, percentage equals 50%

and dark blue, 25%). This relation is presented in two shades of green, the darker being the relation of 75% to 25% and the lighter the relation of 75% to 50%).

		F:25%	F:50%	F:75%	F: 75% → 25%	F: 75% → 50%
E:25%	I:75%	**2,231**	**1,898**	**1,497**	1,490	1,268
E:50%	I:50%	**2,255**	**1,918**	**1,510**	1,493	1,270
E:50%	I:75%	**2,797**	**2,376**	**1,876**	1,491	1,267
E:75%	I:75%	**3,360**	**2,852**	**2,247**	1,495	1,269

Table 12: Relations of Linear Coefficient for Equal Formats %

From the previous table, that represents the linear coefficient relations between the equal formats percentage, it can be seen that all the values calculated for the same relation have approximately the same value. So, to understand if all of them converge to the same values, a statistical analysis was performed:

Relation	*Average*	*Minimum*		*Maximum*	
		Absolute	Δ	Absolute	Δ
F: 75% → F: 50%	1,268	1,267	0,001	1,270	0,002
F: 75% → F: 25%	1,492	1,490	0,002	1,495	0,003

Table 13: Statistics of Relations of Linear Coefficient for Equal Formats %

The statistical analysis presented in Table 13 shows that all the values are almost equal, with such small variations that they can be ignored, thus considered all values are in fact the same. This represents the independency of the parameter, and the values show that, from the baseline to the worst case, the variation is 1.5 times, representing 50% more.

6.1.2.2 Percentage of Exporters and Importers

The other parameter that has influence on the number of needed NIAs, in the H&S approach, is the exporters and importers. As already shown in the simulations, what is relevant is the combination of exporters and importers, and not the order. It is irrelevant for the approach if there are two systems that wants to communicate with one, or one system that wants to communicate with two, because they result in the same number of NIAs. In Table 14 is presented the relation between the baseline (Exporters: 25% and Importers 75%) and the other cases.

E:25%	I:75%	**2,231**		*F:25%*
		1,898		*F:50%*
		1,497		*F:75%*
E:50%	I:50%	**2,255**	1,011	E:25% I:75% → E:50% I:50%
		1,918	1,011	
		1,510	1,009	
E:50%	I:75%	**2,797**	1,254	E:25% I:75% → E:50% I:75%
		2,376	1,252	
		1,876	1,253	
E:75%	I:75%	**3,360**	1,506	E:25% I:75% → E:75% I:75%
		2,852	1,503	
		2,247	1,501	

Table 14: Relations of Linear Coefficient for Exporters/Importers %

Again, like the previous analyses, the H&S algorithm proves to be very coherent, and the deviations between parameter variations are at the minimum. To understand what that minimum is, a statistical analysis is shown in Table 15.

Relation	*Average*	*Minimum*		*Maximum*	
		Absolute	Δ	Absolute	Δ
E:25% I:75% → E:50% I:50%	1,010	1,009	0,001	1,011	0,001
E:25% I:75% → E:50% I:75%	1,253	1,252	0,001	1,254	0,001
E:25% I:75% → E:75% I:75%	1,503	1,501	0,002	1,506	0,003

Table 15: Statistics of Relations of Linear Coefficient for Exporters/Importers %

The statistical analysis confirms that the deviations are an absolute minimal, which proves two things: (1) the coherence of the algorithm, resulting, on average, in a very stable value, and (2) the independency of this parameter from the other parameter (the number of equal formats). Also, the variation can be found to be again around 50% more for the worst case, in comparison to the baseline.

6.1.2.3 H&S NIA Behaviour Conclusion

<u>The NIA behaviour of the H&S Interoperability approach is a linear behaviour</u>. In Figure 81, the variation of the NIAs within all the simulation parameters is shown, including (1) the linear behaviour, and (2) the NIA variation, which in the worse case gets close to 400 NIAs.

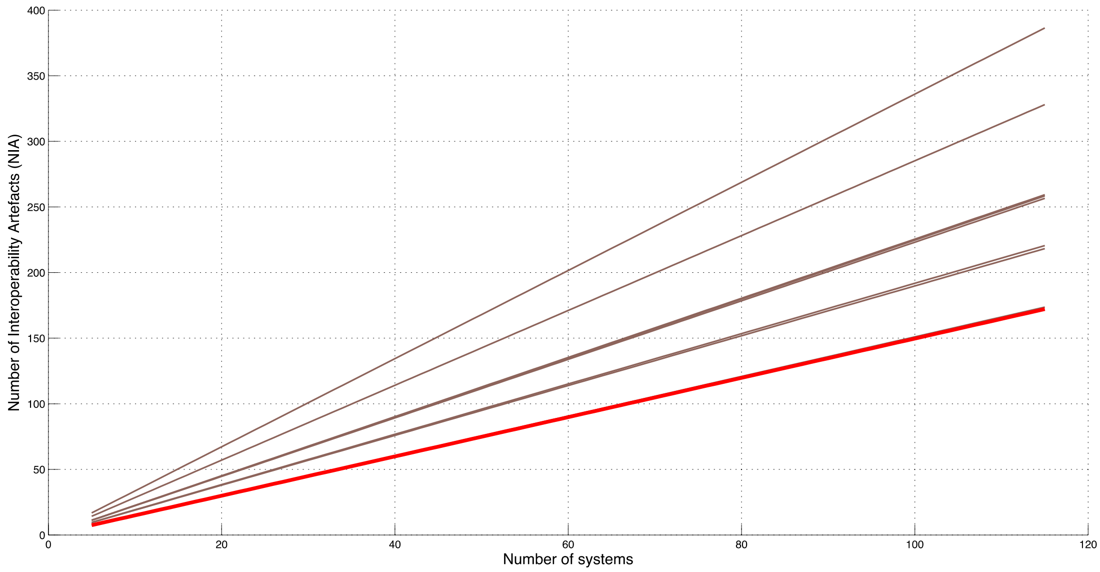


Figure 81: H&S Interoperability NIA-on-a-graphic

To understand the variation between the best case (the baseline for the analysis, shown in the graphic with the colour red) and the worst case (the top line), one can look at the summary table of the parameter variations, shown in Table 16.

Relation	*Average*	*Minimum*		*Maximum*	
		Absolute	Δ	Absolute	Δ
F: 75% → F: 50%	1,268	1,267	0,001	1,270	0,002
F: 75% → F: 25%	1,492	1,490	0,002	1,495	0,003
E:25% I:75% → E:50% I:50%	1,010	1,009	0,001	1,011	0,001
E:25% I:75% → E:50% I:75%	1,253	1,252	0,001	1,254	0,001
E:25% I:75% → E:75% I:75%	1,503	1,501	0,002	1,506	0,003

Table 16: Statistics of Relations of Linear Coefficient

From the table, two main things stand out, (1) regardless of the parameter under analysis there is never more than 1.5 times the increase of the linear coefficient over the baseline, which means that, (2) even with the two parameters combined, the worst simulation only results in 2.25 times more NIA than the baseline.

6.1.2.4 H&S vs. P2P Interoperability on NIA

All of the analyses of the H&S and the P2P Interoperability approaches related to the NIA were based on the approximations of the simulation curves. Therefore, to start this discussion, all the results can be combined in one graphic, as shown Figure 82.

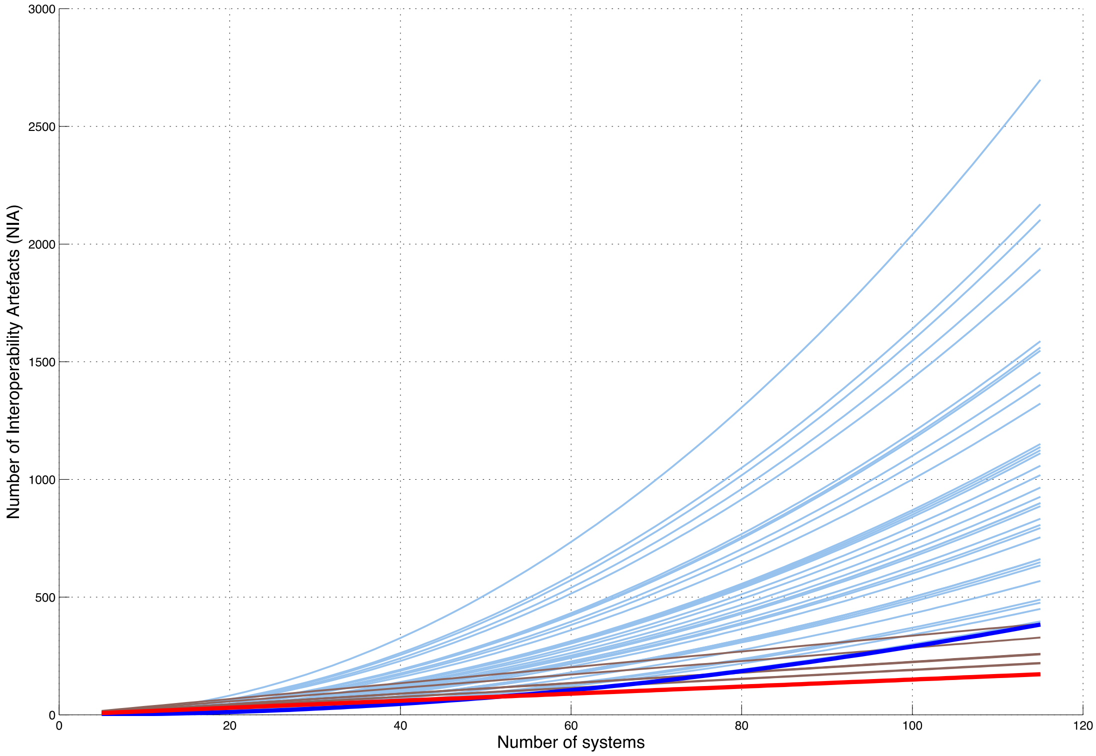


Figure 82: H&S vs. P2P NIA-approximations-on-a-graphic

The first conclusion, and the one that is totally obviously at a glimpse the graphic, is the range of NIA that one method and the other achieves. H&S Interoperability has a maximum close to 400 NIAs, whereas P2P Interoperability goes over 2500 NIAs. Which results on a six times higher NIA on P2P over H&S.

The second conclusion is related to the baselines. It is interesting to see that the P2P approach outperforms the H&S approach when the number of systems is small, but, as designed it performs better on the large-scale, i.e. when the number of systems is bigger. This behaviour will be described in more detail later on, in this chapter.

The third conclusion, as already stressed, is the behaviour of one and the other: P2P has a quadratic behaviour, whereas H&S acts linearly. In Table 17, the coefficient relation between the independent parameter variations is depicted.

Relation	*P2P*	*H&S*
F: 75% → F: 50%	1,700	1,268
F: 75% → F: 25%	2,317	1,492
M: 25% → M: 50%	1,284	–
M: 25% → M: 75%	1,044	–
E:25% I:75% → E:50% I:50%	1,180	1,010
E:25% I:75% → E:50% I:75%	1,672	1,253
E:25% I:75% → E:75% I:75%	2,378	1,503

Table 17: Parameter evolution: P2P vs. H&S

Some analyses can be made based on the table. First is the coefficient variation from the best (baseline) to the worst case: whereas in P2P, (where the worst case is represented by the Formats on 25%, the Modifiable on 50% and the combination of Exporters and Importers of 75% each) results are approximate 7 times higher in coefficient than the baseline, as regards H&S the variation is only of 2.25 times.

Second, and considering the parameters independently, the biggest variation that a single parameter has, in the case of P2P, is of 2.378 over the baseline, whereas the H&S approach has a much more friendly number of 1.503.

Third, and probably most importantly, because it places the last two in a bigger context, in the P2P analyses the relation is over a quadratic coefficient, whereas in H&S analyses it over a linear one, thus raising the H&S performance even more.

6.2 Small-Scale Performance

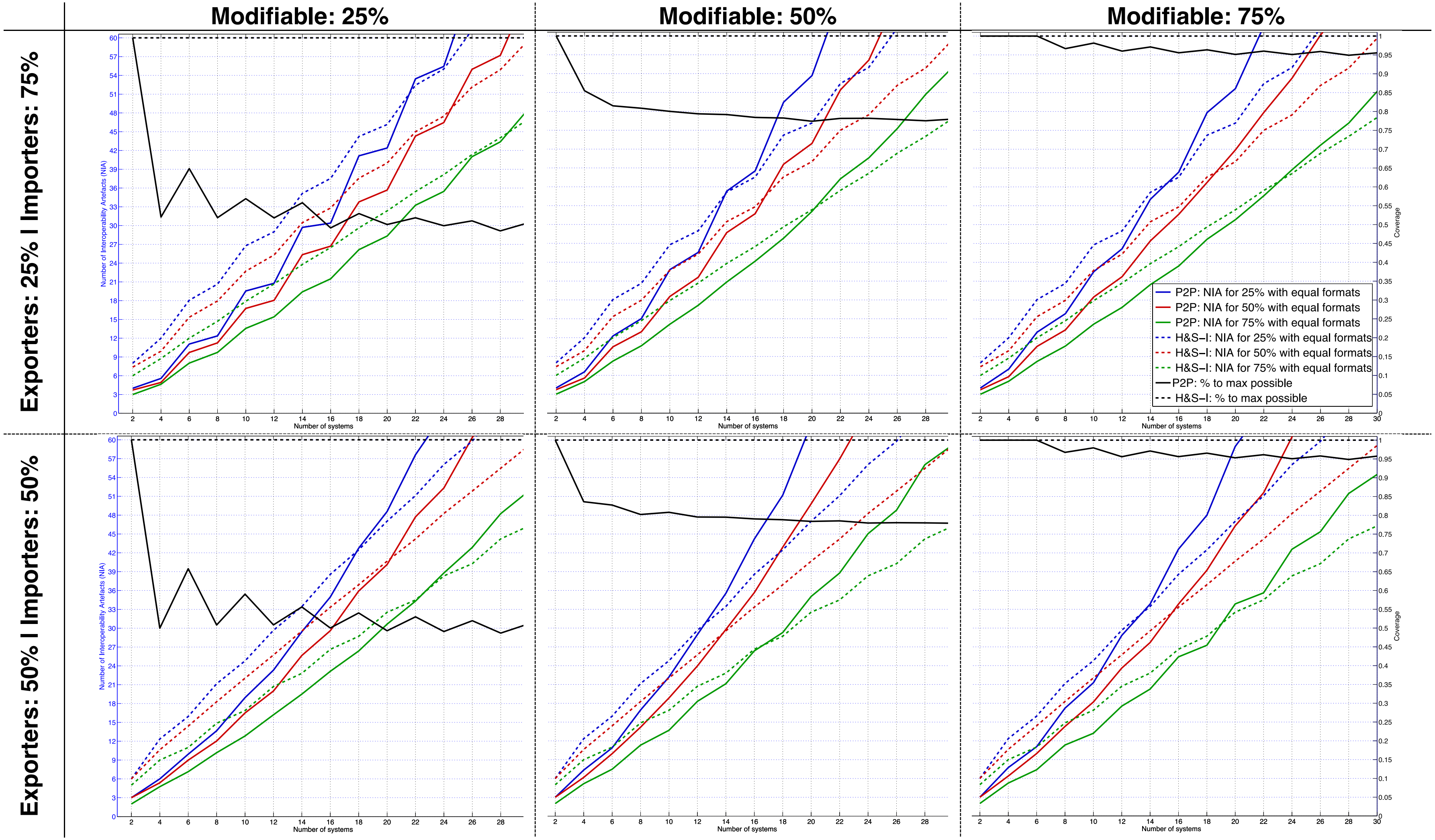
The Hub-and-Spoke interoperability approach was defined to overcome the large-scale problem, but since it is a full interoperability approach, one should check if it could perform well in the conditions of a small number of systems, thus analysing its small-scale performance.

Another problem that the Hub-and-Spoke was designed for, was to overcome the coverage limitation, and as already stated in this evaluation, the Hub-and-Spoke, contrary to the P2P Interoperability approach always achieve full interoperability in an environment, so, this will be a factor on this small-scale analysis.

But, and, the reason why it is important to conduct this small-scale performance study is to assess the impact of using an approach such as the Hub-and-Spoke with a small number of systems. In such interoperability approaches, where there is the need to use a mediator and to develop the needed artefacts to interoperate even systems that are already interoperable, the initial “cost” can potentially be high.

So, this study is to understand if and where the Hub-and-Spoke interoperability performs better in the small-case, and what is the cost, in comparison with the P2P to gain the full interoperability that it always achieves.

To conduct this study, the process consists in grabbing the already made comparison graphics of the two approaches, and to zoom them to the point where the Hub-and-Spoke approach presents less NIAs than the P2P interoperability, which results in the following graphics.


Modifiable: 25%
Modifiable: 50%
Modifiable: 75%
Exporters: 25% I Importers: 75%
Exporters: 50% I Importers: 50%
Number of Interoperability Artefacts (NIA)
Number of systems
Coverage
P2P: NIA for 25% with equal formats
P2P: NIA for 50% with equal formats
P2P: NIA for 75% with equal formats
H&S–I: NIA for 25% with equal formats
H&S–I: NIA for 50% with equal formats
H&S–I: NIA for 75% with equal formats
P2P: % to max possible
H&S–I: % to max possible

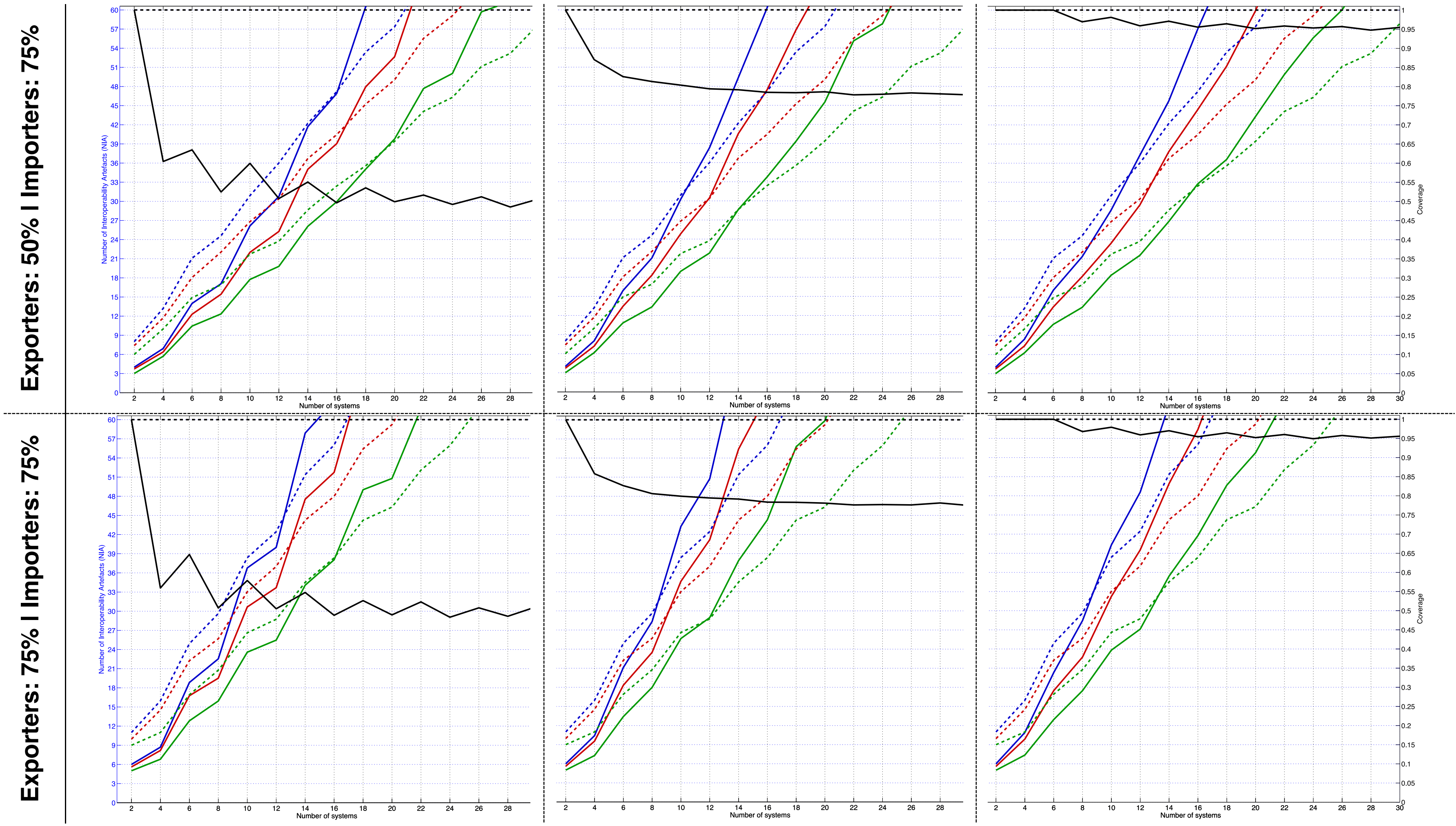


Figure 83: P2P vs. H&S on Small Scale

First, the coverage presents the expected behaviour since, where in the P2P it depends on the modifiability of the systems, in the Hub-and-Spoke it is always the maximum possible between the systems in the interoperability scenario.

However, the big question is what is the cost to have this improvement in the coverage. From the graphics, one can see, that under the point where P2P interoperability presents a better performance than Hub-and-Spoke, the gap between the two is pretty consistent independently of the scenario parameters used. This gap, corresponding to an excess in the Number of Interoperability Artefacts of the H&S over the P2P, consists in average to four Interoperability Artefacts.

Another interesting, but predictable, conclusion is that the increase of the number of needed connections (i.e. the increase of the importers and exporters) results in a reduction of the number of systems where the H&S break-evens the P2P. This is expectable because the H&S was designed to have good performance with a large number of connections. The reduction of the number of systems is relevant, and in the three cases (of the three studied modifiability) the reduction was of 10 systems, which in percentage, represents a reduction of about 50% of the break-even point on H&S vs. P2P. This behaviour states that the H&S approach is very suitable for interoperability scenarios composed by a small number of systems where exists a large number of needed connections.

The modifiability characteristic of the systems, only has impact (in terms of the break-even point) where the modifiability is low, being this point located on higher number of systems not because the P2P performs better but because of its limitation on achieving a good coverage, so, as already stated, despite needing more NIAs, H&S outperforms P2P in terms of coverage, which is also an expectable result, because it was designed to overcome the modifiability limitation.

6.3 Overview

The overview will explore the simulations of each one of the methods, and provide a combined view of each one, so the reader can see a one-on-one comparison between both, and they will be grouped by:

1. Coverage and NIA with all parameters: All of the simulations were made, by using interoperability scenarios based on four parameters: Modifiability, Number of Equal Formats, Exporters and Importers. This will explore the variation of the combinations of those parameters.

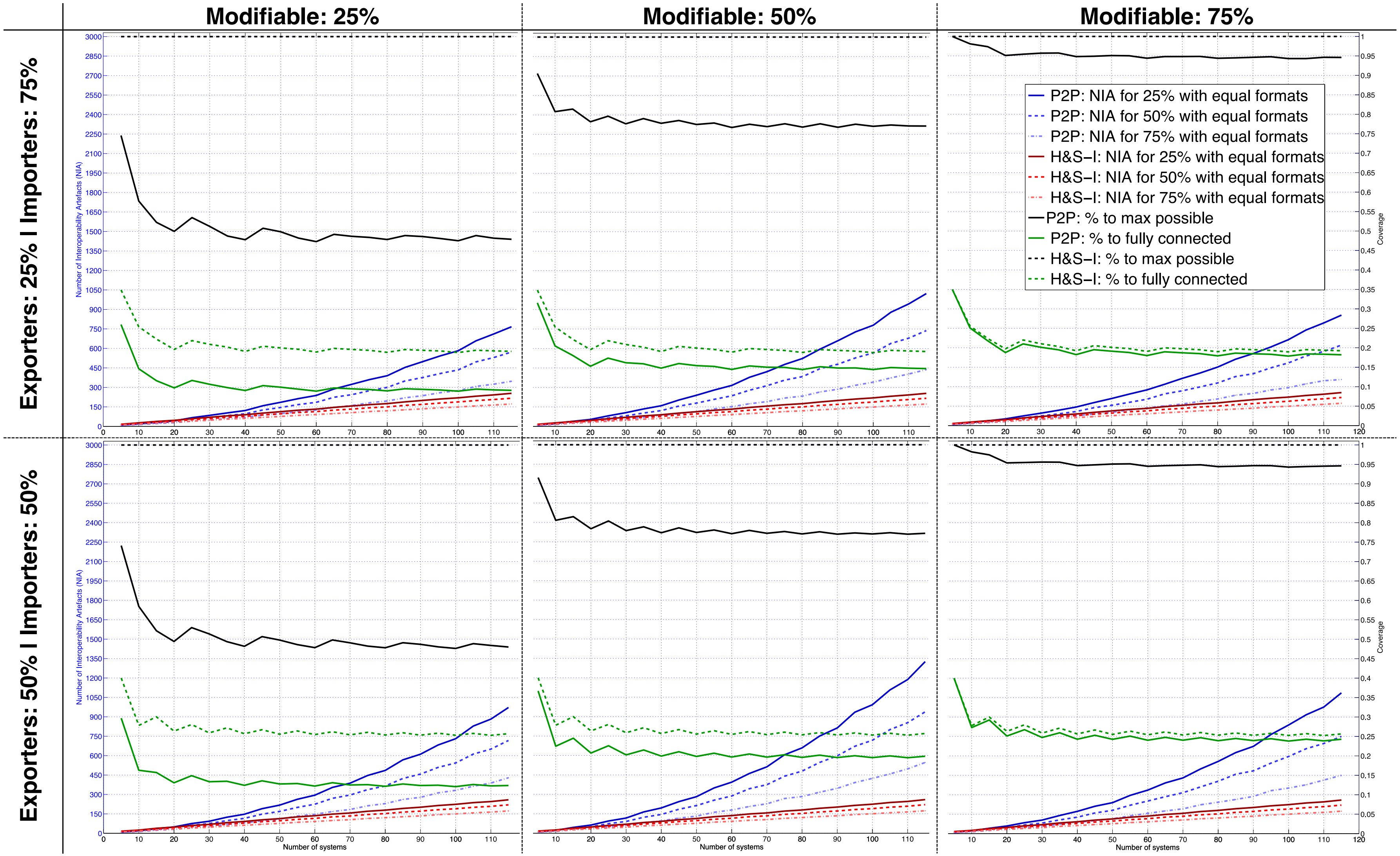
2. Percentage to Fully Connected group by Modifiable Systems: There are some parameters that don't depend on others, such as the coverage on the modifiability and the number of equal formats, so this will show the evolution of the coverage grouped by the modifiability of the systems.

3. Percentage to Fully Connected group by Exporters and Importers: As the previous case, this is also grouped, but by the combination of Exporters and Importers.

4. Percentage to Maximum Possible: The maximum possible in the P2P depends on the modifiability of the systems, while the H&S always get the maximum, so this will show the impact of this.

6.3.1 Coverage and NIA group by all parameters

The next two pages show the comparison of the two algorithms, where the graphics vary by modifiability on the columns, the combination of Exporters and Importers on the rows, and, in each graphic, the variation of equal formats.


Modifiable: 25%
Modifiable: 50%
Modifiable: 75%
Exporters: 25% I Importers: 75%
Exporters: 50% I Importers: 50%
Number of Interoperability Artefacts (NIA)
Coverage
Number of systems
P2P: NIA for 25% with equal formats
P2P: NIA for 50% with equal formats
P2P: NIA for 75% with equal formats
H&S–I: NIA for 25% with equal formats
H&S–I: NIA for 50% with equal formats
H&S–I: NIA for 75% with equal formats
P2P: % to max possible
P2P: % to fully connected
H&S–I: % to max possible
H&S–I: % to fully connected

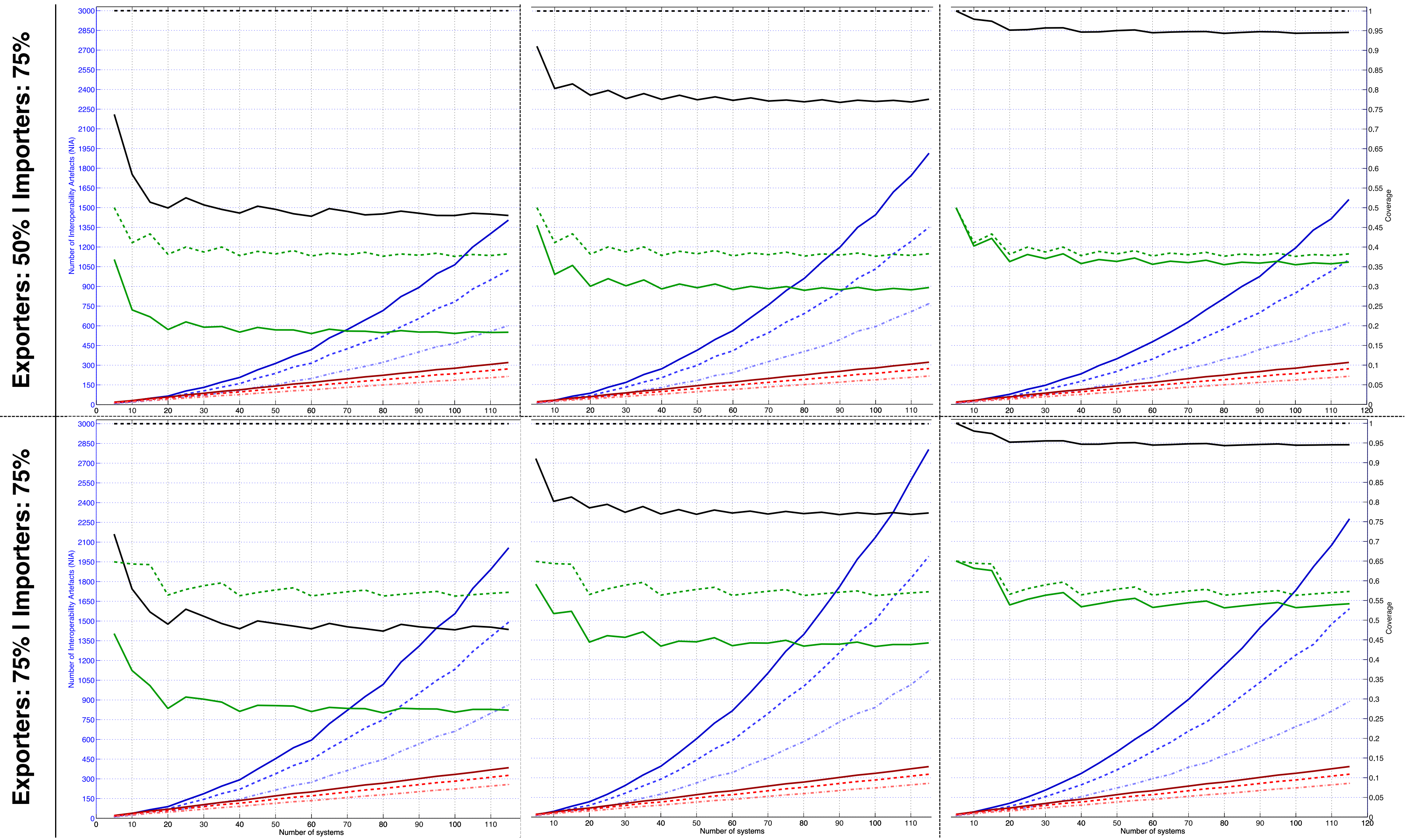


Figure 84: Coverage and NIA of P2P vs. H&S Interoperability with all parameters

6.3.2 Percentage to Fully Connected group by Modifiable Systems

Here, the coverage analysis with the variation of the Modifiable systems is presented. It is important to note that the H&S approach does not vary with the parameter. So, it shows the same behaviour in all the graphics. Next, the graphic for 25% of Modifiable Systems is presented, where it is possible to see that the P2P is limited by the Modifiability of the systems, and is very far away from the H&S approach.

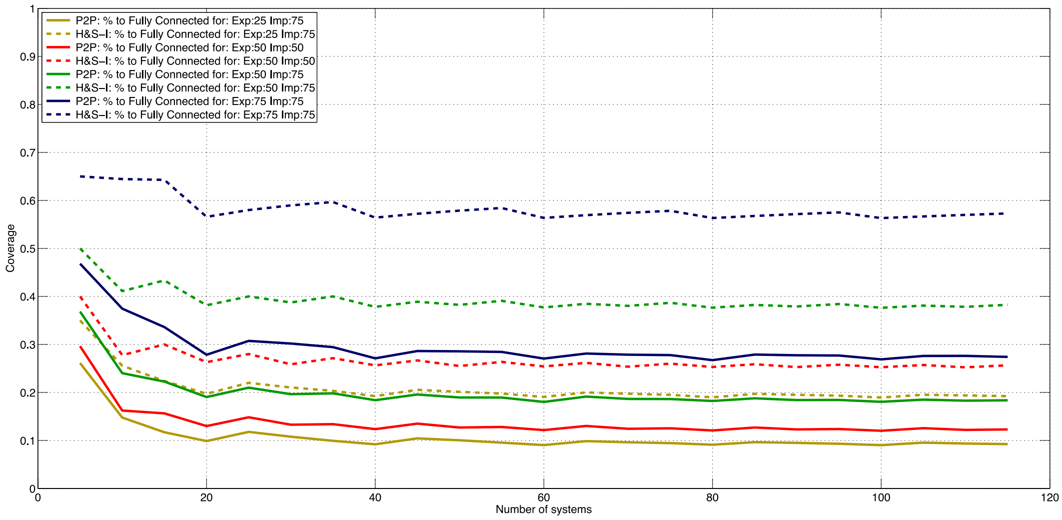


Figure 85: Percentage to Fully Connected of H&S vs. P2P group by 25% Modifiable

The following graphic presents the 50% of Modifiability systems in a scenario, where one can see that with the increase in Modifiability (25% → 50%), coverage also increases, but still far from the H&S Interoperability approach.

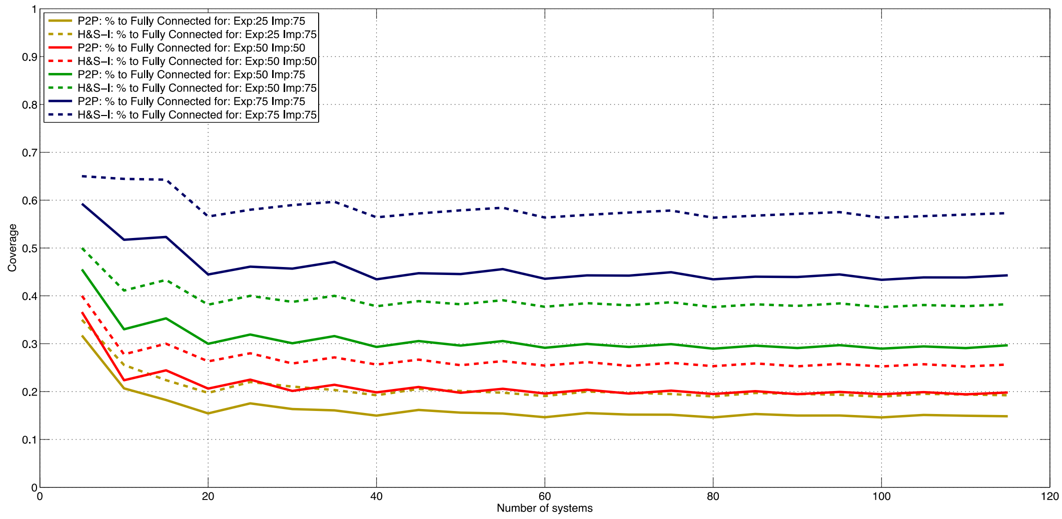


Figure 86: Percentage to Fully Connected of H&S vs. P2P group by 50% Modifiable

Next, the simulation for the 75% of modifiable systems is presented, where the dependability of P2P Interoperability on this parameter is clear, because it starts to be very close in performance to that of the H&S approach.

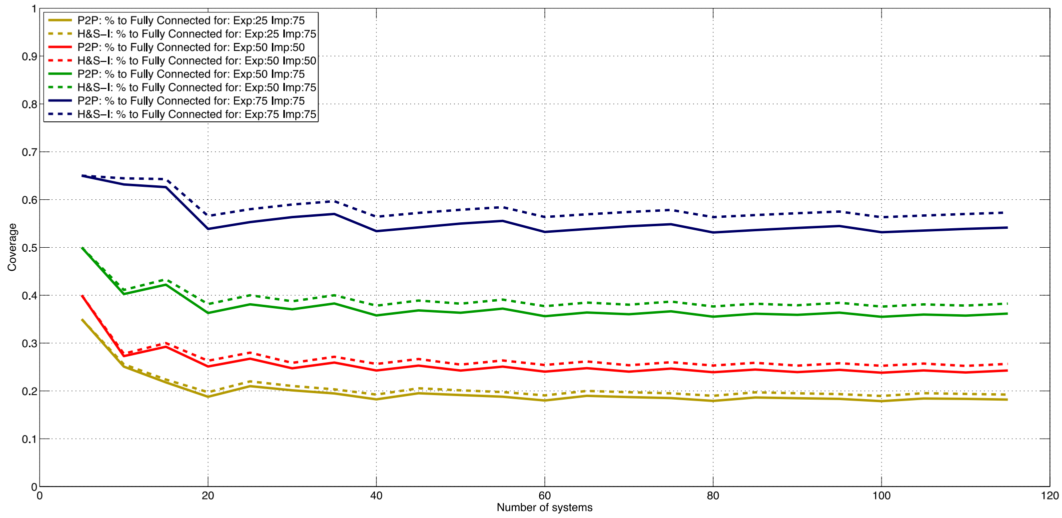


Figure 87: Percentage to Fully Connected of H&S vs. P2P group by 75% Modifiable

6.3.3 Percentage to Fully Connected group by Exporters/Importers

The next four graphics depict the variation of coverage with the combination of Exporters and Importers. Each shows the variation of the modifiability parameter. It must be noted that the number of equal formats is not plotted; this is because it has no impact on coverage, but only on the NIA needed to achieve that coverage. The next graphic plots the combination of Exporters 25% and Importers 75%, where P2P is seen to maintain a relatively close result to the H&S.

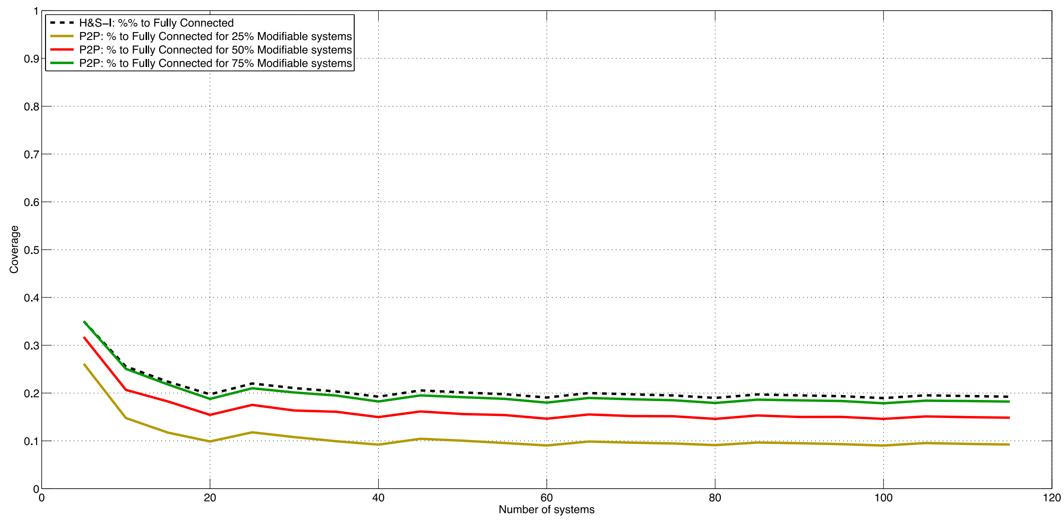


Figure 88: Percentage to Fully Connected of H&S vs. P2P group by E:25% and I:75%

The next graphic plots the behaviour of P2P vs. H&S Interoperability for the case of the Exporters and Importers with the same value, 50%. They are very similar to the previous one due to the fact that there is no system that exports and imports at the same time, and so the graphic depicts the already studied behaviour of the modifiability of systems, where they impact on coverage.

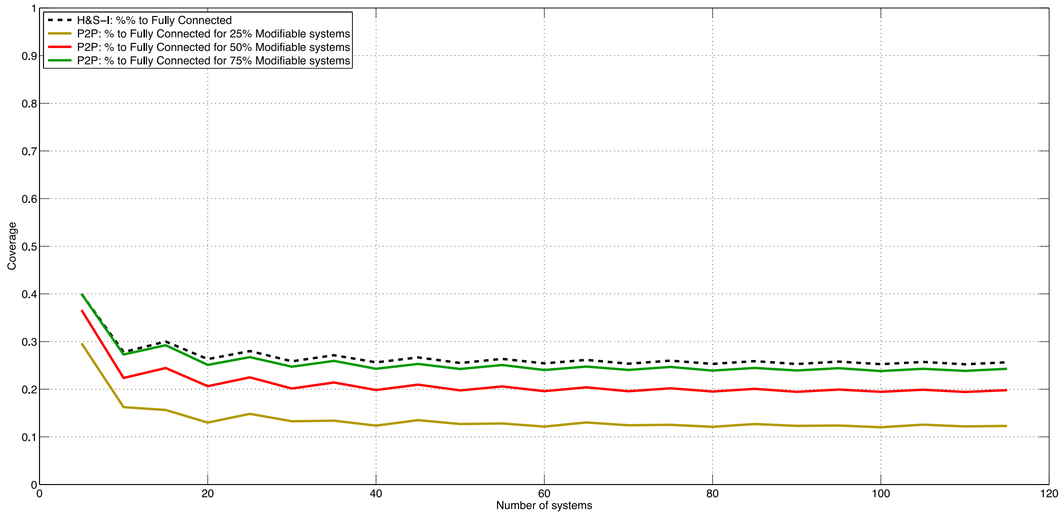


Figure 89: Percentage to Fully Connected of H&S vs. P2P group by E:50% and I:50%

The case of Exporters 50% and Importers 75% is presented in the next graphic. It shows that, despite the behaviour being close to the previous ones, the gap between the P2P and the H&S starts to increase, because the H&S has a better performance when the number of connections increases.

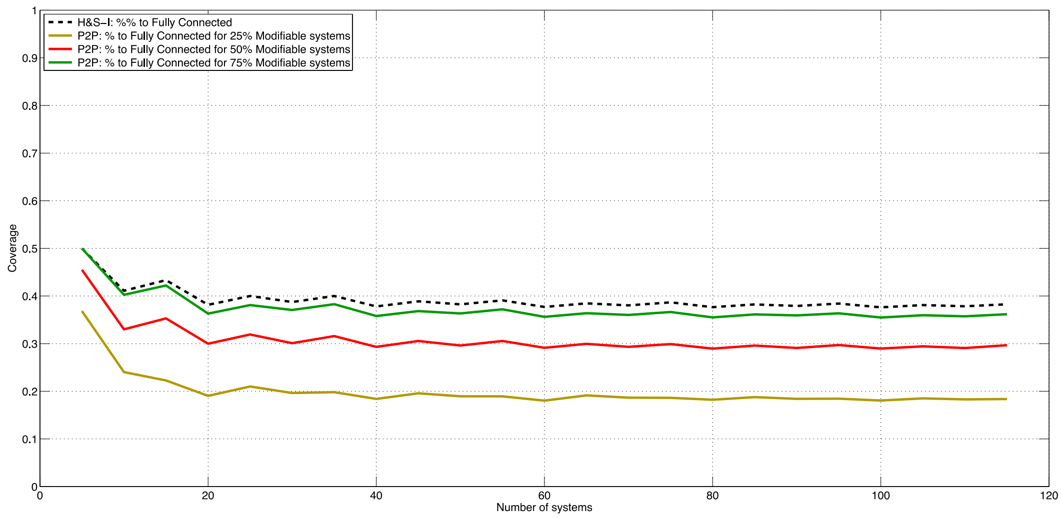


Figure 90: Percentage to Fully Connected of H&S vs. P2P group by E:50% and I:75%

The last graphic, presented in the next figure, shows the case of Exporters 75% and Importers 75%, showing that the gap between the two approaches under study is even wider than in the previous cases.

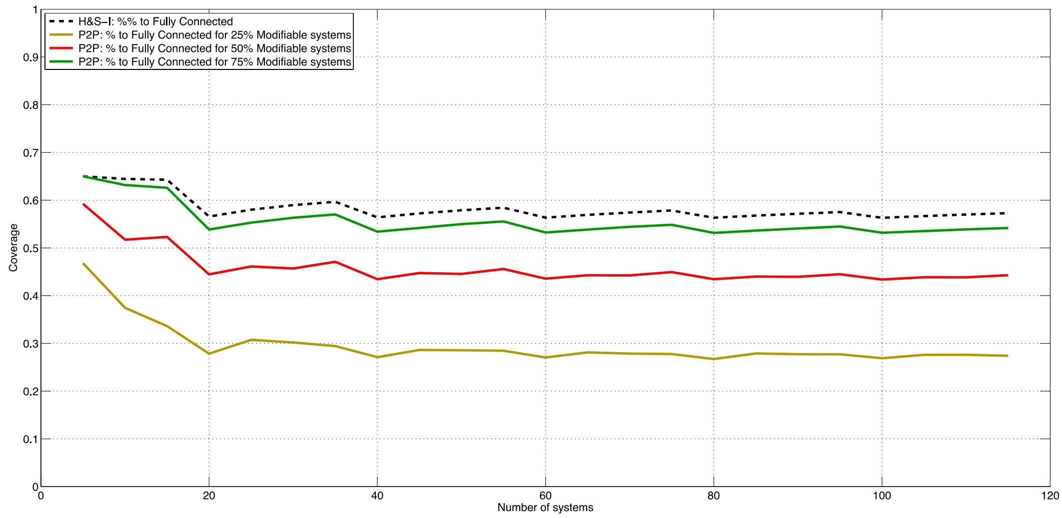


Figure 91: Percentage to Fully Connected of H&S vs. P2P group by E:75% and I:75%

6.3.4 Percentage to Maximum Possible

The maximum possible coverage is a relation between the coverage and the maximum possible coverage in a specific scenario, where 100% of coverage is the case when all systems, that have the capacity to interoperate with each other, do in fact interoperate. The next graphic shows, once more, the dependability of P2P on modifiability, and the outstanding result of H&S, always achieving full interoperability.

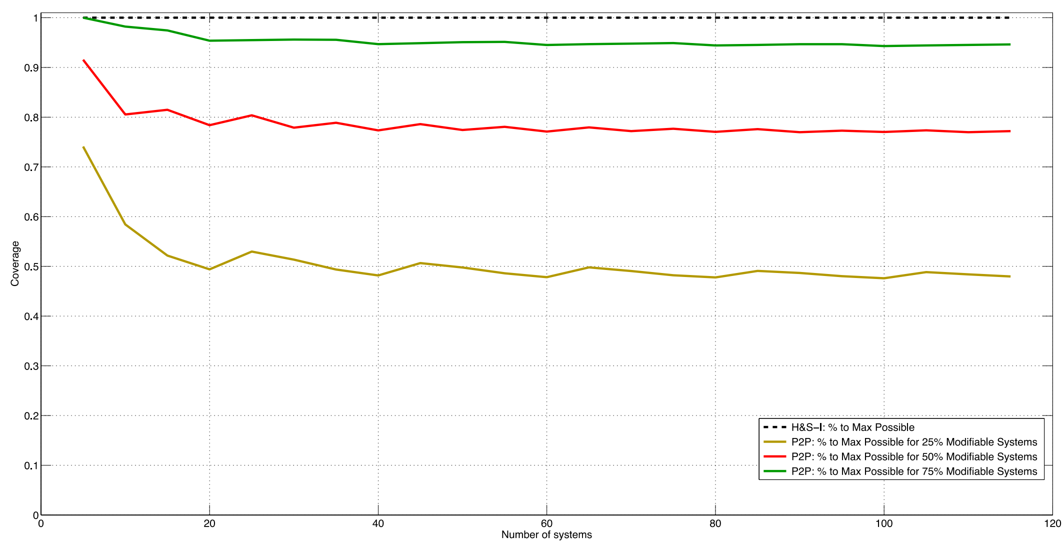


Figure 92: Percentage to Maximum Possible of H&S vs. P2P group by Modifiable

6.4 Verdict

This evaluation process confirms that the Hub-and-Spoke Interoperability approach– as designed here – tackles the problems identified in the large-scale interoperability scenarios, reducing the NIA and increasing the Coverage.

The NIA Problem, which concerned the large number needed for P2P Interoperability, was overcome, by achieving, at the end, a reduction of over six times of the NIA on the large-scale Interoperability scenarios. In Figure 93, the variation of the NIA in P2P (in blue) and the H&S (in brown) can be seen.

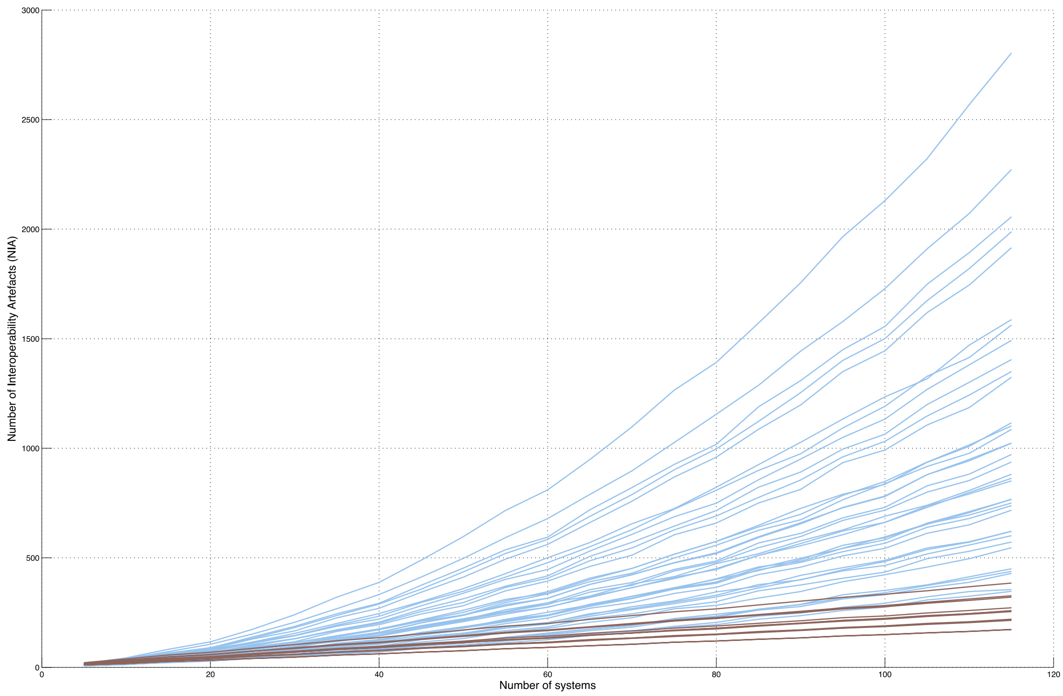


Figure 93: H&S vs. P2P NIA-on-a-graphic

The graphic also shows the type of behaviour of the two approaches, where the quadratic behaviour of P2P - which leads to a bigger problem on an even larger scale (with the increase of the NIA) - is transformed into a linear behaviour in the H&S approach, resulting in much less dependability on the number of systems presented in an interoperability scenario.

This large reduction of the NIA does not come alone. It is accompanied by an increase in the coverage of an interoperability scenario. But this increase, is not a quantified increase, it is a total increase, because the H&S Interoperability approach always achieves the maximum possible/needed interoperability desired.

Figure 94 shows how every simulation made in the H&S Interoperability approach (represented in brown) always give 100%, whereas in the case of the P2P Interoperability approach (represented in blue) a large range of results is achieved, showing its vast dependability on the systems' needs of the scenario.

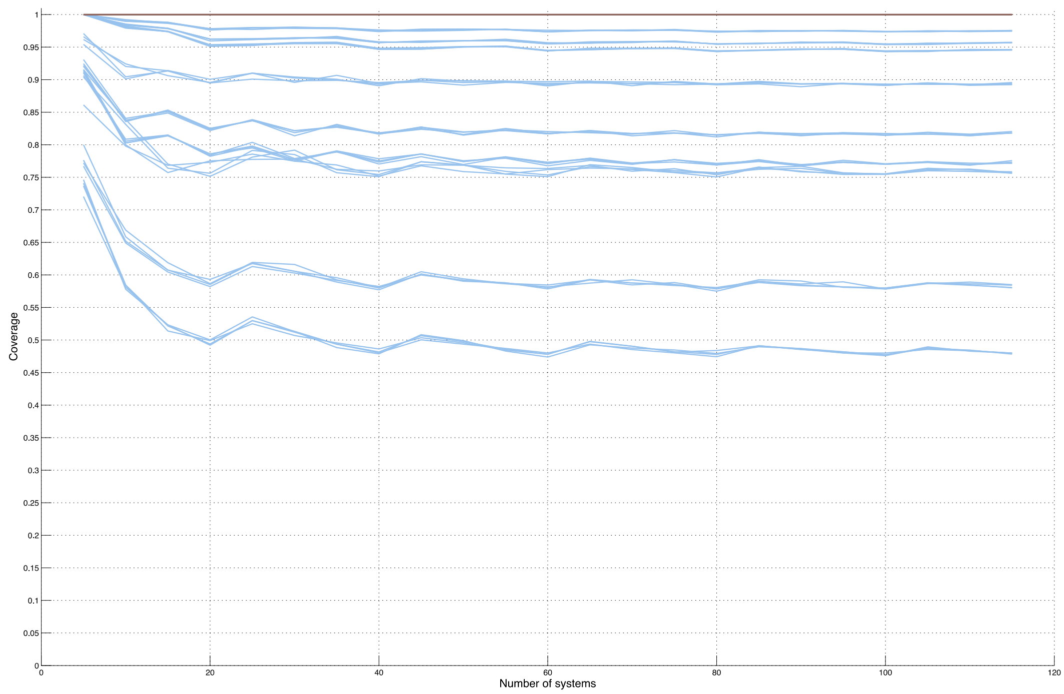


Figure 94: Percentage to Maximum Possible of H&S vs. P2P

However, this good result implies an associated potential problem:

- Small-scale performance: this solution is not always a good solution to apply, especially with respect to small-scale systems, where the P2P approach can achieve interoperability with a small number of NIA. But, this small number also presents a problem, which is the fact that, if there are a relevant number of non-modifiable systems in the environment, total coverage cannot be achieved, whereas the H&S Interoperability approach always guarantees that.

7

Application: Plug'n'Play Interoperability

"The science of today is the technology of tomorrow"

– Edward Teller, the father of the hydrogen bomb

This chapter describes the Plug'n'Play Interoperability (Plug'n'Interoperate, PnI) solution, and how it deals particularly with: (1) the existence of many systems (devices, sensors, etc.) that need to communicate and use disparate data formats, and also; (2) new systems (even not thought at start) that are entering the environment all the time and bring new data formats not present/foreseen in the environment. In PnI, systems "plug" – into a data interoperability support system – and seamlessly exchange data (interoperate) with others. The PnI is realised via the combined use of interoperability specifications, interoperability mediation and interoperability-enabling methods.

The solution makes it possible for systems to be interoperated without the need of being modified/remanufactured because developers/manufacturers now only need to provide interoperability specifications of their own systems and it is the role of an interoperability mediator to enable interoperability. In this way, interoperability is assured at a middleware level, with full technological independence, via methods that promote interoperability where it is needed – from low-level embedded systems to high-level computer systems. And, Plug'n'Interoperate even goes to a next level of performance by supporting systems' interoperability based the organisation and execution of compositions of interoperability specifications.

7.1 Overview

The Plug'n'Interoperate (PnI) solution is essentially a dynamic-interoperability enabling platform. It assists systems on becoming interoperable without the need of changes in the application side, by acting as a mediator to interoperability. The PnI uses Interoperability Specifications – ISs, electronic (interoperability) artefacts that describe all actions needed to be executed by the platform in order to transfer the information between two data formats – to interoperate the systems' heterogeneous data formats. PnI also requires data adapters – at the mediator side – to ensure that is possible to import and export from/to each data format present in the interoperability scenario.

The PnI provides a decision support system which reasons over the environment aiming to maximize the number of interoperable data formats using the minimum number of Interoperability Specifications. Now, and to work properly on providing these interoperability-enabling services, the PnI platform needs a place to store all known interoperability-related information. This information is required to be always available so the PnI may use which Interoperability Artefacts may be useful in each environment. And, the PnI platform also needs an Execution Engine to execute ISs thus transforming the information defined in a source data format into a target data representation.

The set of services provided by PnI, and depicted in Figure 95, are: (1) CRUD Services: which enables users/applications to use the four basic functions of persistent storage to access the platform Interoperability artefacts; (2) Interoperability-Design Support Service: which reasons over the environment Interoperability capabilities with the objective of maximising overall Interoperability; (3) Interoperability Execution Service: enables the execution of an interoperation, by choosing the adequate Interoperability Specification to interoperate the given applications, and execute it, transforming the information from one data format to the destination data format.

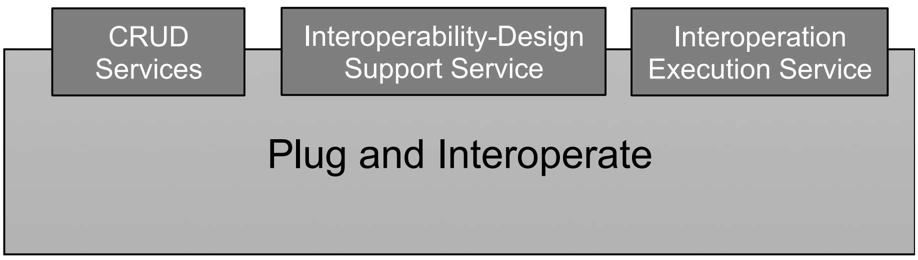


Figure 95: Services provided by the Plug'n'Interoperate Platform

These services expose the key features of the Plug'n'Interoperate technological solution. In detail, such services provide the following:

- Interoperability Artefacts "CRUD" Services: The IA CRUD allows one to Create, Read, Update or Delete Interoperability Artefacts to/from the PnI Solution. This enables applications to enquire for Interoperability Artefacts that are known by the PnI solution. It also enables developers and contributors to add new (or update previously existing) Interoperability Artefacts, or even to delete artefacts that are no longer deemed necessary. And so, the CRUD services enable the community to increase the Interoperability-related information available, and in this way enhancing the interoperability possibilities between systems in an interoperability scenario.

- Interoperability-Design Support Service: This service provides users – especially interoperability designers/engineers/managers – with the chance of enhancing the overall interoperability in a given environment conditions (can be a scenario with several applications with highly heterogeneous data formats, or a scenario with two applications with each ones data format). This exposed functionality analyses applications, Interoperability Artefacts (i.e. data formats adapters and Interoperability Specifications) and with a decision support system, provides a recommendation on how to fully interoperate the given environment. This functionality aims to maximise the number of interoperable data formats using the minimum number of interoperability specifications.

- Interoperation Execution Service: This service "executes Interoperability" by execution the Interoperability Specifications needed to interoperate any given pair of systems. It starts by analysing the Exporter adapters from the source application and the Importer adapters from the target application, and based on them will search within the PnI solution for Interoperability Specifications matching the criteria. Afterwards, the specification (or specifications, if a set of Interoperability Specifications are needed to interoperate the given systems) will be executed, enabling the two given systems to interoperate.

7.2 Definition

The Plug'n'Interoperability (PnI) is about adopting a Plug’n’Play mechanism to Interoperability. The solution is inspired in the alike Plug’n’Play approach in computer systems, where a device (a printer, a pen drive, a peripheral) brings with itself (or can be retrieved from a repository in the web) the driver that the target system can take to properly use the devices’ services. The PnI exploits this same basic principle of self-configuration as to automate, as much as possible, the configuration and participation of systems into the Interoperability environment. In the Plug’n’Interoperate environment, systems simply plug – into the interoperability support system – and promptly interoperate with other systems present in the data-sharing environment.

The Plug’n’Interoperate provides interoperability enabling methods, in order to ensure interoperability between systems that need to communicate but follow disparate data formats, and even with newly created systems (that were not thought at start) that possess new data formats not foreseen in the environment. Plug’n’Interoperate allows systems to plug into the PnI solution and seamlessly interoperate with other systems. The goal of the PnI Solution is to enable interoperability in environments where the system Data Formats prevents them from interoperating with each other.

And so, PnI uses a mediated approach where interoperations are not executed directly by the systems but by other entity, the mediator. The mediator is an entity, present in the environment that executes the interoperation between two data formats as it is requested. When using a mediated approach, only the information and the data formats used by the adapters (exporters and importers) are relevant, becoming irrelevant which systems are involved in the information exchange. Therefore a mediated approach is data format centred, instead of system centred which results in a potential reduction of the Number of Interoperability Artefacts (NIA) required to achieve maximum interoperability coverage, since the NIA required is directly related with the amount of different data formats rather than on the number of different systems.

By following the mediation approach, the scalability is addressed since in any environment the heterogeneity of devices is greater than the heterogeneity of data formats. It also doesn't require any modification of systems in the environment, as the mediator has the role of execute the interoperations, which are defined as Interoperability Specifications.

The solution is made possible by the existence of 'interoperability drivers', which define translations in between data formats – Interoperability Specifications (ISs). An Interoperability Specification is an Interoperability Artefact without the capability to execute interoperations by itself. An IS specifies all the actions that must be executed to perform the information transfer between two data formats: (1) the source system, which corresponds to the information source; and (2) the target system referring to the one that wants to consume the information. This concept is represented in Figure 96.

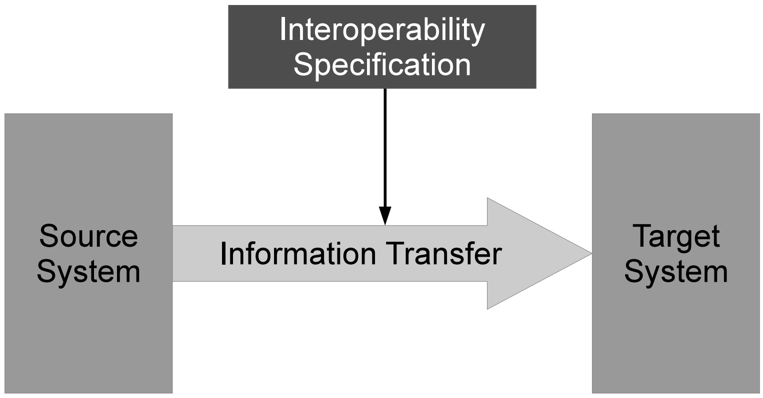


Figure 96: Interoperability Specifications

Prior to the definition of an IS that can enable the information transfer between two systems, the identification of the information representations that each system use to communicate is required. By information representations are considered data formats since all data imported or exported by information systems can be represented through data formats. Therefore, to define an IS it is required a data format used by the source system to export data, the source data format, and a data format used by the target system to import data, the target data format[24].

An Interoperability Specification describes how to perform the interoperation between a source data format and a target data format. The realization of that description is only possible if the concepts represented in each data format are well understood and a matching between the concepts of these data formats can be performed. And for that, the Plug'n'Interoperate platform needs adapters, so it can import the source data

[24] A data format can be seen as a composition of data elements, where a data element consists in an atomic unit of data with a well-defined meaning. Data elements can be organised in concepts, representing its properties and/or characteristics.

format instance, and after executing the Interoperability Specification, needs another adapter to export the information in the target data format, depicted in Figure 97.

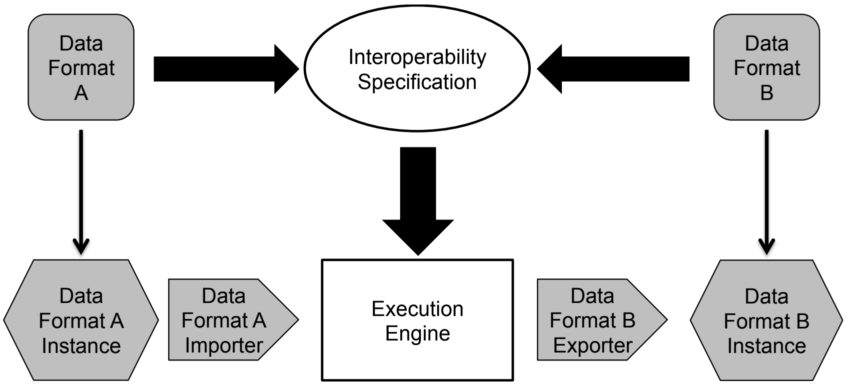


Figure 97: Interoperability Execution

Another key feature is what makes possible interoperability between data formats that don't have an explicit interoperability specification between them to be interoperable. It consists on the application of analytic solutions (e.g. graphs theory) to understand the possibilities of interoperability that are achievable based on the current information available. Figure 98 shows how to interoperate data format A with data format B, and the data format B with data format C, but not how to interoperate data format A with C.

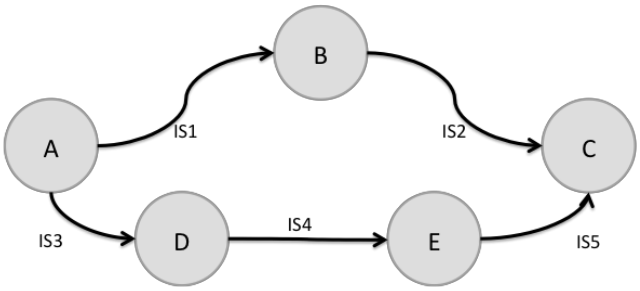


Figure 98: Interoperability Model

However this can be achieved indirectly by executing the IS between data format A and data format B and then execute again the IS to data format C, so applying, for example, graphs and path discovery, is possible to discover every possible "path" between the existent ISs. To enable this feature, the PnI platform needs a place where ISs are available and accessible at all times. This place is a repository, due to fact that repositories facilitate more efficient storage and management of resources, enabling the community to share and discover resources shared by others. There can be several repositories, and the community can, this way contribute to improve the interoperability possibilities offered by the PnI platform.

7.3 Reference Architecture – Logical View

In order to understand the full application reach of Plug and Interoperate, there is the need to devise a reference architecture that provides interoperability as a service in any environment that requires it. As explained before, PnI is a mediated approach, and so this mediator is the entity that will actually execute the interoperations deemed necessary, for two systems to become interoperable. To enable the success of this approach, a well-defined architecture is needed (Maló, Teixeira, Almeida, & Mateus, 2013) (Teixeira, Maló, Almeida, & Mateus, 2011).

The created reference architecture is divided in 3 logical modules (Teixeira, 2012): Interoperability Manager, Interoperability Repository and Execution Engine. Each of these modules has very specific functionalities and is the aggregation of other small modules that will be detailed subsequently.

- Interoperability Manager: The Interoperability Manager module is required to index all interoperability artefacts that are known and available to the Plug and Interoperate solution, as well as to provide methods to expose the PnI functionalities;

- Interoperability Repository: This module represents an Interoperability Repository, which is needed to organise all interoperability-related information required for PnI to operate. It provides methods that enable access to all known interoperability related information;

- Execution Engine: The objective of this Execution Engine module is to provide the necessary environment to handle and execute all required interoperability-related operations, so that data formats may become interoperable with one another.

An important aspect is that both the Interoperability Repository and the Execution Engine are independent modules, which means that they can be used as standalone solutions (without the use of the Interoperability Manager) to fulfil purposes different than with the PnI solution.

The created architecture is presented in Figure 99, where all the modules defined before can be seen. It presents an overview of the reference architecture for the Plug and Interoperate solution.

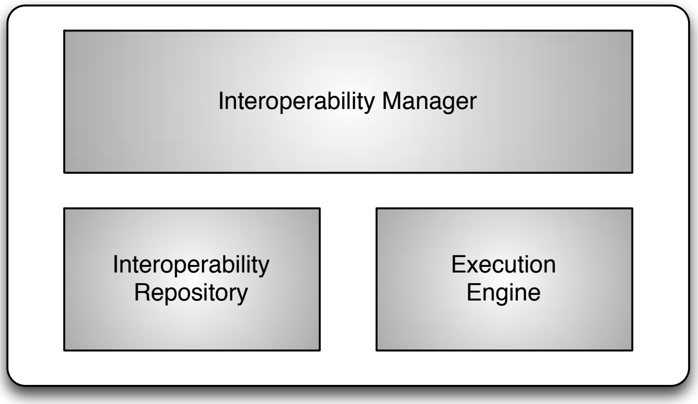


Figure 99: PnI Reference Architecture overview

The following section will provide a specification of the logical modules present in the reference architecture. Each logical module has a description of its objectives and the methods each one provides. Figure 100 presents an example of a logical module. It has sets of methods on top (API), which are methods available for other modules to use. The ones at the bottom (Caller Interface) are methods that the module uses to communicate with the other modules. Each one of the logical modules described next has a figure such as this one in order to explain its own methods.

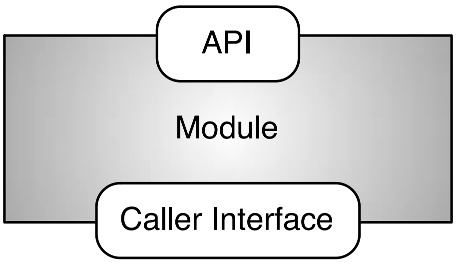


Figure 100: Example of a module with the API and Caller Interface

7.3.1 Interoperability Repository

In order for PnI to realise what it is supposed to, it needs a place where interoperability-related information can be stored and be permanently available. So, PnI needs a repository where interoperability artefacts and specifications will be stored and consulted whenever the PnI is required to interoperate two data formats. This repository that is represented as one logical module of the PnI logical architecture overview, depicted in Figure 99, will now be here specified in more detail.

For a repository to fulfil the functional requirements of PnI, it needs a set of characteristics, due to the special needs of the interoperability realm. This characteristics are: the extreme heterogeneity of all interoperability artefacts, the high dynamism of the environment, and the fact that the repository needs to be completely independent of the PnI solution, so it can be used as a standalone solution, i.e. so that any user can access the interoperability-related information stored within the interoperability repository, without the need to use the whole PnI solution. The repository includes a set of CRUD methods, which are the four basic functions of persistent storage (Martin, 1983). These methods allow both PnI solution modules and independent Users to add, update, delete or retrieve interoperability-related information from the Interoperability Repository.

This Interoperability Repository architecture was devised within a Master Thesis work (Teixeira, 2012) and is constituted by a set of logical modules that can be divided into three main groups: Interface, Repository Core and Persistence.

- Interface: For the Interoperability Repository to expose its functionalities, the definition of a set of methods is required to enable storage and retrieval of information, life cycle management, execution of queries against the repository data, etc. There are three Interface modules:

    - User Space Interface: This provides all the methods needed for the user to interact with the User Spaces within the repository;

    - Information Access Interface: This provides sets of methods to enable the manipulation of the information stored within the repository;

    - Configuration Interface: This interface allows certain aspects of the repository to be configured, such as adding new storage mechanisms to the repository, or creating new subscriptions to pre-determined interoperability artefacts.

- Repository Core: The Interoperability Repository has some modules that are responsible for all the processing of the interoperability-related information. These modules represent the repository's core and so they are responsible for managing all information within the repository and allowing several operations to be executed. Three modules compose the Core of the Interoperability Repository;

    - Metadata Manager: The Metadata Manager is the module responsible for indexing all the interoperability-related information known to the repository;

    - User Space Engine: This User Space Engine is the module that enables users to personalize the repository, increasing and enhancing the capabilities of the repository, by attaching their own functions and algorithms.

    - Notification Manager: The Notification Manager provides subscription and notification mechanisms of any interoperability artefacts, enabling repository users to be notified upon changes.

- Persistence: In terms of persistence, the repository needs a set of storage mechanisms and means to organise them and understand which interoperability artefacts are stored in each storage mechanism. There are two Persistence related modules:

    - Persistence Management: The Persistence Management module handles the persistence of the actual interoperability artefact files and manages the storage mechanisms available to the Interoperability Repository;

    - Storage Mechanism: This module represents one available storage mechanism. Several instances of this module can be part of the repository.

The detailed architecture of the interoperability repository is depicted in Figure 101, showing all its logical modules along with the set of methods each module use or provide to other modules.

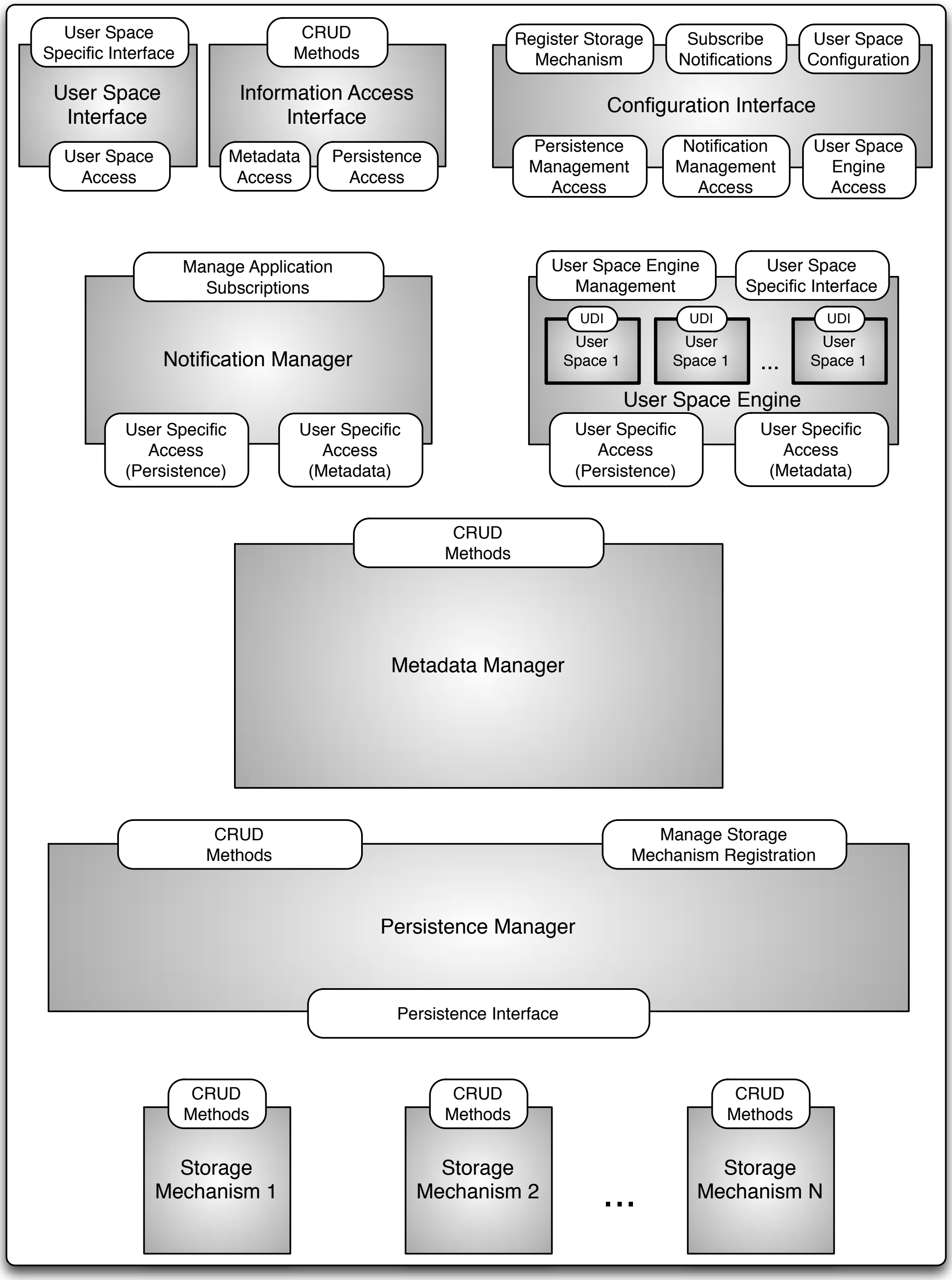


Figure 101: Interoperability Repository Architecture

7.3.1.1 Storage Mechanism

In terms of persistence, the Interoperability Repository needs to have different types of storage mechanisms available, due to the heterogeneity of the interoperability artefacts, i.e. some artefacts may require specific storage mechanisms to be stored. But to accomplish this, each storage mechanism needs to register itself with the Persistence Management module, so the repository knows it exists and which artefacts can be stored inside. As described before, several instances of this module can exist with the repository and an example is presented in Figure 102. It presents CRUD methods in the API, so the repository can store or delete the actual interoperability artefact files.

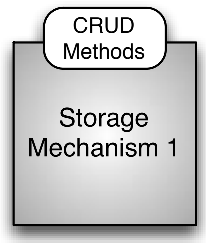


Figure 102: Example of a Storage Mechanism instance

7.3.1.2 Persistence Management

The Persistence Management module is responsible for the management of the persistence mechanisms being used by the Interoperability Repository. It also keeps track of which files are stored within each storage mechanism. Whenever this module receives a file it stores it in one of the storage mechanisms available. The Persistence Management module and its methods are presented in Figure 103.

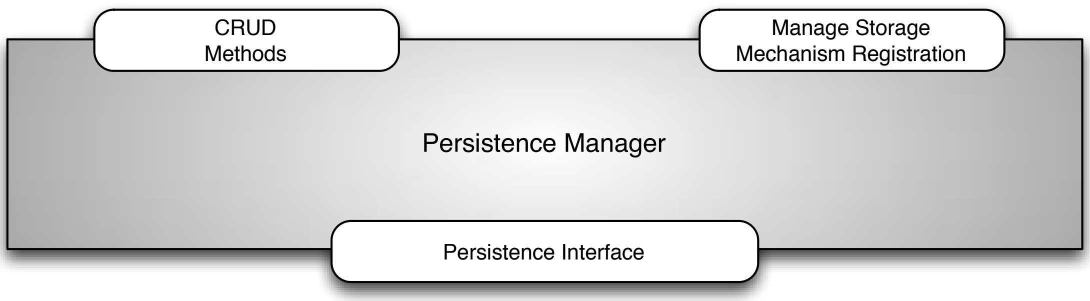


Figure 103: Persistence Management module

This module indexes the files with the storage mechanism where it is stored; it is also where the storage mechanisms are registered, so it can always know which ones are available. Another important characteristic of this module is that it is the one responsible for choosing which storage mechanism to use whenever a new file needs to be stored. That choice is based on the type of the interoperability artefact, considering which storage mechanisms are available.

This module has a set of CRUD methods, which allow other modules to add, update, delete or retrieve information from this module. The Manage Storage mechanism registration is a set of methods to subscribe, unsubscribe or manage the storage mechanisms available. The module also has a persistence interface (like a Caller Interface), which allows communication with the storage mechanisms in order to request the addition or removal of a specific file.

7.3.1.3 Metadata Manager

The Metadata Manager is the logical module responsible for the representation of all interoperability artefacts. This is where all metadata related to heterogeneous interoperability artefacts is indexed. Figure 104 presents the Metadata Manager module and its methods.

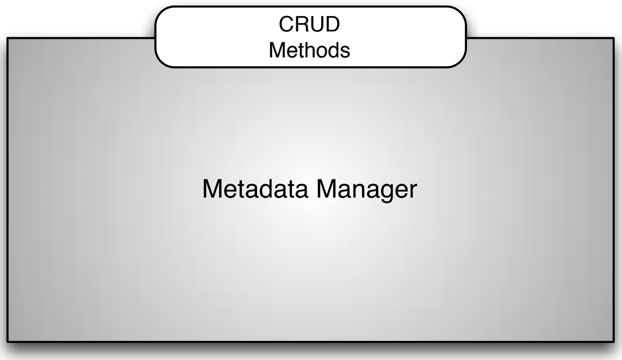


Figure 104: Metadata Manager module

Interaction with this logical module mainly consists of basic adding, retrieving, updating or deleting (CRUD methods) information about interoperability artefacts. To have the meta-information of all artefacts means that it needs to use unique identifiers. The metadata manager is responsible for generating these unique identifiers so each interoperability artefact has its own identifier. This Metadata Manager module possesses CRUD Methods on its API, in order to enable the other modules to manipulate the information within.

7.3.1.4 Notification Manager

This module is responsible for notifying a repository inquirer whenever there is a change in the artefacts within the repository. This notification services is useful for several modelling applications that are performing complementary operations with the same artefact. It is also important to maintain the interoperability specification being used within the environment updated. Figure 105 shows this Notification module with its own methods.

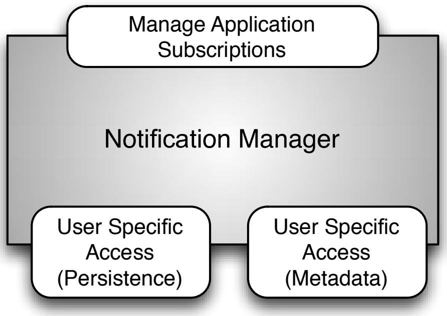


Figure 105: Notification Manager module

This module possesses one API method, called Manage Application subscriptions, which exists in order to provide the methods to manage subscriptions. It allows an inquirer to subscribe to one or several artefacts and, when there is a change in the subscribed artefacts, it notifies the application.

In the Caller Interface, it possesses two methods: User Specific Access (Persistence) and User Specific Access (Metadata) that serve as a connection between this module and both the Persistence Management module and the Metadata Manager module, so the Notification Manager module can notice whenever metadata related to an interoperability artefact, or even the actual file (stored in one of the storage mechanisms available) is changed.

7.3.1.5 User Space Engine

The User Space Engine is the module responsible for the execution of all user-defined operations. The module uses the User Specific Access Interface (as a Caller Interface) to communicate with the other modules. This User Space Engine can perform more complex tasks, such as traverse algorithms and statistic determinations. Figure 106 shows the User Space Engine module and both its API (User Space Engine

Management and User Space Specific Interface) and Caller Interfaces (User Specific Access for the Persistence and for the Metadata Manager).

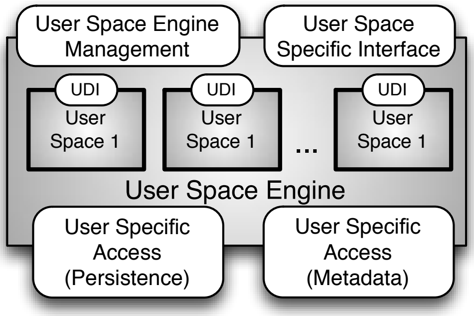


Figure 106: User Space Engine module

This User Space Engine module provides User Spaces so as to allow one user to add its own algorithm to the repository. The repository provides two sets of methods in its own API: the User Space Engine Management, which are the methods provided to manage the user spaces, such as the addition or removal of an algorithm from one User Space; and the User Space Specific Interfaces, which are the methods provided by each algorithm, presented in Figure 106 as the UDI - User Defined Interface, of each User Space. Each User Space algorithm may use the User Specific Access methods (The User Space Engine Caller Interface) to use the functionalities provided by the other modules.

In order to allow interaction, the repository needs to support a well-defined interface with different methods. Each method may have different abstraction levels, as well as distinct objectives. From a more functional point of view, the repository needs to be associated to a set of services. These services can be used to query the repository for interoperability artefacts, allowing the inquirer to enhance the repository's interoperability specification, add/remove entities or relationships to/from the repository, and attach new information about an existing artefact. Thus, three Interface modules are: the User Space Interface, the Information Access Interface and the Configuration Interface.

7.3.1.6 User Space Interface

This User Space Interface module provides access to the User Space Engine module, by means of the User Space Specific interface. This enables a user to interact with the contents of specific user spaces by using the user space interface provided by the Execution Engine. Figure 107 shows this Interface module with its own methods.

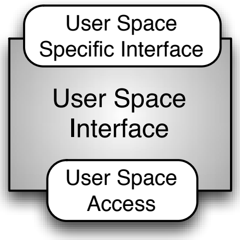


Figure 107: User Space Interface module

7.3.1.7 Information Access Interface

This Information Access Interface module, represented in Figure 108, offers methods to allow repository users to add, retrieve, update or delete artefacts. These CRUD methods will work with both the actual files and the metadata associated with them, which means that this module is the one that will differentiate the files from the metadata, in order to standardize the process of adding and retrieving artefacts. This interface allows a repository user to use any CRUD method to interact with the artefact specific information. These are relevant methods because some applications may need to consult the information regarding specific artefacts.

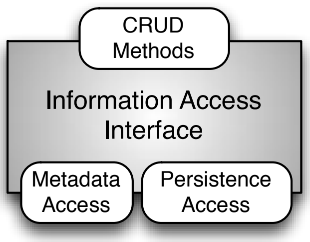


Figure 108: Information Access Interface module

This Interface module is responsible for the interoperability artefacts manipulation within the repository. It contains all the methods available to manipulate both actual files (stored via the Persistence Management module) and the artefacts metadata (stored in the Metadata Manager module). It provides CRUD methods, present in the API of the module in Figure 108, to the end users and then, based on what the user wants, it may communicate with both the Metadata Manager and Persistence Management module through the Caller interface, Metadata Access and Persistence Access, respectively.

It must be noted that, in any of the services provided by this module, it is this module that controls the flow of information. Another functionality of this module is the capability to interpret the files added to the repository, in order to extract the meta-information from it.

7.3.1.8 Configuration Interface

The Configuration Interface module allows the repository system to be configured. The objective of this module is to offer methods to manage the repository functionalities, such as the configuration of the user spaces, the subscription of notifications and the registration of new storage mechanisms. Figure 109 presents this module and its own methods.

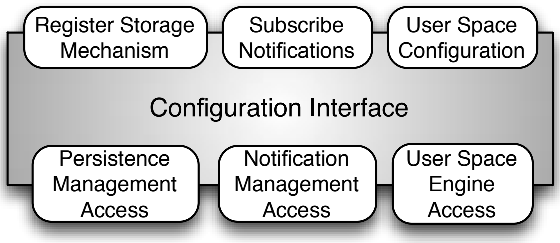


Figure 109: Configuration Interface module

One can register and unregister storage mechanisms in the Persistence Management, in order to provide the repository with more persistence options. It also allows a repository user to subscribe/unsubscribe any artefact present within the repository, meaning that, when that specific artefact changes, the notification manager notifies the user of the change. This module also provides communication support to the Execution Engine module. It gives users the possibility to configure the User Spaces provided by that module, and the user may be able to add, retrieve, delete or update algorithms from the User Spaces. In order to accomplish all of this, the Configuration Interface module possesses a set of Caller interfaces, which provide access to the other repository modules.

7.3.2 Execution Engine

The Execution Engine is another module of the PnI reference architecture and, as explained before, it can provide its functionalities independently of the other PnI modules. The Execution Engine is needed within the PnI architecture to handle and execute the interoperability specifications known by the PnI solution. This Execution

Engine was initially created in a Master Thesis (Pereira, 2012) and is composed of three modules, which are depicted in Figure 110, the Interoperability Specification Handler, the Execution Machine and the Model Handler.

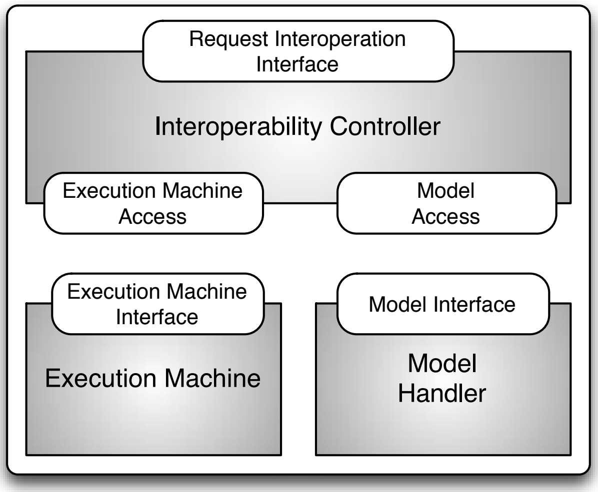


Figure 110: Execution Engine Architecture

7.3.2.1 Interoperability Controller

As the name suggests, the Interoperability Controller is the module that controls and orchestrates the execution of interoperability specifications. It is responsible for interpreting, exporting and providing both data and instructions to the execution machine. The interpreting function is designed to receive the IS, analyse it and create a list of instructions that are readable by the execution machine. In case the IS instructions require elements from data formats, it is also this module that provides them to the Execution Machine module. The exporting function does the opposite, since it gathers and organises the data received from the Execution Machine output, as defined by the interoperability specification, so it can be exported. Figure 111 shows the Interoperability Controller module, with the methods that enable the request of an interoperation and the communication with the Execution Machine module.

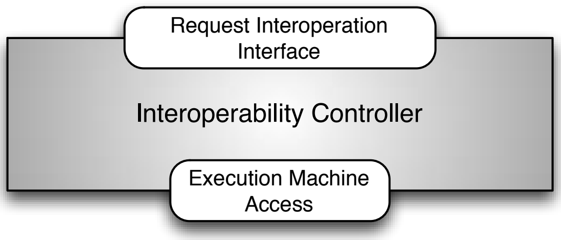


Figure 111: Interoperability Controller module

7.3.2.2 Execution Machine

The Execution Machine module is responsible for executing the IS instructions provided by the Interoperability Controller module. This module receives and executes the provided instructions, as well as elements from data formats whenever the interoperability specifications instructions require them. Figure 112 shows this module, with the Interface that will allow the Interoperability Controller to request the execution of instructions, and the delivery of the data generated from them.

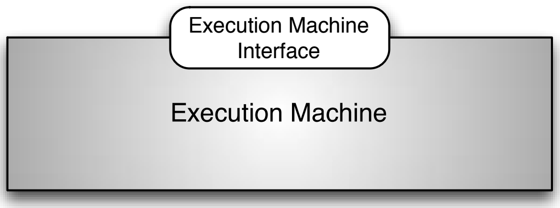


Figure 112: Execution Machine module

7.3.2.3 Model Handler

The Model Handler logical module is responsible for interpreting, exporting and providing model data to the execution engine. The interpreting function is designed to receive model data, analyse it and construe model information that is readable by the system. The exporting function does the opposite, gathering the system model data and turning it into model data, so it can be exported as an output of a model transformation. It is also responsible for providing model data to the execution engine.

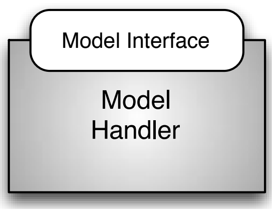


Figure 113: Model Handler module

The workload of this logical module starts with the loading of a model data that will be stored in memory, through the Model Interface API. This API provides methods to load, set and get model data to the system. Once the model data is loaded to memory, the module will provide methods to get and set specific model data information. At the end of the execution process, this module is responsible for providing methods to retrieve the translated model data.

7.3.3 Interoperability Manager

The Interoperability Manager module is yet another module of the PnI reference architecture. It provides mechanisms to index all information about the interoperability artefacts, such as the type of interoperability artefact it is, the source and target data format of each interoperability specification, where each artefact is being stored, which execution engine can execute each interoperability specification, etc.

Another aspect that this module is responsible for is control of the flow of interoperability specifications executions. It needs to know which engines can be used at each moment, and which operations they can execute, so it can orchestrate the execution of a set of operations in order to interoperate two data formats. It also provides a decision support system that gives detailed information on how to improve interoperability in a given environment. To accomplish all this, the Interoperability Manager possesses two logical modules, portrayed in Figure 114.

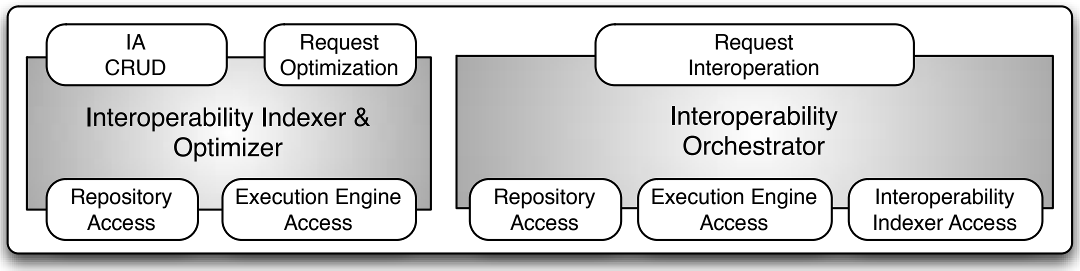


Figure 114: Interoperability Manager Architecture

7.3.3.1 Interoperability Indexer & Optimizer

The Interoperability Indexer & Optimizer, represented in Figure 115, is yet another module of the Interoperability Repository and is responsible for two specific tasks:

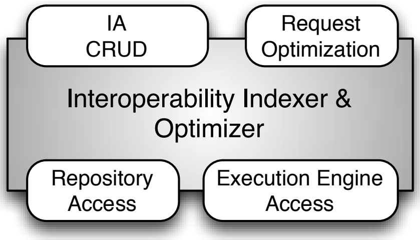


Figure 115: Interoperability Indexer & Optimizer

1. Indexing Interoperability Artefacts: This module is required to index all interoperability-related information available to the PnI solution. It needs to understand which interoperability artefacts the PnI solution has access to and indexes information, such as the source and target data format of each interoperability specification and how it can be executed. In order to index the interoperability artefacts, it needs access to both the repositories, to index the interoperability artefacts, and the execution engines, to know which can be used to execute the interoperability specifications, and for that it uses the methods Repository Access and Execution Engine Access, respectively.

2. Optimizing Interoperability: This module will also provide a way to enhance overall interoperability in a given scenario. This functionality is a crucial part of the PnI solution as an interoperability assistant, by providing a decision support system on how to ensure interoperability between heterogeneous data formats. This functionality aims to maximize the number of interoperable data formats while using the minimum number of interoperability specifications. To enable this functionality, the UI algorithm was devised, which is a reference algorithm that analyses a given scenario, understanding which data formats exist and the interoperations already existent in that scenario. Then with that information, it advises on how to achieve maximum interoperability by recommending the set of interoperability specifications, not yet available to the PnI solution, that, if created and applied, will achieve the maximum interoperability possible within the specific scenario.

The Request Optimization method allows users, or even the Interoperability Orchestrator module, to query on how to optimize interoperability within a scenario, even in the case where there are only two data formats. The IA CRUD API method provides a way for users to Add, Read, Update and Delete Interoperability Artefacts within the PnI Solution.

7.3.3.2 Interoperability Orchestrator

The Interoperability Orchestrator is the module that orchestrates all interoperability execution requests on the PnI solution. This is the module that receives, directly from the user, the requests to interoperate given data formats, and that starts all processes leading to the attainment of interoperability between the given data formats.

The Interoperability Orchestrator controls the flow of information within the PnI solution, i.e. when a user requests the PnI solution for an interoperation, this module asks the Interoperability Indexer & Optimizer module, how to achieve interoperability between the given data formats and, if it has a way indexed, it informs the Interoperability Orchestrator, not only of which interoperability specifications (one or more) to use, but also in which Interoperability Repository they are stored and which Execution Engines can be used to execute each interoperability specification. The Interoperability Orchestrator will then access the Interoperability Repository to retrieve the needed interoperability specification, and then request one Execution Engine to execute it. To enable those requests, the Interoperability Orchestrator module possesses interfaces to communicate with the other PnI modules, as shown in Figure 116.

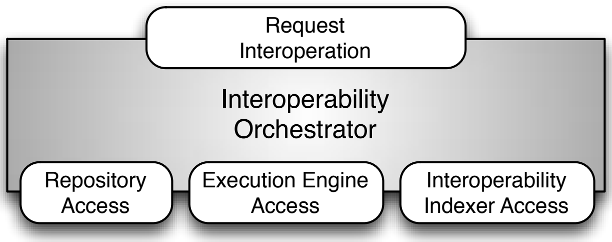


Figure 116: Interoperability Orchestrator module

7.4 *Reference Architecture* – Functional View

The above-described view of the PnI Reference Architecture represents the logical aspects of the architecture. It shows all the logical modules' that compose the architecture, explaining each of the modules' functionalities. The objective of the PnI solution is to act as a communication mediator, in order to ensure interoperability between systems. However, there is a problem with using the mediator pattern, because the mediator manages all the interaction between components, so this component could easily become very complex and hard to maintain (Gamma, Helm, Johnson, & Vlissides, 1994).

Another problem with this mediator pattern is that, because the mediator is the only connection point between the components, interactions become heavily dependent on the mediator. If the mediator does not function properly, no systems will be able to send or receive requests to/from each other. The performance and reliability of the mediator component will have a deep impact on the functioning of the system as a whole.

Therefore, a distributed approach with multiple points of entry is needed, so as not to compromise the achievement of interoperability between systems. However, this is something that cannot be represented in the architecture's logical view. To solve that, a Functional View will be represented, which presents the functional aspects of the PnI solution.

Considering that the PnI solution is mostly needed in scenarios where different systems need to communicate with each other and the PnI Solution mediates that communication, it needs to be distributed and have multiple instances of its own modules, thus enabling the solution to spread throughout the environment. This means the environment can have multiple Interoperability Repositories, each one with its own interoperability related information; it can also have multiple Execution Engines, to support the execution of interoperability specification in different languages and using different engine types; and it can also have multiple Interoperability Managers, although each one will be a replica of the others, not due to a replication mechanism, but because each instance of the Interoperability Manager will index the interoperability information present in all repositories existing in the environment as well as which execution engines are present within the environment and what each of them can execute, regardless of whether the modules share the same location or not.

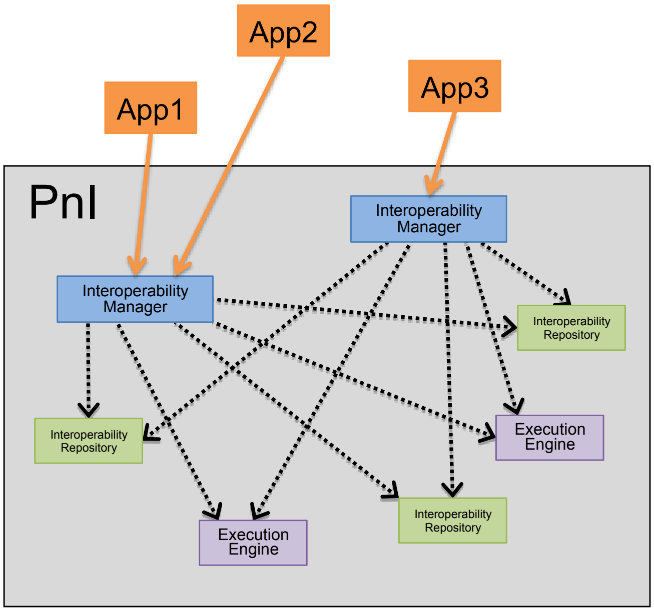


Figure 117: Distributed PnI

As shown in Figure 117, to enable a distributed PnI, there is the need for several module instances spread throughout the environment with communication mechanisms that provide ways for communication between module instances. So, a communication platform was devised to provide the PnI solution with the communication infrastructure needed for it to work seamlessly in this kind of environments.

In order for Applications to take advantage of the functionalities provided by the PnI overall solution, they need to access it. However, the only module that works as an entry point to the PnI solution is the Interoperability Manager. That is depicted in Figure 117, which shows instances of the Interoperability Manager module (PnI Solution entry point) sharing the accesses from the applications.

The idea is that the platform provides the communication infrastructure to the PnI modules instances, so that they can communicate among themselves. This communication platform is based on the one created in the European Research Project CuteLoop FP7-ICT-216420. It provides Zero configuration networking, allowing a PnI module instance to connect to a network automatically, i.e. without manual operator intervention or special configuration servers. It does not have a Single Point of Failure, and a unique ID identifies each module instance, so that it can change its localization address while keeping a constant identification number. For this to work, each Communication Platform is autonomous and operates independently and asynchronously from all other platforms.

The Communication Platform also needs to provide means for the PnI module instances to interact with each other, so it needs to create an overlay transport network which allows a PnI module instance to interact with others, within the same network, even when some of them are behind firewalls and NATs, or use different network transports.

In order to provide the PnI functionalities to Applications outside the network, the Communication Platform was enhanced with a P2P Service Exposer created as a Master Thesis (Melo, 2012). This Service Exposer enables Applications that are not members of the network to access PnI functionalities, by placing special sets of peers at the boundaries of the network so as to facilitate outside communication. If an application wants to access some functionalities provided by the PnI solution inside the network, the service exposer will gather the request, search the network for the PnI module instance that provides the service, request it, and deliver the response to the requester application. This sequence of events will occur in an asynchronous way, because it cannot be guaranteed that both the functionalities and the processing related to its management happen in a limited period of time.

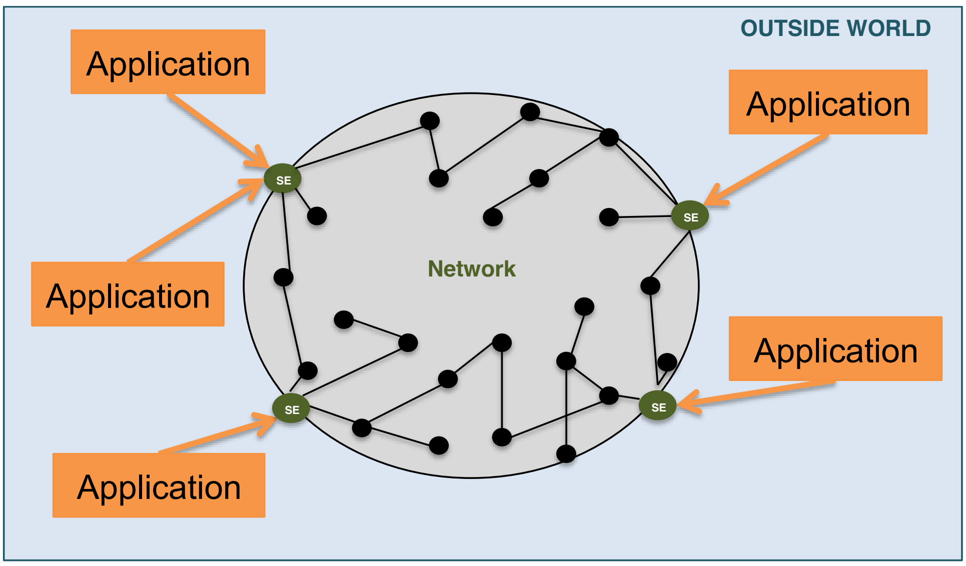


Figure 118: Exposing Services for Applications Outside of Network

8

Conclusions and Future Work

"Any sufficiently advanced technology is indistinguishable from magic"
– Arthur C. Clarke, Science Fiction Writer

Conclusion is an important part of a thesis for the reader. It helps to have a good picture of what a thesis is about – focusing on its highlights –, leading to a better understanding of the message that the writer wants to give. The conclusions focus on showing the reader what was achieved, reaffirming the statements and issues, giving insights of the results, and of the issues.

This section also describes future work. Future work is all about the topics and issues that were uncovered during the thesis, but were somehow out of the scope of the work. They also comprise the next steps of evolution that the work can take, such as a new type of study to be made, testing the solution with different parameters, etc.

Also in the future work, the insights of the author are presented, ideas as to where to continue the research started in this work, so that someone can continue the research begun here. This way, a community of interest in this topic can be created, and long life to the H&S Interoperability ensured.

8.1 Conclusions

The research work of this thesis addresses the (data) interoperability problem in environments like the Internet-of-Things, i.e. heterogeneous settings with a considerable number of disparate systems that support a variety of different data formats and that need to be made interoperable. Due to this heterogeneity of systems, which can be from different manufacturers, legacy systems, different technologies, etc., standardisation is not possible.

In these cases there exists interoperability approaches that consider standardisation unnecessary. For instance, the number of (different) systems in need of interoperating might be relatively small and so it might be possible to straightforwardly interoperate systems altogether. In such cases, it is possible to promptly interoperate systems, by using interoperability approaches such as Point-to-Point (P2P) interoperability. P2P Interoperability is characterised by direct system-to-system connections that establish interoperations between different systems.

The P2P interoperability approach is applied in an interoperability scenario, consisting of a collection of distinct systems, where systems are required to exchange information with each other, in order to be able to achieve its objective. The process of exchanging information within an interoperability scenario is an interoperation that, in the P2P interoperability approach, defines the ability of two systems to exchange data by following a specific data format. To enable that exchange, both systems (source and target system) must use the same Data Format, which defines how the (shared) data is represented and structured. This Data Format needs to be supported at both ends. The source system must have the capability to send information in a data format that the target system is able to accept. This requirement of systems is achieved by two (interoperability) artefacts: data exporters and data importers. These concepts, among others, were duly formalised, both mathematically and graphically.

To study the behaviour of P2P interoperability, an algorithm was defined to maximise interoperability in a scenario using the P2P Interoperability approach. This algorithm uses two views of the interoperability scenario at work: one expressing the export capabilities of each system, and another to indicate the supported data formats that each system uses to import data. Besides these two views, the algorithm also requires the indication of which scenarios are modifiable. The goal of the algorithm is to maximize the number of possible data flows, from systems capable of export data to systems able to import data. The number of possible data flows increases by adding new export or import capabilities to the systems, when possible, in order to match the data formats used to export, with the ones used to import.

An analysis of the limitations of P2P Interoperability regarding its application to large-scale environments requires the definition of the key parameters that are needed to specify an interoperability scenario. These parameters are used to categorize the interoperability scenarios used as input for the algorithm. The definition of a number of metrics was also deemed necessary, in order to focus the scope of analysis and allow the comparison of the results of the algorithm.

Two metrics were identified as being able to express interesting characteristics of an interoperability scenario: (1) Number of Interoperability Artefacts (NIA), representing the amount of interoperability artefacts used in a scenario to enable interoperability between participating systems of the scenario; and (2) Interoperability Coverage, which represents the ratio between the number of systems that are currently interoperable and the theoretical maximum number of systems that could be interoperable, given the conditions that characterise the interoperability scenario. Yet another metric - Interoperability Absolute Coverage - was defined to complement the Interoperability Coverage, giving the notion of the real percentage of systems in the scenario that are interoperable. The study of P2P Interoperability consists in the generation of the probable interoperability scenarios that can occur in a real environment and in the analysis of the metrics that characterise the interoperability of the environment. For this simulation to take place, a set of inputs needs to be cycled, so that the desired metrics for the study are obtained.

The interoperability scenarios are generated based on scenario characteristics such as: (1) the number of different systems present in the environment, (2) the percentage of heterogeneous data formats being used by those systems, and certain systems' properties, such as (3) the percentage of systems capable of exporting data, (4) the percentage of systems capable of importing data, and (5) the percentage of systems capable of being modified, meaning that they can be changed in order to increase their importing or exporting capabilities.

The first and most important observation to acquire through the execution of the simulations is the dependability of the "% to maximum possible" line on the modifiability characteristics of the systems. The conclusion that can be drawn is that the possibility to achieve full interoperability of all systems with one another in the environment is proportionally related to the modifiability of the systems present within the interoperability scenario.

All of the P2P simulation results were approximated to a quadratic function, more specifically, to the coefficient that controls the speed of increase. This quadratic behaviour shows that, when the number of systems in an interoperability scenario increases, the number of needed interoperability artefacts (NIA) also increases at a fast rate. The individual analysis of each parameter variation proved that each one is independent of the other in relation to the baseline.

The Modifiable characteristic has a small increase over the baseline (which is the best case in terms of NIA), not only due to the number of modifications, but more importantly due to the impossibility of achieving full connections when there are few modifiable systems, and because it is possible to reach a good optimization when there are a lot of modifiable systems. So, it is a parameter that does not cause a big impact in terms of NIA, but more on the coverage parameter.

Regarding the Exporter and Importer combined parameter and the formats, they have a huge impact on the baseline. This is directly related to the fact that these parameters control the number of needed connections in the interoperability scenario - which is the big problem related to the NIA of P2P Interoperability. The Point-to-Point (P2P) Interoperability approach reveals a fundamental and crucial problem: the number of non-modifiable systems limits the maximum possible interoperability coverage achievable in an interoperability scenario. The more pairs of non-modifiable systems with disjoint sets of exporting to importing capabilities that exist, the less maximum possible interoperability that is possible to accomplish.

The P2P Interoperability approach presents yet another problem: the greater the number of required interoperations, the more interoperability artefacts are required – for such interoperations to be feasible, obviously. The Number of Interoperability Artefacts (NIA) – exporters and importers – grows when more and more systems are required to interoperate in an interoperability scenario. Furthermore, the Number of Interoperability Artefacts (NIA) grows faster – in quadratic degree – as the number of systems in interoperability scenario increases, making it not suitable on the large-scale applications, where large number of systems exists.

For solving these inherent problems of this typical interoperability approach (P2P Interoperability), the author looked at other fields of application that could provide help to tackle these limitations. The field of study, that required time to understand, because of it similarity with interoperability between systems, was the transport & logistics field.

Transportation & logistics systems share many of the basic needs of a system to support data interoperability. Both systems focus on the move of “things” from source to target: data interoperability supports systems on the move of data, and transport & logistics supports systems on the move of goods, information, people, etc. And both systems are concerned with the best way to connect many origins to many destinations, optimally all: data interoperability supports systems connecting many data consumers to various data sources and transport & logistics supports systems connecting several points of origin to lots of points of consumption.

Transport & logistics is a vast area, but there are some specific cases that are more similar to this problem than others, such as the commercial air transportation systems. The goal of commercial air transportation systems is analogous to that of data interoperability systems: to maximise the interrelation of constituent entities – ultimately to interlink all designated entities – while efficiently accomplishing that interconnectivity. Commercial air transportation connects – via global, regional and local carriers – all airports in the most efficient form; (transportation) efficiency means less resources (airplanes), but it could also mean some best operational scheme considering costumer demand, number of flight hops, passenger flow, partnerships, etc. Likewise, data interoperability support systems aim to maximise the data exchange links between constituent systems – optimally, interoperating all designated systems at the data level – in the most efficient form; (interoperability) efficiency means less resources (number of interoperability artefacts) but also some best operation interoperability scheme considering data-sharing demand, number of interoperations, etc.

Therefore, naturally, the challenges of commercial air transportation systems are very similar to those of data interoperability support systems, namely: (1) How to reach out for a full coverage of worldwide airports, considering that airports cannot service all kinds of airplanes and cannot be modified to support them, which relates to the equivalent challenge of how to interoperate all systems – especially those that cannot be modified to support (import/export) alternate data formats – in data interoperability environments; (2) How to efficiently achieve interconnectivity between airports of a commercial air transportation system using the least resources (airplanes) possible, which relates to the comparable challenge of how to interoperate (at the data level) the many systems of an interoperability environment making optimal use of resources (i.e. interoperability artefacts).

The approach used in commercial aviation is based on the Hub-and-Spoke paradigm (also sometimes referred to as spoke-hub paradigm), which made possible a global commercial air transportation network that is facilitated by major hubs distributed around the globe. This network structure of airport hubs makes it possible for a passenger to cross the whole world, from a given source to destination, in few flight hops. Also, as a result of the extensive networks made possible by hub-hops, carriers are able to attract passengers and, with tight scheduling, to meet passengers' preference for single-carrier service.

The Hub-and-Spoke (H&S) approach allows airliners to service a wide number of city-pairs, without additional resources – particularly airplanes, the key and most costly asset in air transportation – than do point-to-point connections by hopping in multiple (hub) airports. Analogously, and from the perspective of a data interoperability support system, maybe it might also be possible to efficiently interoperate many (source to target) data systems by "hopping" formats – from source-system format to target-system format via other formats. In this way, and as in the hub-and-spoke approach of commercial air transportation systems, it could be possible to achieve (full) data connectivity (i.e. system coverage) in the environment, without the need for lots of interoperations between constituent (source to target) data systems.

In short, the new interoperability approach – powered by the concepts of mediated interoperability and interoperability compositions – might provide the leading edge to data interoperability that moves past the challenges posed by the point-to-point approach to interoperability. To start with, the proposed interoperability approach might establish the best interoperability, in an arbitrary data-exchange environment, using fewer resources when compared to point-to-point interoperability. Also, it might make it possible to interoperate all systems – especially those that cannot be modified to support (import/export) other data formats –, thus putting them all as participants in the data- exchange environment; P2P interoperability fails to accomplish this. These two important features might provide a general solution to the data interoperability problem, one that is efficient, effective and able to scale.

Hypothesis: The combined use of interoperability mediation and interoperability compositions (e.g. as defined by the H&S Interoperability approach) allows the fundamental challenges of data interoperability, i.e. to achieve full interoperability coverage with a suitable number of interoperability artefacts, to be addressed.

The hub-and-spoke paradigm (or model, or network) is a system of connections arranged like a chariot wheel, in which all traffic (interoperations) moves along spokes connected to the hub at the centre. In an abstract sense, a location (object) is selected to be a hub, and the paths that lead from points of origin (data source system) to points of destination (data target systems) are considered spokes. The terms “hub” and “spoke” create a pretty vivid image of how the hub-and-spoke interoperability approach works: a hub is a central format to where interoperability is routed through, and spokes are the many interoperations in between hubs and other available formats.

A novel approach to address the issues highlighted in the Point-to-Point Interoperability approach is proposed. This approach, called Hub-and-Spoke (H&S) Interoperability, resorts to certain concepts, inspired by the Commercial Air Transportation domain, to handle Interoperability in the domain of Information Systems. By doing so, the presented approach aims to solve the Coverage-Modifiability problem and to minimize the NIA-Connections problem. Regarding the first problem, using a mediated solution where no system is subject to modifications solves it. In respect to the second problem, a solution is proposed that reuses interoperability artefacts to render interoperable, systems that are non-interoperable.

The Hub-and-Spoke Interoperability concept addresses these fundamental challenges of data interoperability. First, Hub-and-Spoke Interoperability makes it possible to reach full interoperability, including making interoperable those systems that are non-modifiable. This way, the Hub-and-Spoke Interoperability approach enables interoperability between pairs of non-modifiable systems that have disjoint sets of exporting to importing capabilities. Consequently, it is possible to interoperate all systems (data producers to data consumers) and to achieve full interoperability coverage, regardless of the number or ratio of non-modifiable systems in the scenario.

Then, Hub-and-Spoke Interoperability makes it possible to efficiently achieve full interoperability in an interoperability scenario. Hub-and-Spoke Interoperability establishes a scheme that turns interoperable the many heterogeneous systems of an interoperability scenario, using an adequate number of interoperability resources (interoperability artefacts). Hub-and-Spoke Interoperability makes linear the Number of Interoperability Artefacts (NIA) vs. the number of systems ratio for achieving full interoperability, and with a much lower quantity than that of the equivalent point-to-point interoperability, and that is acceptable to be implemented in practice.

The H&S approach uses all the definitions used by the P2P Interoperability approach: interoperability scenario, system, interoperation, importer, exporter and data format. However, it adds new concepts that derive from the fact that it is a mediated approach, such as the Mediator, which is the agent that acts as the central hub for interoperability support in the H&S approach. The Mediator needs to have Adapters: a Reader, which enables the mediator to read the information in a specific Data Format, and a Writer which enables the mediator to write the information in a specific Data Format.

Another key concept in the H&S Interoperability approach is the concept of Interoperability Function (IF), which specifies all the actions that must be executed to perform the information transfer from a source to a target Data Format. IFs allow systems to interoperate without using the same data formats, by mapping the source and target data formats within the mediator. A path is yet another important concept, defined in this approach as a set of IFs needed to interoperate two data formats that do not have an explicit and direct way to be interoperable. The interoperation concept used before was slightly changed to take these new concepts into account, so that interoperation in the H&S approach is the grouping of the system exporters, the system importers and mediation, which in turn groups the mediator readers, writers and paths between data formats.

An algorithm to maximize interoperability in a scenario using the Hub-and-Spoke Interoperability approach was also defined. This algorithm requires a representation of the interoperability scenario to be analysed, where this representation must be able to express the import and export capabilities of each system. The goal of this algorithm is to enable the data provided by each system with export capability to be used by any system with import capabilities.

The algorithm focuses on a continuous deployment of Interoperability Functions into the scenario until the goal is achieved. Each IF deployed in the scenario corresponds to the IF that will introduce the greatest interoperability improvement, i.e. the one that will enable connection between more systems. When no more improvement can be introduced in the scenario the algorithm stops, returning the set of Interoperability Functions applied to the initial scenario, as well as the metrics corresponding to the scenario produced by the deployment of that set of Interoperability Functions.

Summing up, the new interoperability approach – powered by the concepts of mediated interoperability and interoperability compositions – might provide the leading edge to data interoperability that moves past the challenges posed by the point-to-point approach to interoperability. First, the proposed interoperability approach might establish the best interoperability, in an arbitrary data-exchange environment, using fewer resources when compared to point-to-point interoperability. Also, it might make it possible to interoperate all systems – especially those that cannot be modified to support (import/export) other data formats –, thus putting them all as participants in the data- exchange environment; point-to-point interoperability fails to accomplish this. These two important features might provide a general solution to the data interoperability problem, one that is efficient, effective and able to scale.

An evaluation of this new proposed approach was performed, where the same methodology for simulations was applied, and then the two approaches were compared. This evaluation process confirms that the Hub-and-Spoke Interoperability approach, as designed, tackles the problems identified in large-scale interoperability scenarios, reducing the NIA and increasing the Coverage.

The NIA Problem, which was the large number needed in P2P Interoperability, was overcome, by reducing over six times the NIA in large-scale Interoperability scenarios.

Also, the type of behaviour of the two approaches changed. The quadratic behaviour of P2P, leading on an even larger scale to a bigger problem (with the increase of the NIA), is transformed into a linear behaviour with the H&S approach, resulting in a much less dependability on the number of systems presented in an interoperability scenario.

This extensive reduction of the NIA does not come alone. It is accompanied by an increase in the coverage of an interoperability scenario. But this rise is not a quantified increase; it is a total increase, because the H&S Interoperability approach always achieves the maximum possible/needed interoperability desired.

However, this good result implies a number of associated potential problems, such as its application in small-scale environments, where this solution is possibly not always a good one. This is due to the fact that, in this type of environment, P2P Interoperability can achieve good interoperability with a small number of NIAs. But, this small number also presents a problem, which is the fact that, if there are a relevant number of non-modifiable systems in the environment, total coverage cannot be achieved, whereas the H&S approach always guarantees that.

Solving real interoperability problems with a conceptual approach is not possible. A conceptual approach needs to have a specification, so it can be applied in the real world. The H&S Interoperability approach was the foundation for a new interoperability solution: the so-called, Plug'n'Interoperate solution.

The Plug'n'Interoperate (PnI) is a Plug'n'Play-like mechanism for Data Interoperability. PnI assumes that it is not possible/required/needed to impose a common data representation (standard data format, common ontology) but rather it accommodates (and welcomes) differences and plurality of systems (i.e. data heterogeneity). PnI adopts the same basic principle of Plug'n'Play auto/self-configuration so as to automate, as much as possible, the configuration and participation of systems in the Interoperability environment. The solution is made possible by the existence of "interoperability drivers" (called Interoperability Specifications, or ISs), which define translations between data formats, and that are used by an interoperability support system – the so-called interoperability mediator – to make interoperability of heterogeneous data systems possible. In a Plug'n'Interoperate interoperability environment, systems plug (into the interoperability support system) and are promptly interoperable with others!

The PnI solution makes it possible for systems to be interoperated without the need to be modified/remanufactured, because developers/manufacturers only need to provide interoperability specifications of their own system. Moreover, the PnI makes it possible to reach full interoperability (i.e. interoperate all, data producers to data consumers, systems), including making interoperable those systems that are non-modifiable (due to being proprietary/closed/legacy/etc. or simply because the costs for such changes make them prohibitive). Finally, PnI makes it possible to efficiently achieve interoperability in a given interoperability scenario by using only an adequate (affordable, acceptable) number of interoperability resources.

8.2 Future Work

The whole of the work, as presented before, has been directed at solely proving the hypothesis. That is, the proposed concept was defined and its validation was performed in the specific scope and within certain boundaries as to prove the hypothesis. This means that some paths were identified but deliberately not followed, as they did not present value for the strict purpose of assessing the hypothesis. Future developments might explore further into these and other dimensions of work.

The analysis of the problem was performed with the assumption that one wishes to make fully interoperable a given interoperability scenario looking at it from start. That is, a given interoperability scenario is set – with modifiable systems, non-modifiable systems and it exporters and importers – and an algorithm to maximise interoperability determines interoperability improvements (step-by-step) up until maximum interoperability is achieved (i.e. maximum interoperability coverage). Study about the interoperability system “in motion” was not conducted, that is, looking at an interoperability scenario while new systems enter (or leave) the environment.

Now, this might be simple to do using what was already developed in the context of the thesis as follows: establishing an interoperability scenario, maximise its interoperability using the algorithms proposed in here, and then study behaviour while adding more and more systems to the interoperability scenario. As said, this study was not done yet as it was not critical to judge the veracity of the hypothesis; however, this might be important to understand the way that the interoperability approaches behave in more dynamic/live interoperability scenario that's close to real-life situations. Also, there are some patterns, which were already identified at the simulation analyses of P2P Interoperability, but that were not further explored just yet. Particularly, considering a simulation view of a growing percentage of modifiable systems in the scenario (with fixed percentage of exporters, importers and number of formats), one can see that the Number of Interoperability Artefacts (NIA) has a high speed of increase up to some point where speed of increase slows down.

The reason for this, even if not trivial to perceive right away, is logical: the P2P algorithm always tries to maximize the number of interoperable systems, for each NIA, by using the more prominent format on the environment. The issue is that when there are not many systems to modify, more changes on the modifiable systems could be needed, in order to adapt to the non-modifiable systems. In the cases close to a full modifiable environment there are less non-modifiable, so the environment will tend faster to a more prominent format, thus requiring a lower number of interoperability artefacts. Note however that this does not imply anything on the interoperability coverage, as it always increases with the increase of modifiable systems. Now, the exact point(s) where such inflection happens – from high speed of increase to slower speed of increase – was not identified. And thus, it is not comprehended what such point-of-change would exactly mean in respect to the behaviour of the Number of Interoperability Artefacts functions. This might highlight some outstanding property of interoperability and that could lead to some additional enlightenment of interoperability-as-a-whole. Such study is still to be performed.

Another interesting thing to do but was not on this work is the determination of the Interoperability Formula for both the Point-to-Point Interoperability and Hub-and-Spoke Interoperability approaches. Such formula would allow relating the Number of Interoperability Artefacts and the Interoperability Coverage with the parameters used to describe the scenarios. Some work was done that can contribute to such formulas, such as the draw of the conclusion that all parameters are independent and the determination of the coefficients that relate the NIA obtained. The deduction and validation of such formulas presents an interesting theme for future research activities.

As concluded in the Evaluation chapter, the NIA required by the H&S Interoperability approach to maximise the interoperability within a scenario is lesser than the amount required by the P2P Interoperability approach for scenarios with a considerable dimension. Such behaviour is related with the use of multiple IFs in a row to enable interoperations between systems, forming interoperability compositions. The reuse of IFs by the mediator increases, as more different data formats exist within the scenario, being used as “paths” between Readers and Writers in the mediator. Although the fact that the use of interoperability compositions greatly improves the amount of interoperable systems, it can potentially have some impact on the data exchanged. Two causes were identified as possible threats to the integrity of the exchanged data.

The first threat is related with the human proneness to make mistakes, making that the development of any interoperability artefact error prone. The higher is the amount of interoperability artefacts that an interoperation depends on, the greater is the probability of data being lost or badly handled. The H&S approach is more sensible to this threat than the P2P approach since the data flow depends on the adaptors in mediator and in the set of IFs used to convert data between readers and writers.

The second threat is related with the data formats used by each interoperability composition. This dependence is due to the fact that one or more data formats of an interoperability composition cannot support all the data being transfer. For instance, if the data to be transferred has temperature and atmospheric pressure values and is subject to an interoperability composition that uses a data format only capable of express temperature values, then the atmospheric pressure values will be lost. This situation assumes a critical situation in the limit case where a data format used by the interoperability composition is not in the same application domain of the data format used by the source system, which can result in the corruption of transferred data.

In order to study and quantify the impact of these threats in an interoperability scenario, the measurement of the real data transferred is needed. An initial work was performed in this direction in (Maló, Mateus, Almeida, & Teixeira, 2013) (Mateus, 2012) (Mateus, Maló, Almeida, & Teixeira, 2011), where is presented a methodology to measure the data transfer between two data formats specified in an instance of an Interoperability Function. However, work contributes on an early stage to the required study since it is very limited, as it does not consider, e.g. the semantics of the used data formats.

As a last observation, there is one additional study could add even more value to this work: the study of some use-cases of the Plug’n’Play Interoperability (PnI) solution. These use-cases could be the study of how to implement the PnI solution in some reference technologies and the analysis of impact of such implementations in the performance of these reference technologies. Use-case could especially focus on Internet-of-Things and Enterprise Systems as two main domains of PnI application.

Considering for example an Internet-of-Things domain of application, the PnI solution should be included on a middleware that could exist to facilitate data connectivity of information feeds (e.g. Wireless Sensor Networks) to applications (data consumers). The trend on IoT deployments consists of lots differentiated data feeds coming from geographically dispersed heterogeneous Wireless Sensor Networks and that need to be made available to many and different applications. The multi-layer architectural model proposed by (Maló, Almeida, Melo, Kalaboukas, & Cousin, Self-Organised Middleware Architecture for the Internet-of-Things, 2013) suggests the idea of a Data Distribution MiddleWare (DDMW) to mediate data delivery in these IoT environments.

Considering Enterprise (applications and software) System the PnI solution might be implemented at the level of an ESB (Enterprise Service Bus). Somewhat similar to IoT data distribution middlewares, the ESB is also an integration approach that provides keystones for loosely coupled and highly distributed integrated systems. Formally, the ESB is a standards-based integration platform that combines messaging, web services, data transformation and intelligent routing to reliably connect and coordinate the interaction of significant numbers of diverse application across enterprises systems (Chappell, 2004) using the service-oriented paradigm. And as such, because the ESB already provides data integration tools and technologies, it could straightaway take-up the PnI solution to make fully interoperable many data-heterogeneous enterprise systems applications and software.

This study was not performed because it was not critical to validate the Hub-and-Spoke Interoperability approach and due to the fact that this dissertation is already a bit extensive without such study.

Bibliography

AirlineRouteMaps.com. (2011, February 5). *Delta Air Lines USA route network from Atlanta hub*. Retrieved August 15, 2013, from AirlineRouteMaps.com: http://www.airlineroutemaps.com/USA/Delta_Airlines_ATL.shtml

Airports Council International. (2013). *2012 World Airport Traffic and Rankings.* Montréal: Airports Council International.

Almeida, B. (2009). *Integração de Ambientes Industriais Colaborativos através da Visualização Orientada pelos Modelos.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica. Lisboa: Unpublished.

Anderson, C., & Andersson, M. P. (2007). *The long tail.* Stockholm: Bonnier fakta.

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - The Story So Far. (A. Sheth, Ed.) *International Journal on Semantic Web and Information Systems (IJSWIS), 5*(3), 1-22.

Bonnefoy, P., de Neufville, R., Asce, L., & Hansmann, J. (2008). Evolution and development of multi airport systems: a worldwide perspective. *Journal of Transportation Engineering, 136*(11), 8.

Bonsor, K. (2001, June 14). *How Airlines Work.* Retrieved August 14, 2013, from HowStuffWorks: http://science.howstuffworks.com/transport/flight/modern/airline3.htm

Bouquet, A. (2013, July 17). *History and Operations of Delta Airlines' Scissors Hub at Tokyo's Narita International Airport (NRT)* . Retrieved August 29, 2013, from The Talkative Man: http://www.talkativeman.com/delta-airlines-scissors-hub-at-tokyo-narita-international-airport-nrt/

Bourrières, J.-P. (2006). The INTEROP Network of Excellence. *Interoperability of Enterprise Software and Applications* (pp. 455-457). Springer.

Braden, B., & Hagan, P. (1989). *A Dream Takes Flight: Hartsfield Atlanta International Airport and Aviation in Atlanta.* Athens, Georgia, USA: University of Georgia Press.

Chappell, D. A. (2004). *Enterprise Service Bus.* Sebastopol, California, USA: O'Reilly Media.

Chen, D., & Doumeingts, G. (2003). European initiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap. *Annual Reviews in Control, 27*(2), 153-162.

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

Chituc, C.-M., Toscano, C., & Azevedo, A. (2008). Interoperability in Collaborative Networks: Independent and industry-specific initiatives - The case of the footwear industry. *Computers in Industry, 59*(7), 741-757.

Commodore-Amiga, Inc. (1991). *Amiga Hardware Reference Manual* (3rd Edition ed.). Boston: Addison-Wesley.

CONNECT. (2009). CONNECT: Emergent Connectors for Eternal Software Intensive Networked Systems. *FET'09 - The European Future Technologies Conference and Exhibition (2009)* (p. 2). Paris-Rocquencourt: INRIA.

Costa, R. (2007). *A Framework to support Interoperability for Collaborative Business Processes in e-Procurement.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica. Lisboa: Unpublished.

de Neufville, R., & Odoni, A. (2013). *Airport Systems: Planning, Design, and Management.* New York: McGraw-Hill Professional.

de Neufville, Richard. (1995, June). Management of multi-airport systems: A development strategy. *Journal of Air Transport Management, 2*(2), 99-110.

Delgado, M. (2008). *Harmonisation of STEP and MDA conceptual models using Model Morphisms.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica. Lisboa: Unpublished.

Delta Air Lines. (2013, July). *Press Kit - Stats & Facts*. Retrieved August 25, 2013, from Delta Air Lines Newsroom: http://news.delta.com/index.php?s=18&cat=47

Doumeingts, G., & Chen, D. (2003). Interoperability of enterprise applications and software - An European IST Thematic Network Project: IDEAS. *e2003 eChallenges Conference*, (pp. 22-24). Bologna.

EC. (2006). *Connected Health - Quality and Safety for European Citizens.* Commission of the European Communities, Information Society & Media DG. Luxembourg: Office of Official Publications of the European Communities.

Eco, U. (1977). *Come si fa una tesi di laurea.* Milano, Milano, Italia: Bompiani.

Elvesæter, B., Hahn, A., Berre, A.-J., & Neple, T. (2006). Towards an interoperability framework for model-driven development of software systems. *Interoperability of Enterprise Software and Applications* (pp. 409–420). Springer.

Eriksson, A., Feizabadi, A. R., & Zamani, B. (2001). *Integration of Heterogeneous Systems - A Component-Based Approach.* School of Economics and Commercial Law at Gothenburg University, Department of Informatics. Gothenburg: Gothenburg University.

ETSI. (2008). *Achieving Technical Interoperability - the ETSI Approach.* European Telecommunications Standards Institute. Sophia Antipolis: European Telecommunications Standards Institute.

European Commission. (2003). *Information Society Technologies: 2003-2004 Workprogramme.* European Commission, Information Society Technologies. Brussels: European Commission.

Fasone, V., & Maggiore, P. (2013). Multi-airport system and territorial competitiveness: evidence from an Italian case study. In A. S. Bergantino, F. Carlucci, A. Cirà, E. Marcucci, & E. Musso, *I sistemi di trasporto nell'area del Mediterraneo: infrastrutture e competitività* (p. 368). Milano, Italy: FrancoAngeli Edizioni.

FCT-UNL. (2001, July 18). Especialidades e disciplinas afins do Departamento de Engenharia Electrotécnica em que se confere o grau de doutor. *Diário da República - II Série, 165*, pp. 12021-12022.

FCT-UNL. (2012, Oct). Despacho n.o 13109/2012 - Regulamento da Avaliação do Desempenho dos Docentes. *Diário da República, 2*(193), pp. 33500-33511.

Ford, T. C., Colombi, J. M., Graham, S. R., & Jacques, D. R. (2007). A Survey on Interoperability Measurement. In *Adapting C2 to the 21st Century.* Newport, RI, USA: Department of Defense - Office of the Assistant Secretary of Defense Networks and Information Integration.

Friess, P. (2012). Towards Dynamism and Self-Sustainability. In I. -I.-o.-T. Cluster, *The Internet of Things 2012 - New Horizons* (pp. 12-21). Halifax, UK: Platinum.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software* (1st Edition ed.). Upper Saddle River, New Jersey, USA: Addison-Wesley Professional.

Georgouleas, K., Kalaboukas, K., Otero, M. G., & Maló, P. (2012). A Hierarchical Network Architecture for Intelligent Manufacturing and Logistics. *Embedded World Conference 2012*, (pp. 1-11). Nuremberg.

Gol. (2012, October 26). *Gol começará a voar dia 15 de dezembro para EUA.* Retrieved August 14, 2013, from Veja: http://veja.abril.com.br/noticia/economia/gol-comecara-a-voar-dia-15-de-dezembro-para-eua

Google. (2013). *Google Maps*. Retrieved 10 2013, from maps.google.com: http://maps.google.com

Haik, Y., & Shahin, T. (2003). *Engineering Design Process.* Stamford, CT, USA: Cengage Learning.

Hoogerwerf, J. J. (2010, December 13). Roots: From Crop Duster to Airline; The Origins of Delta Air Lines to World War II . 217. Auburn, Alabama, USA: Faculty of Auburn University.

Horn, R. A. (1989). The Hadamard Product. In C. R. Johnson (Ed.), *Proceeding Symposia of Applied Mathematics* (Vol. 40, pp. 87-170). USA: American Methematical Society.

IEC TC 65. (2003). *Device Profile Guideline.* International Electrotechnical Commission, TC65: Industrial Process Measurement and Control. IEC.

IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology.* Institute of Electrical and Electronics Engineers, C/S2ESC - Software & Systems Engineering Standards Committee. New York: Institute of Electrical and Electronics Engineers.

IEEE. (1991). *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries.* Piscataway, NJ, USA: Institute of Electrical & Electronics Engineers Press.

IERC. (2013). *IoT Semantic Interoperability: Research Challenges, Best Practices, Solutions and Next Steps.* European Research Cluster on the Internet of Things, IERC Activity Chain 4 - Interoperability. Brussels: European Commission.

Indo-Asian News Service. (2010, November 2). *Air India now flies non-stop to Canada*. Retrieved August 28, 2013, from NDTV - New Delhi Television: http://www.ndtv.com/article/world/air-india-now-flies-non-stop-to-canada-63942

Iseki, H., Taylor, B., & Miller, M. (2006). *The Effects of Out-of-Vehicle Time on Travel Behavior: Implications for Transit Transfers (Deliverable #1).* University of California, Institute of Transportation Studies. Sacramento: California Department of Transportation.

ISO/IEC. (2004). *ISO/IEC Guide 2:2004 Standardization and related activities -- General vocabulary.* International Organization for Standardization, Technical Management Board Groups. Geneva: ISO.

ISO14258. (1998). *ISO 14258:1998 Industrial automation systems -- Concepts and rules for enterprise models.* International Organization for Standardization, ISO Technical Committee 184, Industrial automation systems and integration, Subcommittee 5, Architecture and communications. Geneva: International Organization for Standardization.

Issarny, V., & Bennaceur, A. (2013). Composing Distributed Systems: Overcoming the Interoperability Challenge. In E. Giachino, R. Hähnle, F. de Boer, & M. Bonsangue, *Formal Methods for Components and Objects* (Vol. 7866, p. 29). Heidelberg: Springer Berlin Heidelberg.

Jardim-Goncalves, R., Popplewell, K., & Grilo, A. (2012, Sep 1). Sustainable Interoperability: The future of Internet based industrial enterprises. *Computers in Industry, 63*(8), 731-738.

Jardim-Gonçalves, R., Silva, H., Vital, M., Sousa, S.-G. A., & Pamiés-Teixeira, J. (1997). Implementation of computer integrated manufacturing systems using SIP: CIM case studies using a STEP approach. *International Journal of Computer Integrated Manufacturing, 10*(1-4), 172-180.

Jardim-Gonçalves, R., Sousa, P. C., Pimentao, J. B., & Steiger-Garcao, A. (1999). Integrating manufacturing systems using ISO 10303 (STEP): An overview of UNINOVA projects. *International Journal of Computer Applications in Technology, 12*(1), 39-45.

Jardim-Gonçalves, R., Tavares, R., Grilo, A., & Steiger-Garção, A. (2000). A framework for integration of supply chain management applications based on the IAI/IFC model. *17th IAARC/CIB/IEEE/IFAC/IFR International Symposium on Automation and Robotics in Construction (ISARC)*, (pp. 745-750). Taipei.

Kishore, J., Vasundhra, S., & Anand, T. (2011, Oct 9). Formulation of a Research Question. (A. Prakash, Ed.) *Indian Journal of Medical Specialities, 2*(2), 184-188.

Kothari, C. R. (2004). *Research Methodology: Methods and Techniques.* New Delhi, Delhi, India: New Age International (P) Limited.

Kubicek, H., & Cimander, R. (2011). Layers of Interoperability. In K. Herbert, R. Cimander, & H. J. Scholl, *Organizational Interoperability in E-Government* (pp. 85-96). Heidelberg, Berlin, Germany: Springer Berlin Heidelberg.

Lancioni, R. A., Smith, M. F., & Oliva, T. A. (2000, Jan). The role of the internet in supply chain management. *Industrial Marketing Management, 29*(1), 45-56.

Lewis, D., & Newton, W. (1979). *Delta: The History of an Airline* (First Edition ed.). Athens, Georgia, USA: University of Georgia Press.

Liebich, T. (2010, Jan 30). *IFC2x3 Implementation Guidance*. Retrieved Jul 3, 2013, from buildingSMART Tech: http://www.buildingsmart-tech.org/implementation/ifc-implementation

Liebich, T., & Wix, J. (1999). Highlights of the development process of industry foundation classes. In M. Lacasse, & D. Vanier (Eds.), *Durability of Building Materials and Components* (Vol. 8, pp. 2758-2775). Ottawa, Canada: Institute for Research in Construction.

Maassen, P., & Stensaker, B. (2011, June). The Knowledge Triangle - European higher education policy logics and policy implications. *Higher Education, 61*(6), 757-769.

Maia, E. (2012, November 7). *Companhia aérea brasileira passa a voar entre Rio e Miami.* Retrieved August 6, 2013, from OGlobo: http://oglobo.globo.com/boa-viagem/companhia-aerea-brasileira-passa-voar-entre-rio-miami-6659125

Malhotra, N. K., Kim, S. S., & Agarwal, J. (2004). Internet Users' Information Privacy Concerns (IUIPC): The Construct, the Scale, and a Causal Model. *Information Systems Research, 15*(4), 336-355.

Maló, P., & Steiger-Garção, A. (2007). *Parecer sobre o CONVERIS para a gestão da produção científica na UNL.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica. Caparica: Unpublished.

Maló, P., & Steiger-Garção, A. (2007). *Repositório Institucional @ FCT-UNL.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica. Caparica: Unpublished.

Maló, P., Almeida, B., Melo, R., Kalaboukas, K., & Cousin, P. (2013). Self-Organised Middleware Architecture for the Internet-of-Things. *Proceedings of the 2013 IEEE International Conference on Internet of Things (iThings2013)* (pp. 1-7). Beijing: IEEE Publishing.

Maló, P., Almeida, B., Melo, R., Kalaboukas, K., & Cousin, P. (2013). Self-Organised Middleware Architecture for the Internet-of-Things. *The 2013 IEEE International Conference on Internet of Things (iThings2013).* Beijing: Institute of Electrical and Electronics Engineers.

Maló, P., Jardim-Gonçalves, R., Borràs, M., & Steiger-Garção, A. (2002). A platform for automated interoperability inside a networked enterprise: Facilitating e-Business between Manufacturers, Suppliers, Retailers and Shops. *PDT Europe 2002: Product Data Technology Europe 2002 11th Symposium - Interoperability inside the Networked Enterprise*, (pp. 1-8). Turin.

Maló, P., Mateus, M., Almeida, B., & Teixeira, T. (2013). Measuring Data Transfer in Heterogeneous IoT Environments. *Proceedings of the 2013 IEEE International Conference on Internet of Things* (pp. 1-9). Beijing: IEEE Publishing .

Maló, P., Teixeira, T., Almeida, B., & Mateus, M. (2013). Interoperability Repository System for IoT. In *Proceedings of the Symposia on Frontiers of Internet of Things (SymIoT)* (pp. 1-9). Beijing, China: IEEE Publishing.

MARICOPA. (2009, December 8). *Transportation, Distribution & Logistics.* Retrieved August 13, 2013, from Maricopa County Community College District - Center for Workforce Development: http://www.maricopa.edu/work/pdfs/SummaryTransportation.pdf

Martin, J. (1983). *Managing the Data-Base Environment* (2nd Edition ed.). California: Prentice-Hall.

Mateus, M. (2012). *Measuring Data Transfer in Heterogeneous IoT Environments.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica e de Computadores. Caparica: Universidade Nova de Lisboa.

Mateus, M., Maló, P., Almeida, B., & Teixeira, T. (2011). Towards measuring information interoperability based on model transformations. *Proceedings of the 6th Iberian Conference on Information Systems and Technologies (CISTI'2011)* (pp. 1-4). Chaves: IEEE Publishing .

McLaughlin, B. (2002). *Java & XML data binding.* Sebastopol: O'Reilly Media.

Melo, R. (2012). *P2P Service Exposer Architecture.* Faculdade de Ciências e Tecnologia da Universisade Nova de Lisboa, Departamento de Engenharia Electrotécnica e de Computadores. Caparica: Universidade Nova de Lisboa.

Mooney, C. (1997). *Monte Carlo Simulation.* SAGE Publications.

Morris, E., Levine, L., Meyers, C., Place, P., & Plakosh, D. (2004). *System of Systems Interoperability (SOSI).* Carnegie Mellon University, Software Engineering Institute. Hanscom: SEI Joint Program Office.

NTIS. (1988). *Safe Skies for Tomorrow: Aviation Safety in a Competitive Environment.* United States Congress, Office of Technology Assessment. Washington: U.S. Government Printing Office.

Nuñez, M., Jardim-Gonçalves, R., Togores, A., Garcia, R., Panetto, H., Lauro, L., et al. (2006). *Patent No. ISO10303-236.* Switzerland.

Panetto, H. (2007). Towards a classification framework for interoperability of enterprise application. *International Journal of Computer Integrated Manufacturing, 20*(8), 727-740.

Pereira, H. (2012). *Interoperability on Low Power Devices.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica e de Computadores. Caparica: Universidade Nova de Lisboa.

Pratt, M. J. (2001). Introduction to ISO 10303 - the STEP standard for product data exchange. *Journal of Computing and Information Science in Engineering, 1*(1), 102-103.

Rajasekar, S., Philominathan, P., & Chinnathambi, V. (2013). *Research Methodology.* Essay, Bharathidasan University, School of Physics, Tamilnadu.

Reiche, R., Lehmann, R. J., Fritz, M., & Schiefer, G. (2011). Business process support using RFID-based information services in the distribution of fresh fruits and vegetables. *Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft*, *181*, pp. 161-164. Oppenheim.

Ruggaber, R. (2006). Athena-Advanced technologies for Interoperability of heterogeneous enterprise networks and their applications. *Interoperability of Enterprise Software and Applications* (pp. 460-463). Springer London.

Sousa, P., Jardim-Gonçalves, R., Pimentão, J. P., Pamiés-Teixeira, J., & Steiger-Garção, A. (1999). Seeking intelligent product development - an integrator environment based on STEP. *Journal of Intelligent Manufacturing, 10*(3-4), 313-321.

Sousa, P., Pimentão, J. P., Jardim-Gonçalves, R., & Steiger-Garção, A. (1999). ISO 10303 Application interfaces supported by Genesis environment - The funStep ESPRIT project experience. *3rd IMACS International Multiconference on Circuits, Systems, Communications and Computers (CSCC'99/IEEE)*, (pp. 1-5). Athens.

Sundmaeker, H., & Kovacikova, T. (2010). CuteLoop-An Approach for Networked Devices Enabled Intelligence. *Fifth International Conference on Software Engineering Advances (ICSEA)* (pp. 205-212). Nice: IEEE Publishing.

Sundmaeker, H., Scholze, S., Stokic, D., & Faltus, S. (2008). Intelligent Networked Devices for Enabling Proactive Collaboration with Customers. *14th International Conference on Concurrent Enterprising* (pp. 1-8). Nottingham: Centre for Concurrent Enterprises.

Tayal, S. (2013). Engineering Design Process. *International Journal of Computer Science and Communication Engineering*, 1-5.

Teixeira, T. (2012). *Organising Interoperability Information on Highly Dynamic and Heterogeneous Environments.* Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Electrotécnica e de Computadores. Caparica: Universidade Nova de Lisboa.

Teixeira, T., Maló, P., Almeida, B., & Mateus, M. (2011). Towards an Interoperability Management System. *6th Iberian Conference on Information Systems and Technologies (CISTI'2011). 1*, pp. 1-4. Chaves: IEEE Publishing.

The Times of India. (2013, April 20). *Jet Airways eyes global footprint via Abu Dhabi*. (Bennett, Coleman & Co) Retrieved August 16, 2013, from The Times of India: http://www.timesofindia.com/business/india-business/Jet-Airways-eyes-global-footprint-via-Abu-Dhabi/articleshow/19642581.cms

UK Cabinet Office. (2004). *e-Government Interoperability Framework.* London: Crown.

UKOLN. (2006). *Interoperability Focus: About*. Retrieved Jul 15, 2013, from UK Office for Library Networking: http://www.ukoln.ac.uk/interop-focus/about/

UNH-IOL. (2013, Jul 25). *The History of the UNH-IOL.* Retrieved Jul 25, 2013, from University of New Hampshire - InterOperability Laboratory: http://www.iol.unh.edu/general/history.php

Van der Ploeg, D. J., Renting, H., Brunori, G., Knickel, K., Mannion, J., Marsden, T., et al. (2000). Rural development: from practices and policies towards theory. *Sociologia ruralis, 40*(3), 391-408.

W3C SWEO. (2013, May 13). *LinkingOpenData*. Retrieved Sep 6, 2013, from W3C Communities Wiki: http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

Wensveen, J. G. (2011). *Air Transportation: A Management Perspective - 7th ed.* (7th Edition ed.). Farnham, Surrey, England: Ashgate Publishing Limited.

Wikipedia. (2012, Oct 5). *Interoperability*. Retrieved Oct 23, 2012, from Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Interoperability

Wikipedia. (2013, May 16). *Loose coupling*. Retrieved August 23, 2013, from Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/w/index.php?title=Loose_coupling

Wikipedia. (2013, July 25). *Technical Standard*. Retrieved Aug 2, 2013, from Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Technical_standard

Wikipedia contributors. (2013, May 16). *Loose coupling*. Retrieved August 23, 2013, from Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/w/index.php?title=Loose_coupling

Appendix A:
MATLAB Simulation

In this appendix it is presented the methodology used to simulate the P2P and H&S Interoperability approaches and the MATLAB code used to perform these simulations, as well as the MATLAB code used to produce the graphics presented in this thesis. The distributed approach used to execute the simulations is also presented along with the corresponding MATLAB code.

Simulation Methodology

In Image 1 is presented a representation of the methodology used in this thesis to simulate the P2P Interoperability and the Hub-and-Spoke Interoperability approaches. The first stage of the methodology is the generation of the scenarios that will be used in the simulations. This operation is performed by the function *simBuilder*, generation scenarios based on the parameters used as input: the percentage of modifiable systems, the percentage of systems in the scenario that have export and/or import capabilities, the maximum percentage of systems that can support an equal data format and the number of systems that composed the scenario to be generated.

The *simBuilder* function is able to generate several scenarios by varying the number o systems that compose the scenario. The number of generated scenarios depends on the difference between the minimum and the maximum amount of systems specified and on the step used to increment the number of system from one scenario to another. All the generated scenarios are storage in an object of the class *Simulations*. The purpose of this class is to provide structures to store all data related with the simulations of the approaches, i.e. all scenarios used in the simulations and its results. This class also provides methods to access the stored data and to execute simulations over the stored scenarios.

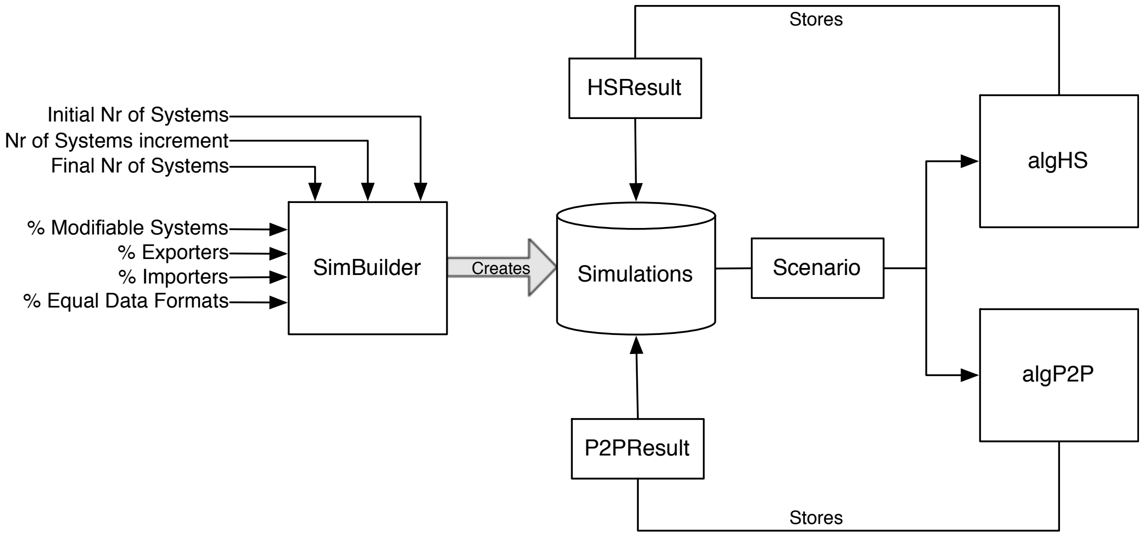


Image 1: Overview of the methodology used to simulate the P2P and H&S approaches

Each instance of Simulations has the capability to run one algorithm (*algHS* corresponding to the H&S approach or *algP2P* to apply the P2P approach) over one scenario, which information is represented in a *Scenario* data structure. The results of the execution of an algorithm are stored in a specific data structure: *HSResult* for the case *algHS* and *P2PResult* when the execute algorithm is the *algP2P*. These data structures are then stored into the *Simulations* instance, keeping the reference of the scenario that it corresponds to.

The code corresponding to the *simBuilder* function; *Simulations* class; *algHS* and *algP2P* algorithms; and Scenario, *HSResult* and *P2PResult* data structures are presented below.

simBuilder

This function is the responsible for the coordination of the whole process of execution of a simulation. It is able to generate representations of interoperability scenarios through random definition of the data formats used and random distribution of data formats by the systems that compose the scenario. This random distribution is performed assuming that, in this stage, each system with export capabilities support can only export using one data format, as well as a system that with import capabilities can only import from one data format. The determination of which systems are modifiable is also random, depending on the *modifRate* parameter to specify the amount of modifiable systems in the scenario. After the generation of the scenarios, they are added to an instance of the *Simulations* class and methods of this class are evocated to run the algorithms of the P2P and H&S approaches (methods *runP2P* and *runHS*).

```
function [sim] = simBuilder( initSysCount, stepCount, finalSysCount, formatsRate, modifRate, exportRate, importRate)
%simBuilder Creates a Simulation structure with scenarios generated based
%on the input arguments
%   Produces scenarios where the number of system ranges from initSysCount
%   to finalSysCount, with stepCount as step;
%   formatsRate - corresponds to the maximum percentage of systems with the
%   same data format;
%   modifRate - represents the portion of systems that are modifiable;
%   exportRate - indicates the portion of systems that have export
%   capabilities;
%   importRate - indicates the portion of systems that have import
%   capabilities.
%

%% validation of the export/import rates
if ((exportRate + importRate) <1) || exportRate==0 || importRate==0 || exportRate > 1 || importRate > 1
    err = MException('ResultChk:OutOfRange','Invalid occupation rate.');
    throw(err);
end
sim = Simulations;
for i=initSysCount:stepCount:finalSysCount
    nExp = ceil(exportRate * i);
    nImp = ceil(importRate * i);
    nIntf = nExp + nImp;
    allFormats = zeros(1,nIntf,'single');
    nFormats = ceil(nIntf * formatsRate);
    if nFormats==0
        nFormats = 1;
    end
    allFormats(1,1:nFormats)=1;
    remainingFormats = 2:nIntf; %? - why 2 -> different formats(different from the 1st data format)
    difForms=size(remainingFormats,2);
    formatCounter = zeros(1,difForms,'single');
    for j=nFormats+1:nIntf
        fIndex=ceil(rand*difForms);
        allFormats(1,j) = remainingFormats(1,fIndex);
        formatCounter(1,fIndex) = formatCounter(1,fIndex)+1;
        if formatCounter(1,fIndex) == nFormats
            remainingFormats(1,fIndex)=0;
            remainingFormats=remainingFormats(remainingFormats~=0);
            difForms=difForms-1;
            formatCounter=formatCounter(formatCounter~=nFormats);
        end
    end
    %% Defines the importers and exporters
    Matrix = zeros(i,nFormats,2,'single');
    formatsAvailable = allFormats;
    for j=1:nExp
        index = ceil(rand * i);
        while(sum(Matrix(index,:,1))~=0)
            if index < i
                index = index + 1;
            else
                index = 1;
            end
        end
        formIndex = ceil(rand*size(formatsAvailable,2));
        Matrix(index,formatsAvailable(1,formIndex),1) = 1;
        formatsAvailable(1,formIndex) = 0;
        formatsAvailable = formatsAvailable(formatsAvailable~=0);
    end
    sysExp = sum(Matrix(:,:,1),2);
    impIndex = find(sysExp==0);
    for j=1:size(impIndex,1)
        formIndex = ceil(rand*size(formatsAvailable,2));
        Matrix(impIndex(j,1),formatsAvailable(1,formIndex),2) = 1;
        formatsAvailable(1,formIndex) = 0;
        formatsAvailable = formatsAvailable(formatsAvailable~=0);
    end
    % distribute remaining data formats
    if ~isempty(formatsAvailable)
        nF=size(formatsAvailable,2);
        for j=1:nF
            while(sum(Matrix(index,:,2))~=0)
                if index < i
```

```
                index = index + 1;
            else
                index = 1;
            end
        end
        formIndex = ceil(rand*size(formatsAvailable,2));
        Matrix(index,formatsAvailable(1,formIndex),2) = 1;
        formatsAvailable(1,formIndex) = 0;
        formatsAvailable = formatsAvailable(formatsAvailable~=0);
    end
end
formatsAvailable = allFormats;
nImpf = nImp - (i - nExp + j); %determines the amount of importers left to be placed
% deal with the remaining importers (needed???)
for j=1:nImpf
    disp('#NEEDED!!!!!!')
    index = ceil(rand * i);
    while(sum(Matrix(index,:,2))~=0)
        if index < i
            index = index + 1;
        else
            index = 1;
        end
    end
    formIndex = ceil(rand*size(formatsAvailable,2));
    Matrix(index,formatsAvailable(1,formIndex),2) = 1;
    formatsAvailable(1,formIndex) = 0;
    formatsAvailable = formatsAvailable(formatsAvailable~=0);
end
%% Determinates the number of modifiable systems and generates the modifiable array
nModif = ceil(i * modifRate);
if i-nModif >= nModif
    maxIte = nModif;
    flag=1; %mark as modifiable
    nodesModif=zeros(i,1,'single');
else if i-nModif < nModif
        maxIte = i-nModif;
        flag=0; %mark as non-modifiable
        nodesModif=ones(i,1,'single');
    end
end
for j=1:maxIte
    index = ceil(rand * i);
    while(nodesModif(index,1)==flag)
        if index < i
            index = index + 1;
        else
            index = 1;
        end
    end
    nodesModif(index,1) = flag;
end
scen = Scenario(Matrix,nodesModif);
sim = addScenario(sim,scen);
end
%% Run the algorithm with the P2P algorithm
% sim = runP2P (sim);
%% Run the algorithm with the H&S algorithm
sim = runHS(sim);
end
```

Simulations

Class designed to store the scenario representations to be used (property *scenario*) in the simulations as well as its results (properties *P2PResults* and *HSResults*). This class also provides the methods to execute the algorithm that simulates the P2P approach (*runP2P*) and the algorithm that simulates the H&S approach (*runHS*). It is also provided methods to access the data of the simulations' results (*getP2Pdata* and *getHSdata*) and to graphically represent the simulations' results in graphics (plotP2P and *plotHS*).

```
classdef Simulations
    %Simulations Class used to structure the data required for a scenario's
    %simulation and to store its results. It also as the methods to execute
    %the algorithms to maximum interoperability

    properties
        nScenarios=0
        scenarios
        P2PResults
```

```
    HSResults
end

methods
    %% Add a new scenario to be applied the algorithms to maximum interoperability
    function obj = addScenario(obj,newScenario)
        if obj.nScenarios == 0
            scenariosArray(1,1) = newScenario;
        else
            scenariosArray = [obj.scenarios,newScenario];
        end
        obj.nScenarios = obj.nScenarios+1;
        obj.scenarios=scenariosArray;
    end

    %% Run the Algorithm to Maximum Interoperability using the P2P Interoperability Approach
    function obj = runP2P(obj)
        if obj.nScenarios ~= 0
            results(1,obj.nScenarios)=P2PResult;
            scens = obj.scenarios;
            nScens = obj.nScenarios;
            for i=1:nScens
                [effort,coverage,absCover,finalExp,finalImp]=algP2P(scens(1,i).nodesMatrix(:,:,1),scens(1,i).nodesMatrix(:,:,2),scens(1,i).nodesModif);
                [nSys,nForm] = size(finalExp);
                finalMatrix=zeros(nSys,nForm,2,'single');
                finalMatrix(:,:,1) = finalExp;
                finalMatrix(:,:,2) = finalImp;
                results(1,i) = saveResult( results(1,i),effort,coverage,absCover,finalMatrix,scens(1,i).nodesModif,size(scens(1,i).nodesMatrix,1));
            end
            if size(obj.P2PResults)>0
                obj.P2PResults=[obj.P2PResults;results];
            else
                obj.P2PResults=results;
            end
        end
    end



    %% Run the Algorithm to Maximum Interoperability using the Hub-and-Spoke Interoperability Approach
    function obj = runHS(obj)
        if obj.nScenarios ~= 0
            results(1,obj.nScenarios)=HSResult;
            for i=1:obj.nScenarios
                [ NIA,coverage,absCoverage,finalExp,finalImp,IF ] = algHS( obj.scenarios(1,i).nodesMatrix(:,:,1), obj.scenarios(1,i).nodesMatrix(:,:,2));
                [nSys,nForm] = size(finalExp);
                finalMatrix=zeros(nSys,nForm,2,'single');
                finalMatrix(:,:,1) = finalExp;
                finalMatrix(:,:,2) = finalImp;
                results(1,i) = saveResult(results(1,i),NIA,coverage,absCoverage,finalMatrix,IF,size(obj.scenarios(1,i).nodesMatrix,1),nForm);
            end
            if size(obj.HSResults)>0
                obj.HSResults=[obj.HSResults;results];
            else
                obj.HSResults=results;
            end
        end
    end



    %% Returns the data obtained from the simulations of the P2P approach
    function [Effort,nNodes,Coverage,AbsCover] = getP2PData(obj,minSys,maxSys)
        [sims,ite]=size(obj.P2PResults);
        nNodes=zeros(1,ite);
        for i=1:ite
            nNodes(1,i)=obj.P2PResults(1,i).nNodes;
        end
        if nargin ==3
            minIndex = find(nNodes==minSys,1);
            if isempty(minIndex)
                minIndex=1;
            end
            maxIndex = find(nNodes==maxSys,1);
            if isempty(maxIndex)
                maxIndex=ite;
            end
        else
            minIndex=1;
            maxIndex=ite;
        end
        nSys = (maxIndex-minIndex) + 1;
        nNodes=zeros(1,nSys);
        efforts=zeros(sims,nSys);
        coverages=zeros(sims,nSys);
        absCoverages=zeros(sims,nSys);
        meanEffort = zeros(1,nSys);
        meanCover = zeros(1,nSys);
        meanAbsCover = zeros(1,nSys);
        intEffUpper = zeros(1,nSys);
        intEffLower = zeros(1,nSys);
        intCovUpper = zeros(1,nSys);
        intCovLower = zeros(1,nSys);
        intAbsCovUpper = zeros(1,nSys);
        intAbsCovLower = zeros(1,nSys);
        index=1;
        for i=minIndex:maxIndex
```

```
        nNodes(1,index)=obj.P2PResults(1,i).nNodes;
        for j=1:sims
            efforts(j,index)=obj.P2PResults(j,i).effort;
            coverages(j,index)=obj.P2PResults(j,i).coverage;
            absCoverages(j,index)=obj.P2PResults(j,i).absCoverage;
        end
        meanEffort(1,index) = mean(efforts(:,index));
        meanCover(1,index) = mean(coverages(:,index));
        meanAbsCover(1,index) = mean(absCoverages(:,index));
        intEffUpper(1,index) = max(efforts(:,index))-meanEffort(1,index);
        intEffLower(1,index) = meanEffort(1,index) - min(efforts(:,index));
        intCovUpper(1,index) = max(coverages(:,index))-meanCover(1,index);
        intCovLower(1,index) = meanCover(1,index) - min(coverages(:,index));
        intAbsCovUpper(1,index) = max(absCoverages(:,index))-meanAbsCover(1,index);
        intAbsCovLower(1,index) = meanAbsCover(1,index) - min(absCoverages(:,index));
        index = index + 1;
    end
    Effort = [ meanEffort; intEffLower; intEffUpper ];
    Coverage = [ meanCover; intCovLower; intCovUpper ];
    AbsCover = [ meanAbsCover; intAbsCovLower; intAbsCovUpper ];
end

%% Returns the data obtained from the simulations of the H&S approach
function [Effort,nNodes,Coverage,AbsCover] = getHSData(obj,minSys,maxSys)
    [sims,ite]=size(obj.HSResults);
    nNodes=zeros(1,ite);
    for i=1:ite
        nNodes(1,i)=obj.HSResults(1,i).nNodes;
    end
    if nargin==3
        minIndex = find(nNodes==minSys,1);
        if isempty(minIndex)
            minIndex=1;
        end
        maxIndex = find(nNodes==maxSys,1);
        if isempty(maxIndex)
            maxIndex=ite;
        end
    else
        minIndex=1;
        maxIndex=ite;
    end
    nSys = (maxIndex-minIndex) + 1;
    nNodes=zeros(1,nSys);
    efforts=zeros(sims,nSys);
    coverages=zeros(sims,nSys);
    absCoverages=zeros(sims,nSys);
    meanEffort = zeros(1,nSys);
    meanCover = zeros(1,nSys);
    meanAbsCover = zeros(1,nSys);
    EffUpper = zeros(1,nSys);
    EffLower = zeros(1,nSys);
    CovUpper = zeros(1,nSys);
    CovLower = zeros(1,nSys);
    AbsCovUpper = zeros(1,nSys);
    AbsCovLower = zeros(1,nSys);
    index = 1;
    for i=minIndex:maxIndex
        nNodes(1,index)=obj.HSResults(1,i).nNodes;
        for j=1:sims
            efforts(j,index)=obj.HSResults(j,i).effort;
            coverages(j,index)=obj.HSResults(j,i).coverage;
            absCoverages(j,index)=obj.HSResults(j,i).absCoverage;
        end
        meanEffort(1,index) = mean(efforts(:,index));
        meanCover(1,index) = mean(coverages(:,index));
        meanAbsCover(1,index) = mean(absCoverages(:,index));
        EffUpper(1,index) = max(efforts(:,index)) - meanEffort(1,index);
        EffLower(1,index) = meanEffort(1,index) - min(efforts(:,index));
        CovUpper(1,index) = max(coverages(:,index))-meanCover(1,index);
        CovLower(1,index) = meanCover(1,index) - min(coverages(:,index));
        AbsCovUpper(1,index) = max(absCoverages(:,index))-meanAbsCover(1,index);
        AbsCovLower(1,index) = meanAbsCover(1,index) - min(absCoverages(:,index));
        index = index + 1;
    end
    Effort = [ meanEffort; EffLower; EffUpper ];
    Coverage = [ meanCover; CovLower; CovUpper ];
    AbsCover = [ meanAbsCover; AbsCovLower; AbsCovUpper ];
end

function fig = plotP2P(obj,simName)
    [Effort,nNodes,Coverage,AbsCover] = obj.getP2PData;
    fig=figure;
    [AX,H1,H2] = plotyy(nNodes, Effort(1,:), nNodes, Coverage(1,:),'plot');
    set(fig, 'CurrentAxes', AX(1));
    set(get(AX(1),'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
    ylim(AX(1),[0 max(Effort(1,:)+Effort(3,:))+10]);
    %% Plot NIA
    set(H1,'LineWidth',1.3,'Marker','.','DisplayName','NIA');
    hold on;
    e1 = errorbar(nNodes,Effort(1,:),Effort(2,:),Effort(3,:),'b');
    set(e1,'LineWidth',1.3)
    hAnnotation = get(e1,'Annotation');
    hLegendEntry = get(hAnnotation','LegendInformation');
    set(hLegendEntry,'IconDisplayStyle','off')
    %% Plot Coverage
```

```
        set(H2,'color','red');
        set(fig, 'CurrentAxes', AX(2));
        set(AX(2),'ycolor','r');
        set(get(AX(2),'Ylabel'),'String','Coverage','FontSize',14);
        set(get(AX(2),'Xlabel'),'String','Number of systems','FontSize',14);
        ylim(AX(1),[0 4000]);
        set(AX(1),'ytick',(0:200:4000));
        ylim(AX(2),[0 1]);
        set(AX(2),'ytick',(0:0.05:1)');
        set(H2,'LineWidth',1.3,'Marker','.','DisplayName','% to max possible');
        set(AX(1),'XTick',0:10:120,'FontSize',14);
        set(AX(2),'FontSize',14)
        hold on;
        e2 = errorbar(nNodes,Coverage(1,:),Coverage(2,:),Coverage(3,:),'r');
        set(e2,'LineWidth',1.3)
        hAnnotation = get(e2,'Annotation');
        hLegendEntry = get(hAnnotation','LegendInformation');
        set(hLegendEntry,'IconDisplayStyle','off')
        %% Plot Absolute Coverage
        hold on;
        H3=errorbar(nNodes,AbsCover(1,:),AbsCover(2,:),AbsCover(3,:),'r.--');
        set(H3,'LineWidth',1.3,'DisplayName','% to fully connected');
        grid(AX(1));
        grid on;
        title(simName,'FontSize',16,'FontWeight','bold');
        lh=legend('show','Location','Southeast');
        set(lh,'Color','white');
        set(fig, 'CurrentAxes', AX(1));
    end

    function fig=plotHS(obj,simName)
        [Effort,nNodes,Coverage,AbsCover] = obj.getHSData;
        fig=figure;
        [AX,H1,H2] = plotyy(nNodes, Effort(1,:), nNodes, Coverage(1,:),'plot');
        set(fig, 'CurrentAxes', AX(1));
        set(get(AX(1),'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
        ylim(AX(1),[0 max(Effort(1,:)+Effort(3,:))+10]);
        %% Plot NIA
        set(H1,'LineWidth',1.3,'Marker','.','DisplayName','NIA');
        hold on;
        e1 = errorbar(nNodes,Effort(1,:),Effort(2,:),Effort(3,:),'b');
        set(e1,'LineWidth',1.3)
        hAnnotation = get(e1,'Annotation');
        hLegendEntry = get(hAnnotation','LegendInformation');
        set(hLegendEntry,'IconDisplayStyle','off')
        %% Plot Coverage
        set(H2,'color','red');
        set(fig, 'CurrentAxes', AX(2));
        set(AX(2),'ycolor','r');
        set(get(AX(2),'Ylabel'),'String','Coverage','FontSize',14);
        set(get(AX(2),'Xlabel'),'String','Number of systems','FontSize',14);
        ylim(AX(1),[0 500]);
        set(AX(1),'ytick',(0:25:500));
        ylim(AX(2),[0 1]);
        set(AX(2),'ytick',(0:0.05:1)');
        set(H2,'LineWidth',1.3,'Marker','.','DisplayName','% to max possible');
        set(AX(1),'XTick',0:10:120,'FontSize',14);
        set(AX(2),'FontSize',14)
        hold on;
        e2 = errorbar(nNodes,Coverage(1,:),Coverage(2,:),Coverage(3,:),'r');
        set(e2,'LineWidth',1.3)
        hAnnotation = get(e2,'Annotation');
        hLegendEntry = get(hAnnotation','LegendInformation');
        set(hLegendEntry,'IconDisplayStyle','off')
        %% Plot Absolute Coverage
        hold on;
        H3=errorbar(nNodes,AbsCover(1,:),AbsCover(2,:),AbsCover(3,:),'r.--');
        set(H3,'LineWidth',1.3,'DisplayName','% to fully connected');
        grid(AX(1));
        grid on;
        title(simName,'FontSize',16,'FontWeight','bold');
        lh=legend('show','Location','Southeast');
        set(lh,'Color','white');
        set(fig, 'CurrentAxes', AX(1));
    end
  end

end
```

algHS

This function implements the algorithm to maximum interoperability using the Hub-and-Spoke Interoperability approach. Using the a representation of a scenario, this function determines the Number of Interoperability Artefacts required to maximise the interoperability in that scenario, as well as the Interoperability Coverage and

Interoperability Absolute Coverage of the scenario after the execution of the algorithm. This function also returns the set of IFs that maximise the interoperability in scenario represented.

```
function [ NIA,coverage,absCoverage,exportMatrix,importMatrix,IFdir ] = algHS( OrigExportMatrix,OrigImportMatrix )
%algHS H&S algorithm implementation using matrices
%   Inputs: exportMatrix - representation of the export capabilities of a specific scenario
%           importMatrix - representation of the import capabilities of a specific scenario
%
%   Outputs: NIA - number of interoperability artefacts in the scenario
%
%         coverage - relation between the number of connections in the scenario and the maximum number of connections possible for that specific scenario
%
%         absCoverage - relation between the number of connections in the scenario and the maximum number of connections given by the fully  connected
%
%         OrigExportMatrix - final state of the representation of the export capabilities in a specific scenario
%
%         OrigImportMatrix - final state of the representation of the import capabilities in a specific scenario
%
%         IFdir - matrix representing all the IF applied to the initial scenario

exportMatrix = OrigExportMatrix;
importMatrix = OrigImportMatrix;
[nExp,nFormats] = size(exportMatrix);
%% Determination of the initial NIA
Texp = sum(sum(exportMatrix));
Timp = sum(sum(importMatrix));
Nadap = Texp + Timp;          % Number of adapters in systems
expForm = sum(logical(sum(OrigExportMatrix,1))); % number of formats used to export -> number of export adapters in mediator
impForm = sum(logical(sum(OrigImportMatrix,1))); % number of formats used to import -> number of import adapters in mediator
NIA = Nadap + expForm + impForm + 1; % the one corresponds to the mediator

ident = ~eye(nExp);
formIdent = ~eye(nFormats);
modExp = single(logical(sum(exportMatrix,2)));
modExpMat = repmat(modExp,1,nExp); % exporters filter
modImp = single(logical(sum(importMatrix,2)));
modImpMat = repmat(modImp,1,nExp); % importers filter
IF = zeros(nFormats,'single');
IFdir = zeros(nFormats,'single');
allIndIF = zeros(nFormats,'single');

while true
    %% Identification of all the IF that can improve the scenario
    connectMatrix = logical(~(exportMatrix * importMatrix') & ident);
    connectMatrix = (connectMatrix & modImpMat') & modExpMat;
    possibleIF=zeros(nFormats,'single');
    for i=1:nExp % counts the number of systems that each is will connect if applied
        % performs the matching between each exporter and all the importers
        % that still cannot import information from that system
        possibleIF = possibleIF + (exportMatrix(i,:)'*sum(importMatrix(connectMatrix(i,:),:),1));
    end
    %% Application of the use of indirect IF penalty
    appliedIF = IF+allIndIF;
    targetFormatWeight = sum(appliedIF,1);
    sourceFormatWeight = sum(appliedIF,2);
    possibleIF = possibleIF -(0.0001 * (repmat(sourceFormatWeight',nFormats,1) + repmat(targetFormatWeight',1,nFormats)));
    maxVal = max(possibleIF(:));
    %% exit condition: there are no more IF that can improve the scenario
    if maxVal <= 0
        break;
    end
    %% selection of the best IF
    j = find(possibleIF==maxVal);
    IFindex = j(ceil(rand*size(j,1)),1);
    c=ceil(IFindex/nFormats);
    r=IFindex-nFormats*(c-1);
    newIF = zeros(nFormats,'single');
    newIF(r,c) = 1;
    %% Application of the selected IF to the scenario
    indIF1 = newIF * IF;  % determine IF compositions based on new IF' target format
    indIF2 = IF * newIF;  % determine IF compositions based on new IF' source format
    allIndIF = (allIndIF + indIF1 + indIF2) .* formIdent; % determine all IF compositions
    IFdir = IFdir | newIF; % add the new IF to the set of direct IFs applied to the scenario
    IF = IFdir | indIF1 | indIF2;
    expIF = exportMatrix * IF; % IF deployment into the exporters' representation
    exportMatrix = single(exportMatrix | expIF);
    impIF = importMatrix * IF'; % IF deployment into the importers' representation
    importMatrix = single(importMatrix | impIF);
    %% increment of NIA
    NIA = NIA +1;
end
%% Computation of the coverage
maxConnects = sum(sum(modExp * modImp' & ident) );
connects = sum(sum(logical(exportMatrix * importMatrix' & ident)));
coverage = connects/maxConnects;% interoperability coverage
maxTheo = nExp * (nExp-1);     % fully connected formula
absCoverage = connects/maxTheo; % coverage regarding to the fully connected
end
```

algP2P

The algP2P function implements the algorithm to maximise the interoperability in an interoperability scenario using the Point-to-Point Interoperability approach. This function requires a scenario representation as input, presenting as output the Number of Interoperability Artefacts required to maximise the interoperability in scenario represented, as well as the Interoperability Coverage and Interoperability Absolute Coverage that characterise the interoperability state of the scenario represented after the execution of the algorithm.

```
function [NIA,coverage,absCoverage,exportMatrix,importMatrix] = algP2P(exportMatrix,importMatrix,modifMatrix)
%algP2P Implementation of the algorithm to Maximum Possible
%Interoperability using the P2P Interoperability approach
%   Inputs: exportMatrix - representation of the export capabilities in a specific scenario
%           importMatrix - representation of the import capabilities in a specific scenario
%           modifMatrix - vector representing the systems that are
%           modifiable, (rows marked with '1')
%
%   Outputs: NIA - number of interoperability artefacts added to the scenario
%            coverage - relation between the number of connections in the
%            scenario and the maximum number for the specific scenario
%           absCoverage - relation between the number of connections in the scenario and the maximum number given by the fully connected
%           exportMatrix - final state of the representation of the export capabilities in a specific scenario
%           exportMatrix - final state of the representation of the import capabilities in a specific scenario

[nExp,nFormats] = size(exportMatrix);
nElem = nExp * nFormats;
%% Determination of the initial NIA
Texp = sum(sum(exportMatrix));
Timp = sum(sum(importMatrix));
NIA = Texp + Timp;
ident = ~eye(nExp);
modifiable=repmat(modifMatrix,1,nExp) & ident ;
modExp = single(logical(sum(exportMatrix,2)));
modExpMat = repmat(modExp,1,nExp) & modifiable;     % exporters filter
modImp = single(logical(sum(importMatrix,2)));
modImpMat = repmat(modImp,1,nExp) & modifiable;     % importers filter
while true
    %% Determination of the systems that don't communicate with each other and want to
    connectMatrix = ~(exportMatrix * importMatrix');
    connectExp = connectMatrix & modExpMat;
    connectImp = connectMatrix' & modImpMat;
    possibleImp = connectImp * exportMatrix;
    possibleExp = connectExp * importMatrix;
    allMatrix = [possibleImp(:);possibleExp(:)];
    %% Determination of the the system alteration that allow the greater improve in the scenario communication
    [v,~]=max(allMatrix);
    if v == 0
        break;
    end
    j = find(allMatrix==v);

    if size(j,1)==1
        i=j;
        matrix=1;      %system to modify is an importer
        if i > nElem
            i=i-nElem;
            matrix=2;   %system to modify is an exporter
        end
        c=ceil(i/nExp);
        r=i-nExp*(c-1);
    else
        formatCount = sum(exportMatrix,1) + sum(importMatrix,1);
        [maxF,~] = max(formatCount);        % finds the format more used
        formats = find(formatCount==maxF);  % in case of tie random select one format with the max count
        forInd = formats(ceil(rand*size(formats,1)),1);
        allMatrix = [possibleImp(:,forInd);possibleExp(:,forInd)];
        [v,~]=max(allMatrix);
        if v == 0                   % cannot increase the usage of most common data format
            i = j(ceil(rand*size(j,1)),1); %in case of a tie it is resolved by a random selection
            matrix=1;                  %system to modify is an importer
            if i > nElem
                i=i-nElem;
                matrix=2;              % system to modify is an exporter
            end
            c=ceil(i/nExp);
            r=i-nExp*(c-1);
        else
            set = find(allMatrix==v);
            i = set(ceil(rand*size(set,1)),1); %in case of a tie it is resolved by a random selection
```

```
            c=forInd;
            if i<=nExp
                r=i;
                matrix=1;
            else
                r=i-nExp;
                matrix=2;
            end
        end
    end
    %% Application of the system modification to the scenario
    switch matrix
        case 1
            importMatrix(r,c)=1;
        case 2
            exportMatrix(r,c)=1;
    end
    NIA=NIA+1;
end
%% Computation of the coverage
maxConnects = sum(sum((modExp * modImp') & ident) );
connects = sum(sum(logical(exportMatrix * importMatrix' & ident)));
coverage = connects/maxConnects;
maxTheo = nExp * (nExp-1);        % fully connected formula
absCoverage = connects/maxTheo;   % coverage regarding to the fully connected
end
```

Scenario

This class has the purpose of store all the data required to represent an interoperability scenario. It stores a matric representation of the scenario (*nodesMatrix*), an indication of the systems that are modifiable (*nodesModif*) and the number of different data formats supported in the interoperability scenario.

```
classdef Scenario < handle
    %Scenario Stores the initial settings of a scenario
    %   Has one matrix that represents all the systems present in the
    %   scenario and all data formats that each one supports
    properties
        nodesMatrix %double matrix with systems as rows and the supported data formats as columns. The first represents the export capabilities and the second represents the import capabilities
        nodesModif % vertical vector that indicates which systems are modifiable
        nFormats
    end
    methods
        function obj = Scenario(nodesMatrix, nodesModif)
            obj.nodesMatrix =  nodesMatrix;
            obj.nodesModif =  nodesModif;
            obj.nFormats = size(nodesMatrix,2);
        end
    end
end
```

HSResult

This class is designed to store the results of the execution of the *algHS* function over an interoperability scenario. It stores the total amount of Number of Interoperability Artefacts required to maximise the interoperability in the corresponding scenario in the property *effort*, the Interoperability Coverage and the Interoperability Absolute Coverage in properties coverage and *absCoverage*, the set of IFs the can maximise the interoperability in the scenario in *IFset*, the representation of the interoperability scenario on the maximum interoperability state is stored in the property *finalMatrix*, and the number of systems and data formats in the scenario is indicated in the properties *nNodes* and *nFormats*.

```
classdef HSResult
   %HSResult Stores the result a H&S simulation
properties
      effort         % Stores the final NIA
      coverage       % Stores the Interoperability Coverage
      absCoverage    % Stores the Interoperability Absolute Coverage
      IFset          % Stores the set of IFs applied to the scenario
      finalMatrix    % Scenario representation after the simulation
      nNodes         % Number of system in the scenario
      nFormats       % Number of different data formats in the scenario
   end
   methods
      function obj = saveResult(obj,effort,coverage,absCoverage,finalMatrix,IFset,nNodes,nFormats)
         obj.effort = effort;
         obj.coverage = coverage;
         obj.absCoverage = absCoverage;
         obj.finalMatrix = finalMatrix;
         obj.IFset = IFset;
         obj.nNodes = nNodes;
         obj.nFormats = nFormats;
      end
   end
end
```

P2PResult

This class is designed to store the results of the execution of the algP2P function over an interoperability scenario. It stores the total amount of Number of Interoperability Artefacts required to maximise the interoperability in the corresponding scenario in the property *effort*, the Interoperability Coverage and the Interoperability Absolute Coverage in properties coverage and *absCoverage*, the representation of the interoperability scenario on the maximum interoperability state is stored in the property *finalMatrix*, an indication of which systems are modifiable is represented in property *modifMatrix*, and the number of systems that compose the scenario is indicated in the property *nNodes*.

```
classdef P2PResult
   %P2PResult Stores the result of one P2P algorithm simulation

   properties
      effort         % Stores the final NIA
      coverage       % Stores the Interoperability Coverage
      absCoverage    % Stores the Interoperability Absolute Coverage
      finalMatrix    % Scenario representation after the simulation
      modifMatrix    % Indication of the modifiable systems
      nNodes         % Number of system in the scenario
   end
methods
      function obj = saveResult(obj,effort,coverage,absCoverage,finalMatrix,modifMatrix,nNodes)
         obj.effort = effort;
         obj.coverage = coverage;
         obj.absCoverage = absCoverage;
         obj.finalMatrix = finalMatrix;
         obj.modifMatrix = modifMatrix;
         obj.nNodes = nNodes;
      end
   end
end
```

Distributed Execution of Simulations

In order to enable a more accurate study of the behaviour of the P2P and the H&S Interoperability approaches, the simulation of those approaches in several scenarios is required. Many of those simulations are executed on scenarios produced with the same parameters in order to enable a scenario-independent study. For this reason, each variation of parameters is used to create 500 random scenarios, process repeated during the simulation of each approach. The execution of such simulation methodology consumes a large amount time. In order to deal with this problem the use of a distributed computing approach was required. The best solution found is the use of the MATLAB Distributed Computing Server and of the Parallel Computing Toolbox. The methodology followed to execute distributed simulations and the corresponding MATLAB code are presented in this section.

The first step required to perform a distributed computation is the creation of the parallel job that will be executed, i.e. the definition of the functions that will be concurrently executed. This step is performed in the function *startJob*, where a parallel job is created to run the algorithm of one interoperability approach (P2P Interoperability or H&S Interoperability) on 500 random scenarios created with the same input parameters, as represented in Image 2. This corresponds to 500 executions of the simulation methodology presented in Image 1, i.e. corresponds to 500 executions of the function *simBuilder* using the same input parameters.

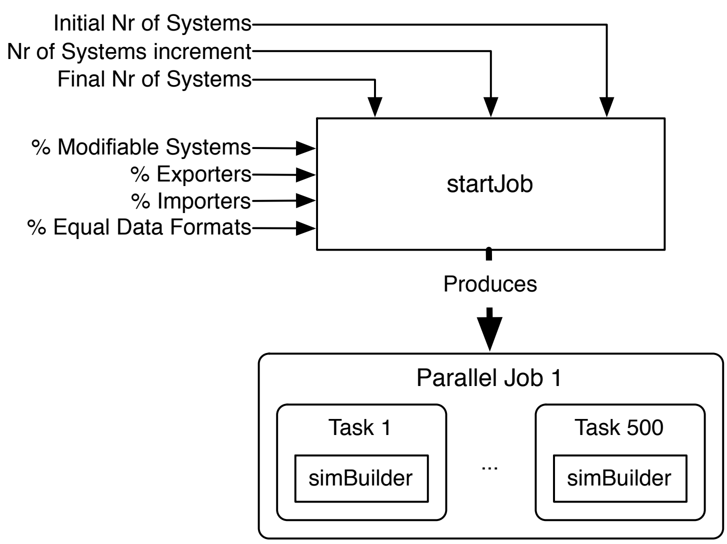


Image 2: Overview of the process of a job for parallel computing

A parallel job is composed by one or more tasks, where each task represents a block of code that must be executed by only one processing unit. In this scope, a task corresponds to a function *simBuilder* with the required input parameters. This setup allows each simulation to be executed independently of the others and therefore to be executed in parallel.

In the distributed computing used, there are three types of entity: a MATLAB Client whose function is to create the parallel job; a MATLAB Job Scheduler, responsible for the distribution of the work and the aggregation of the results; and one to several MATLAB Workers, responsible by the execution of the work. These entities and their relations are presented in Image 3.

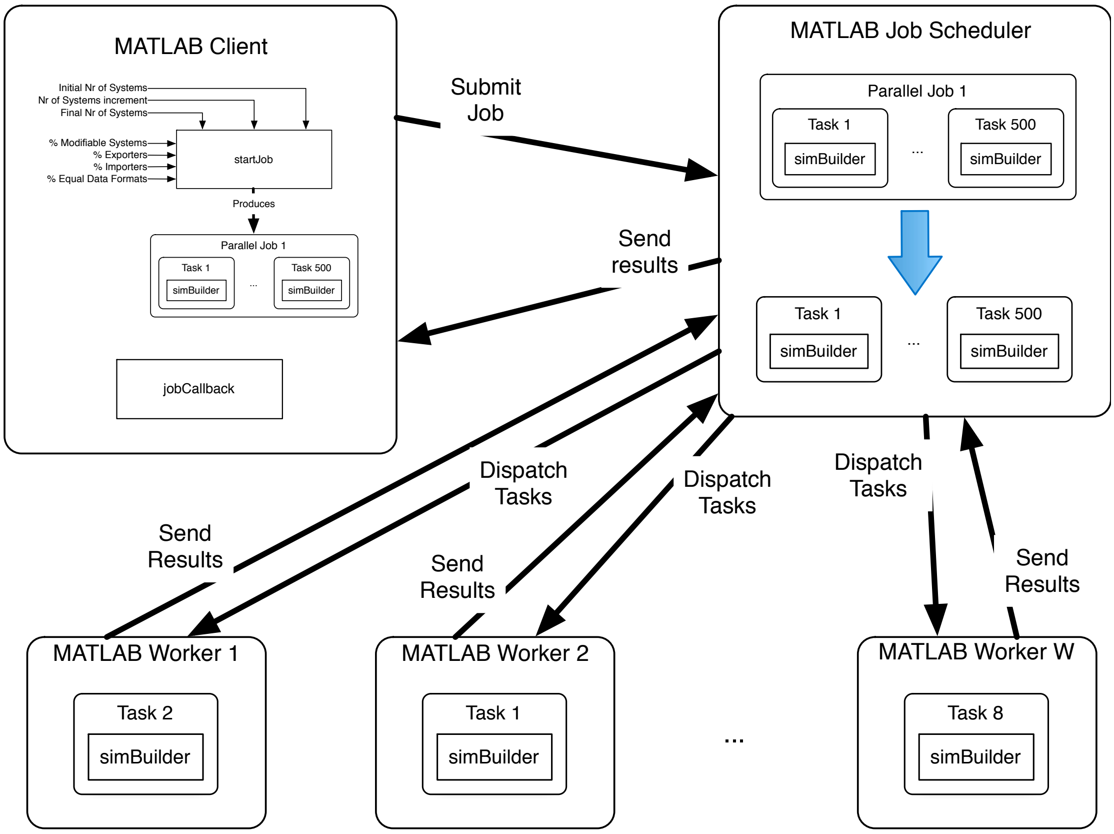


Image 3: Overview of the Distributed Computing Approach used

The distributed execution starts with the definition of the parallel job to be executed. This process corresponds to the *startJob* and is executed by the MATLAB Client. After the creation of the parallel job, it is submitted to the MATLAB Job Scheduler where all the tasks that compose the job are extracted and prepared for execution. The MATLAB Job Scheduler randomly dispatches tasks to idle MATLAB Workers and waits for the reception of the corresponding results. When all the tasks are executed and all results are received by the MATLAB Job Scheduler, all results are aggregated and send to the MATLAB Client that submitted the job, which fires the corresponding jobCallback function, designed to handle these results.

startJob

This function creates the parallel job used to perform the distributed computation of either the P2P or the H&S Interoperability approaches and submits it to the MATLAB Job Scheduler. Each job consists in the execution of ITERATIONS *simBuilder* functions with the same input parameters.

```
function job = startJob(initialSys,stepSys,finalSys,formatsRate,modifRate,exportRate,importRate)
%startJob Creates a Job to simulate the P2P Interoperability or H&S
%algorithms using parallel computation
ITERATIONS =500;
cluster=parcluster;
%% Creates a job the execute the P2P algorithm for 'ITERATIONS' scenarios
% job = createJob(cluster,'FinishedFcn',@jobCallbackP2P,'UserData',{sprintf('Sim-
f%dm%de%di%d',formatsRate*100,modifRate*100,exportRate*100,importRate*100); sprintf('Different Formats: %d%% | Modifiable Systems: %d%% |
Exporters: %d%% | Importers: %d%%',formatsRate*100,modifRate*100,exportRate*100,importRate*100)});
%% Creates a job the execute the H&S Interoperability algorithm for 'ITERATIONS' scenarios
job = createJob(cluster,'FinishedFcn',@jobCallbackHS,'UserData',{sprintf('Sim-
f%dm%de%di%d',formatsRate*100,modifRate*100,exportRate*100,importRate*100); sprintf('Different Formats: %d%% | Modifiable Systems: %d%% |
Exporters: %d%% | Importers: %d%%',formatsRate*100,modifRate*100,exportRate*100,importRate*100)});
for i=1:ITERATIONS
createTask(job,@simBuilder,1,{initialSys,stepSys,finalSys,formatsRate,modifRate,exportRate,importRate});
end
%% Submits the job to the grid for parallel computation
submit(job);
end
```

jobCallbackHS

This function consists the callback fired to handle the results of distributed executions of the algorithm corresponding to the H&S Interoperability approach.

```
function jobCallbackHS( job, ~ )
%jobCallbackHS CallBack function to handle the results of a H&S simulation
%   Fetches the simulations' results from the grid and handles them
fprintf('\n###### JOB %d finished @ %s ########\n',job.ID,datestr(clock));
simCells = fetchOutputs(job);
simSize = size(simCells,1);
for i=1:simSize
   if i==1
      simResult = simCells{i,1};
   else
      simResult.HSResults = [simResult.HSResults;simCells{i,1}.HSResults];
   end
end
filename = job.UserData{1,1};
titleStr = job.UserData{2,1};
% save the results in the filesystem
save(sprintf('../SimResults/HS/%s',filename),'simResult');
% Plot a graphic with the obtained results
fig = plotHS(simResult,titleStr);
% save the graphic data in the filesystem
saveas(fig,sprintf('../SimResults/HS/Figs/%s',filename),'fig');
end
```

jobCallbackP2P

This function consists the callback fired to handle the results of distributed executions of the algorithm corresponding to the P2P Interoperability approach.

```
function jobCallbackP2P( job, ~ )
%jobCallback CallBack function to handle the results of a P2P simulation
%   Fetches the simulations' results from the grid and handles them
fprintf('\n###### JOB %d finished @ %s ########\n',job.ID,datestr(clock));
simCells = fetchOutputs(job);
simSize = size(simCells,1);
for i=1:simSize
   if i==1
      simResult = simCells{i,1};
   else
```

```
        simResult.P2PResults = [simResult.P2PResults;simCells{i,1}.P2PResults];
    end
end
filename = job.UserData{1,1};
titleStr = job.UserData{2,1};
% save the results in the filesystem
save(sprintf('../SimResults/P2P/%s',filename),'simResult');
% Plot a graphic with the obtained results
fig = plotP2P(simResult,titleStr);
% save the graphic data in the filesystem
saveas(fig,sprintf('../SimResults/P2P/Figs/%s',filename),'fig');
end
```

Graphics generation

iteDeterm

Function used to determine the minimum amount of repetitions of simulations with the same parameters required to allow each approach to be scenario-independent. For one approach to be studied the corresponding function to get data must be uncomment and the function of the other approach should be commented.

```
function iteDeterm
% function to plot the graph used to identify the number of iterations
% required to realise scenario-independent simulation
ITERATIONS = 1000;
nia=zeros(1,ITERATIONS);
meanNia=zeros(1,ITERATIONS);
for i=1:ITERATIONS
   sim = simBuilder(50,1,50,0.5,0.8,0.5,0.5);
% Get data from P2P simulations
%    [NIA,~,~,~]=getP2PData(sim);
% Get data from H&S simulations
    [NIA,~,~,~]=getHSData(sim);
   nia(1,i)=NIA(1,1);
   meanNia(1,i) = mean(nia(1,1:i));
end
figure;
title('NIA Cumulative Mean','FontSize',16,'FontWeight','bold');
set(get(gca,'Ylabel'),'String','NIA','FontSize',14);
set(get(gca,'Xlabel'),'String','Nr. Simulations','FontSize',14);
ylim(gca,[0 100]);
hold on
hNIA = plot(meanNia);
set(hNIA,'LineWidth',3);
hold on;
line = ones(1,ITERATIONS).*meanNia(1,ITERATIONS);
hLine = plot(line,'r');
set(hLine,'LineWidth',2.5);
hold on;
grid on;
end
```

loadSimulations

Function used to load the data of simulations stored by jobCallBack functions. It loads the data of simulations for the 25-50-75% of equal data formats at once.

```
function simArray = loadSimulations(modifRate, expRate, impRate)
%loadSimulations Load the data from H&S and P2P simulations
%   modifRate - rate of modifiable systems in the scenario; -1 refes to H&S data
%   expRate - rate of systems with export capabilities
%   impRate - rate of systems with import capabilities
if modifRate == -1  % load H&S data
    dir = '../SimResults/HS/';
    modifRate = 25;
else                % load P2P data
    dir = '../SimResults/P2P/';
end
simArray(1,3) = Simulations;
for f=25:25:75 % load data for the 25-50-75% of equal data formats
    data = load(sprintf('%sSim-f%dm%de%di%d.mat',dir,f,modifRate,expRate,impRate));
    simArray(1,f/25) = data.simResult;
end
end
```

PlotSimilarP2P

This function plots NIA, Interoperability Coverage and Absolute Interoperability Coverage for the simulations of the P2P approach. All the simulations over scenarios with the same percentage of Modifiable systems, percentage of systems with export capabilities, and percentage of systems with import capabilities are grouped in the same graphic.

```
function PlotSimilarP2P( modifRate, expRate, impRate )
%PlotSimilarP2P Plot the behaviour of NIA and Coverages of all P2P simulations
%with modifRate of modifiable systems, expRate of systems with export
%capabilities, and impRate of systems with import capabilities

% Graphic formatting parameters
plotFormat = [0 0 0;'--' 0;'.-.'];
colors = [1 0.5 0.2];

% load of the simulations's data
simArray = loadSimulations( modifRate, expRate, impRate);
nSim = size(simArray,2);

[Effort,nNodes,Coverage,AbsCover] = simArray(1,1).getP2PData;
fig=figure;
[AX,H1,H2] = plotyy(nNodes, Effort(1,:), nNodes, Coverage(1,:),'plot');

set(H2,'color','red');
set(H1,'LineWidth',1.8,'Marker','.','DisplayName','NIA for 25% with equal formats');

determineEquations(nNodes, Effort(1,:),AX(1))

set(H2,'LineWidth',1.8,'Marker','.','DisplayName','% to max possible');

set(get(AX(1),'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
ylim(AX(1),[0 3000]);
set(AX(1),'ytick',(0:150:3000));
set(AX(1),'XTick',0:10:120,'FontSize',14);
hold on;
set(fig, 'CurrentAxes', AX(2));

set(AX(2),'ycolor','r');
set(get(AX(2),'Ylabel'),'String','Coverage','FontSize',14);
set(get(AX(2),'Xlabel'),'String','Number of systems','FontSize',14);
ylim(AX(2),[0 1]);
set(AX(2),'ytick',(0:0.05:1)');
set(AX(2),'FontSize',14);

hold on;

H3=plot(nNodes,AbsCover(1,:),'r.--');
set(H3,'LineWidth',1.8,'DisplayName','% to fully connected');
grid(AX(1));
grid on;
title(sprintf('Equal Formats for: Modifiable Systems - %d%% | Exporters - %d%% | Importers - %d%%',modifRate, expRate,
impRate),'FontSize',16,'FontWeight','bold');
set(fig, 'CurrentAxes', AX(1));
for j=2:nSim
    hold on;
    [eff,nNodes,~,~] = simArray(1,j).getP2PData;
    eh = plot(nNodes,eff(1,:),plotFormat(j,:));
    set(eh,'color',[colors(1,j) 0 colors(1,j)],'LineWidth',3,'DisplayName',sprintf('NIA for %d%% with equal formats',25*j));
    determineEquations(nNodes, eff(1,:),AX(1))
end
lh=legend('show','Location','Southeast');
set(lh,'Color','white');
%-----------------------------------------------------------------------%
%                  AUXILIARY Functions
%-----------------------------------------------------------------------%
function determineEquations(x,y,axis)
[xData, yData] = prepareCurveData( x, y );

% Set up fittype and options.
ft = fittype( 'a*x^2', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( ft );
opts.Display = 'Off';
opts.Lower = -Inf;
opts.StartPoint = 0.403912145588115;
opts.Upper = Inf;

% Fit model to data.
[fitresult, ~] = fit( xData, yData, ft, opts );
```

```
% Save type of fit for "Show equations"
fittypesArray1(1) = 3;

% Save coefficients for "Show Equation"
coeffs1{1} = [coeffvalues(fitresult),0,0];

% "Show equations" was selected
showEquations(fittypesArray1, coeffs1, 2, axis);

%------------------------------------------------------------------------%
function showEquations(fittypes1, coeffs1, digits1, axesh1)
%SHOWEQUATIONS(FITTYPES1,COEFFS1,DIGITS1,AXESH1)
%  Show equations
%  FITTYPES1:  types of fits
%  COEFFS1:  coefficients
%  DIGITS1:  number of significant digits
%  AXESH1:  axes
n = length(fittypes1);
txt = cell(length(n + 1) ,1);
txt{1,:} = ' ';
for i = 1:n
    txt{i + 1,:} = getEquationString(fittypes1(i),coeffs1{i},digits1,axesh1);
end
text(.05,.95,txt,'parent',axesh1, ...
    'verticalalignment','top','Units','normalized','FontSize',13,'FontWeight','bold');
%------------------------------------------------------------------------%
function [s1] = getEquationString(fittype1, coeffs1, digits1, axesh1)
%GETEQUATIONSTRING(FITTYPE1,COEFFS1,DIGITS1,AXESH1)
%  Get show equation string
%  FITTYPE1:  type of fit
%  COEFFS1:  coefficients
%  DIGITS1:  number of significant digits
%  AXESH1:  axes
if isequal(fittype1, 0)
    s1 = 'Cubic spline interpolant';
elseif isequal(fittype1, 1)
    s1 = 'Shape-preserving interpolant';
else
    op = '+-';
    format1 = ['%s %0.',num2str(digits1),'g*x^{%s} %s'];
    format2 = ['%s %0.',num2str(digits1),'g'];
    xl = get(axesh1, 'xlim');
    fit =  fittype1 - 1;
    s1 = sprintf('NIA ~=');
    th = text(xl*[.95;.05],1,s1,'parent',axesh1, 'vis','off');
    if abs(coeffs1(1) < 0)
        s1 = [s1 ' -'];
    end
    for i = 1:fit
        sl = length(s1);
        if ~isequal(coeffs1(i),0) % if exactly zero, skip it
            s1 = sprintf(format1,s1,abs(coeffs1(i)),num2str(fit+1-i), op((coeffs1(i+1)<0)+1));
        end
        if (i==fit) && ~isequal(coeffs1(i),0)
            s1(end-5:end-2) = []; % change x^1 to x.
        end
        set(th,'string',s1);
        et = get(th,'extent');
        if et(1)+et(3) > xl(2)
            s1 = [s1(1:sl) sprintf('\n     ') s1(sl+1:end)];
        end
    end
    if ~isequal(coeffs1(fit+1),0)
        sl = length(s1);
        s1 = sprintf(format2,s1,abs(coeffs1(fit+1)));
        set(th,'string',s1);
        et = get(th,'extent');
        if et(1)+et(3) > xl(2)
            s1 = [s1(1:sl) sprintf('\n     ') s1(sl+1:end)];
        end
    end
    delete(th);
    % Delete last "+"
    if isequal(s1(end),'+')
        s1(end-1:end) = []; % There is always a space before the +.
    end
    if length(s1) == 3
        s1 = sprintf(format2,s1,0);
    end
end
```

PlotSimilarHS

This function plots NIA, Interoperability Coverage and Absolute Interoperability Coverage for the simulations of the H&S Interoperability approach. All the simulations over scenarios with the same percentage of systems with export capabilities, and percentage of systems with import capabilities are grouped in the same graphic.

```
function PlotSimilarHS( expRate, impRate )
%PlotSimilarHS Plot the behaviour of NIA and Coverages of all H&S simulations
%with modifRate of modifiable systems, expRate of systems with export
%capabilities, and impRate of systems with import capabilities

% Graphic formatting parameters
plotFormat = [0 0 0;':' 0;'.-.'];
colors = [1 0.5 0.2];

simArray = loadSimulations( -1, expRate, impRate); %modifRate is -1 to load the HS data
nSim = size(simArray,2);

[Effort,nNodes,Coverage,AbsCover] = simArray(1,1).getHSData;
fig=figure;
[AX,H1,H2] = plotyy(nNodes, Effort(1,:), nNodes, Coverage(1,:),'plot');

determineEquations(nNodes, Effort(1,:),AX(1))

set(H2,'color','red');
set(H1,'LineWidth',1.5,'Marker','.','DisplayName','NIA for 25% with equal formats');
set(H2,'LineWidth',1.5,'Marker','.','DisplayName','% to max possible');
set(get(AX(1),'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
ylim(AX(1),[0 505]);
set(AX(1),'ytick',(0:25:500));
set(AX(1),'XTick',0:10:120,'FontSize',14);
set(fig, 'CurrentAxes', AX(2));
set(AX(2),'ycolor','r');
set(get(AX(2),'Ylabel'),'String','Coverage','FontSize',14);
set(get(AX(2),'Xlabel'),'String','Number of systems','FontSize',14);
ylim(AX(2),[0 1.01]);
set(AX(2),'ytick',(0:0.05:1)');
set(AX(2),'FontSize',14)
set(AX(2),'XTick',0:10:120,'FontSize',14);
hold on;
H3=plot(nNodes,AbsCover(1,:),'r.--');
set(H3,'LineWidth',3,'DisplayName','% to fully connected');
grid(AX(1));
grid on;
title(sprintf('Equal Format for: Exporters - %d%% | Importers - %d%%', expRate, impRate),'FontSize',16,'FontWeight','bold');
set(fig, 'CurrentAxes', AX(1));
for i=2:nSim
    hold on;
    [eff,nNodes,~,~] = simArray(1,i).getHSData;
    eh = plot(nNodes,eff(1,:),plotFormat(i,:));
    set(eh,'color',[colors(1,i) 0 colors(1,i)],'LineWidth',3,'DisplayName',sprintf('NIA for %d%% with equal formats',25*i));
    determineEquations(nNodes, eff(1,:),AX(1))

end
lh=legend('show','Location','Southeast');
set(lh,'Color','white');
%-----------------------------------------------------------------------%
%                  AUXILIARY Functions
%-----------------------------------------------------------------------%
function determineEquations(x,y,axis)
[xData, yData] = prepareCurveData( x, y );

% fitResults1 = polyfit(xData, yData, 1);

% Set up fittype and options.
ft = fittype( 'a*x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( ft );
opts.Display = 'Off';
opts.Lower = -Inf;
opts.StartPoint = 0.403912145588115;
opts.Upper = Inf;

% Fit model to data.
[fitresult, ~] = fit( xData, yData, ft, opts );
% coeffvalues(fitresult)

% Save type of fit for "Show equations"
fittypesArray1(1) = 2;

% Save coefficients for "Show Equation"
coeffs1{1} = [coeffvalues(fitresult),0,0];

% "Show equations" was selected
showEquations(fittypesArray1, coeffs1, 4, axis);
%-----------------------------------------------------------------------%
function showEquations(fittypes1, coeffs1, digits1, axesh1)
%SHOWEQUATIONS(FITTYPES1,COEFFS1,DIGITS1,AXESH1)
%  Show equations
%  FITTYPES1:  types of fits
%  COEFFS1:  coefficients
%  DIGITS1:  number of significant digits
%  AXESH1:  axes

n = length(fittypes1);
txt = cell(length(n + 1) ,1);
txt{1,:} = ' ';
for i = 1:n
    txt{i + 1,:} = getEquationString(fittypes1(i),coeffs1{i},digits1,axesh1);
end
```

```
text(.05,.95,txt,'parent',axesh1, ...
    'verticalalignment','top','Units','normalized','FontSize',13,'FontWeight','bold');
%-----------------------------------------------------------------------%
function [s1] = getEquationString(fittype1, coeffs1, digits1, axesh1)
%GETEQUATIONSTRING(FITTYPE1,COEFFS1,DIGITS1,AXESH1)
%  Get show equation string
%  FITTYPE1:  type of fit
%  COEFFS1:  coefficients
%  DIGITS1:  number of significant digits
%  AXESH1:  axes
if isequal(fittype1, 0)
    s1 = 'Cubic spline interpolant';
elseif isequal(fittype1, 1)
    s1 = 'Shape-preserving interpolant';
else
    op = '+-';
    format1 = ['%s %0.',num2str(digits1),'g*x^{%s} %s'];
    format2 = ['%s %0.',num2str(digits1),'g'];
    xl = get(axesh1, 'xlim');
    fit =  fittype1 - 1;
    s1 = sprintf('NIA ~=');
    th = text(xl*[.95;.05],1,s1,'parent',axesh1, 'vis','off');
    if abs(coeffs1(1) < 0)
        s1 = [s1 ' -'];
    end
    for i = 1:fit
        sl = length(s1);
        if ~isequal(coeffs1(i),0) % if exactly zero, skip it
            s1 = sprintf(format1,s1,abs(coeffs1(i)),num2str(fit+1-i), op((coeffs1(i+1)<0)+1));
        end
        if (i==fit) && ~isequal(coeffs1(i),0)
            s1(end-5:end-2) = []; % change x^1 to x.
        end
        set(th,'string',s1);
        et = get(th,'extent');
        if et(1)+et(3) > xl(2)
            s1 = [s1(1:sl) sprintf('\n     ') s1(sl+1:end)];
        end
    end
    if ~isequal(coeffs1(fit+1),0)
        sl = length(s1);
        s1 = sprintf(format2,s1,abs(coeffs1(fit+1)));
        set(th,'string',s1);
        et = get(th,'extent');
        if et(1)+et(3) > xl(2)
            s1 = [s1(1:sl) sprintf('\n     ') s1(sl+1:end)];
        end
    end
    delete(th);
    % Delete last "+"
    if isequal(s1(end),'+')
        s1(end-1:end) = []; % There is always a space before the +.
    end
    if length(s1) == 3
        s1 = sprintf(format2,s1,0);
    end
end
```

PlotSimilarVS

This function plots NIA, Interoperability Coverage and Absolute Interoperability Coverage for the simulations of both P2P and H&S Interoperabilty approaches. All the simulations over scenarios with the same percentage of Modifiable systems, percentage of systems with export capabilities, and percentage of systems with import capabilities are grouped in the same graphic.

```
function PlotSimilarVS( modifRate, expRate, impRate )
%PlotSimilarVS Plot the behaviour of NIA and both coverage values of both
%P2P and H&S approaches for all simulations with modifRate of modifiable
%systems, expRate of systems with export capabilities, and impRate
%of systems with import capabilities
plotFormat = ['-' 0 0;'--' 0;'.-.'];
colorP2P = [0 0 0.8000; 0.2000 0.2000 1.0000; 0.5000 0.5000 1.0000];
colorHS = [0.6000 0 0; 1 0 0; 1.0000 0.4000 0.4000];
simP2P = loadSimulations( modifRate, expRate, impRate);
simHS= loadSimulations( -1, expRate, impRate);
nSim = size(simP2P,2);
[Effort,nNodes,Coverage,AbsCover] = simP2P(1,1).getP2PData;
fig=figure;
[AX,H1,H2] = plotyy(nNodes, Effort(1,:), nNodes, Coverage(1,:),'plot'); % plot P2P Interoperability Coverage
delete(H1);
set(H2,'color','k');
set(H2,'LineWidth',3,'DisplayName','P2P: % to max possible');
set(get(AX(1),'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
```

```
ylim(AX(1),[0 3030]);
set(AX(1),'ytick',(0:150:3000));
set(AX(1),'XTick',0:10:120,'FontSize',14);
hold on;
set(fig, 'CurrentAxes', AX(2));
set(AX(2),'ycolor','k');
set(get(AX(2),'Ylabel'),'String','Coverage','FontSize',14);
set(get(AX(2),'Xlabel'),'String','Number of systems','FontSize',14);
ylim(AX(2),[0 1.01]);
set(AX(2),'ytick',(0:0.05:1)');
set(AX(2),'FontSize',14)
hold on;
h=plot(nNodes,AbsCover(1,:)); % plot P2P Absolute Interoperability Coverage
set(h,'Color',[0 0.6 0],'LineWidth',3,'DisplayName',' P2P: % to fully connected');
[~,nNodes,Coverage,AbsCover] = simHS(1,1).getHSData;
h=plot(nNodes,Coverage(1,:),'--k'); % plot H&S Interoperability Coverage
set(h,'LineWidth',3,'DisplayName',' H&S: % to max possible');
h=plot(nNodes,AbsCover(1,:),'--'); % plot H&S Absolute Interoperability Coverage
set(h,'Color',[0 0.6 0],'LineWidth',3,'DisplayName',' H&S: % to fully connected');
set(fig, 'CurrentAxes', AX(1));
% plot P2P NIA
for j=1:nSim
    [eff,nNodes,~,~] = simP2P(1,j).getP2PData;
    eh = plot(nNodes,eff(1,:),plotFormat(j,:));
    set(eh,'color',colorP2P(j,:),'LineWidth',3,'DisplayName',sprintf(' P2P: NIA for %d%% with equal formats',25*j));
    hold on;
end
% plot H&S NIA
for j=1:nSim
    [eff,nNodes,~,~] = simHS(1,j).getHSData;
    eh = plot(nNodes,eff(1,:),plotFormat(j,:));
    set(eh,'color',colorHS(j,:),'LineWidth',3,'DisplayName',sprintf(' H&S: NIA for %d%% with equal formats',25*j));
    hold on;
end
title(sprintf('NIA P2P vs H&S: Modifiable Systems - %d%% | Exporters - %d%% | Importers - %d%%',modifRate, expRate,
impRate),'FontSize',16,'FontWeight','bold');
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
set(get(gca,'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
grid(AX(1));
grid on;
set(gca,'FontSize',14)
lh=legend('show','Location','SouthEast');
set(lh,'Color','white');
end
```

PlotSimilarVSsmall

This function is similar to function *PlotSimilarVS*, differing on the number of systems presented in order to favor the study of the small-scale behavior. New simulations were required to produce the data used in this function. The alterations were the reduction of the system step to 2 and of the maximum number of systems in a scenario to 30.

```
function PlotSimilarVSsmall( modifRate, expRate, impRate )
%PlotSimilarVSsmall Plots the NIA and Interoperability of both P2P and
%H&S in smaller scenarios in order to study the small-scale behaviour
% Graphic formatting parameters
plotFormat = ['-' 0 0;'--' 0;'.-.'];
colorP2P = [0 0 0.8000; 0.8 0 0; 0 0.6 0];
colorHS = [0 0 0.8000; 0.8 0 0; 0 0.6 0];
% definition of the observation window
maxSys = 30;
minSys =2;
simP2P = loadSimulations( modifRate, expRate, impRate);
simHS = loadSimulations( -1, expRate, impRate);
nSim = size(simP2P,2);
[Effort,nNodes,Coverage,~] = simP2P(1,1).getP2PData(minSys,maxSys);
fig=figure;
[AX,H1,H2] = plotyy(nNodes, Effort(1,:), nNodes, Coverage(1,:),'plot');
delete(H1);
set(H2,'color','k');
set(H2,'LineWidth',3,'DisplayName','P2P: % to max possible');
set(get(AX(1),'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
ylim(AX(1),[0 60.6]);
set(AX(1),'ytick',(0:3:60));
xlim(AX(1),[1 maxSys]);
xlim(AX(2),[1 maxSys]);
set(AX(1),'XTick',2:2:maxSys,'FontSize',14);
set(AX(2),'XTick',2:2:maxSys,'FontSize',14);
hold on;
set(fig, 'CurrentAxes', AX(2));
set(AX(2),'ycolor','k');
set(get(AX(2),'Ylabel'),'String','Coverage','FontSize',14);
set(get(AX(2),'Xlabel'),'String','Number of systems','FontSize',14);
ylim(AX(2),[0 1.01]);
set(AX(2),'ytick',(0:0.05:1)');
```

```
set(AX(2),'FontSize',14)
hold on;
[~,nNodes,Coverage,~] = simHS(1,1).getHSData(minSys,maxSys);
h=plot(nNodes,Coverage(1,:),'--k');
set(h,'LineWidth',3,'DisplayName',' H&S: % to max possible');
set(fig, 'CurrentAxes', AX(1));
hold on;
for j=1:nSim
    [eff,nNodes,~,~] = simP2P(1,j).getP2PData(minSys,maxSys);
    eh = plot(nNodes,eff(1,:),plotFormat(1,:));
    set(eh,'color',colorP2P(j,:),'LineWidth',3,'DisplayName',sprintf(' P2P: NIA for %d%% with equal formats',25*j));
    hold on;
end
for j=1:nSim
    [eff,nNodes,~,~] = simHS(1,j).getHSData(minSys,maxSys);
    eh = plot(nNodes,eff(1,:),plotFormat(2,:));
    set(eh,'color',colorHS(j,:),'LineWidth',3,'DisplayName',sprintf(' H&S: NIA for %d%% with equal formats',25*j));
    hold on;
end
title(sprintf('NIA P2P vs H&S: Modifiable Systems - %d%% | Exporters - %d%% | Importers - %d%%',modifRate, expRate,
impRate),'FontSize',16,'FontWeight','bold');
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
set(get(gca,'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
grid(AX(1));
grid on;
set(gca,'FontSize',14)
lh=legend('show','Location','SouthEast');
set(lh,'Color','white');
end
```

PlotSimilarCovP2P

This function is designed to plot the Interoperability Coverage and Absolute Interoperability Coverage obtained in the simulations of the P2P approach. All coverage values obtained from scenarios whose percentage of modifiable systems is equal to *modifRate* are grouped in a graphic.

```
function PlotSimilarCovP2P( modifRate )
%PlotSimilarCovP2P Groups and plots the Interoperability Coverage and Absolute
%Interoperability Coverage for all the P2P simulations on scenarios with
%modifRate percentage of modifiable systems
% definition of the export and import rates to be considered
expRate = [25 50 50 75];
impRate = [75 50 75 75];
colors = ['b';'g';'k';'m'];    %definition of the graphics colours
simArray = cell(1,4);
for i=1:4
    simArray(1,i) = {loadSimulations( modifRate, expRate(1,i), impRate(1,i))};
end
simresult = simArray{1,1};
[~,nNodes,Coverage,~] = simresult(1,1).getP2PData;
figure;
h_cov = plot(nNodes,Coverage(1,:),'r--');
set(h_cov,'LineWidth',1.7,'DisplayName',sprintf('%s',' Interoperability Coverage'));
set(get(gca,'Ylabel'),'String','Coverage','FontSize',16);
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',16);
ylim(gca,[0 1]);
set(gca,'ytick',(0:0.05:1)');
set(gca,'FontSize',16)
set(gca,'XTick',0:10:120,'FontSize',16);
hold on;
grid on;
title(sprintf('Different Coverages for: Modifiable Systems - %d%%',modifRate),'FontSize',18,'FontWeight','bold');
for i=1:4
    hold on;
    simresult = simArray{1,i};
    [~,nNodes,~,absCover] = simresult(1,1).getP2PData;
    eh = plot(nNodes,absCover(1,:),colors(i,:));
    set(eh,'LineWidth',1.7,'DisplayName',sprintf('Absolute Coverage for %d/%d;',expRate(1,i),impRate(1,i)));
end
lh=legend('show','Location','Southeast');
set(lh,'Color','white');
end
```

PlotSimilarCovHS

This function is designed to plot, in a single graphic, the Interoperability Coverage and Absolute Interoperability Coverage obtained in the simulations of the H&S approach.

```
function PlotSimilarCovHS
%PlotSimilarCovHS Groups and plots the Interoperability Coverage and Absolute
%Interoperability Coverage for all simulations of the H&S Interoperability

% definition of the export and import rates to be considered
expRate = [25 50 50 75];
impRate = [75 50 75 75];
colors = ['b';'g';'k';'m'];     %definition of the graphics colours
simArray = cell(1,4);

for i=1:4
    simArray(1,i) = {loadSimulations( -1, expRate(1,i), impRate(1,i))};
end
simresult = simArray{1,1};
[~,nNodes,Coverage,~] = simresult(1,1).getHSData;

figure;
h_cov = plot(nNodes,Coverage(1,:),'r--');
set(h_cov,'LineWidth',1.7,'DisplayName',sprintf('%s',' Interoperability Coverage'));
set(get(gca,'Ylabel'),'String','Coverage','FontSize',16);
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',16);
ylim(gca,[0 1]);
set(gca,'ytick',(0:0.05:1)');
set(gca,'FontSize',16)
set(gca,'XTick',0:10:120,'FontSize',16);
hold on;
grid on;
title('Different Coverages for H&S','FontSize',18,'FontWeight','bold');
for i=1:4
    hold on;
    simresult = simArray{1,i};
    [~,nNodes,~,absCover] = simresult(1,1).getHSData;
    eh = plot(nNodes,absCover(1,:),colors(i,:));
    set(eh,'LineWidth',1.7,'DisplayName',sprintf('Absolute Coverage for %d/%d;',expRate(1,i),impRate(1,i)));
end
lh=legend('show','Location','Southeast');
set(lh,'Color','white');
end
```

PlotSimilarCovVS

This function is designed to plot the Interoperability Coverage and Absolute Interoperability Coverage obtained in the simulations of the P2P and H&S Interoperability approaches. All coverage values obtained from scenarios whose percentage of modifiable systems is equal to *modifRate* are grouped in a graphic.

```
function PlotSimilarCovVS( modifRate )
%PlotSimilarCovVS Groups and plots the Absolute Interoperability Coverage
%and/or the Interoperability Coverage of all the P2P and H&S simulations
%on scenarios with modifRate percentage of modifiable systems

% Graphic formatting parameters
expImp = [25,75;50,50;50,75;75,75];
colors = [0 0 0; 0.7 0.6 0; 1 0 0; 0 0.6 0;...
    0 0 0.4];
markerP2P = '-';
markerHS = '--';

simP2P = cell(1,4);
simHS = cell(1,4);
for i=1:4
    simP2P(1,i) = {loadSimulations( modifRate, expImp(i,1), expImp(i,2))};
    simHS(1,i) = {loadSimulations( -1, expImp(i,1), expImp(i,2))};
end
figure;
%% plot interoperability coverages
simrP2P = simP2P{1,1};
[~,nNodes,Coverage,~] = simrP2P(1,1).getP2PData;
eh = plot(nNodes,Coverage(1,:),markerP2P);
set(eh,'color',colors(1,:),'LineWidth',3,'DisplayName',' P2P: % to Max Possible');
hold on;
simrHS = simHS{1,1};
[~,nNodes,Coverage,~] = simrHS(1,1).getHSData;
eh = plot(nNodes,Coverage(1,:),markerHS);
set(eh,'color',colors(1,:),'LineWidth',3,'DisplayName',' H&S: % to Max Possible');
hold on;
%% plot absolute coverages
for i=1:4
    %plot P2P coverage
    simrP2P = simP2P{1,i};
    [~,nNodes,~,absCov] = simrP2P(1,1).getP2PData;
    eh = plot(nNodes,absCov(1,:),markerP2P);
    set(eh,'color',colors(i+1,:),'LineWidth',3,'DisplayName',sprintf(' P2P: %% to Fully Connected for: Exp:%d Imp:%d',expImp(i,1),expImp(i,2)));
    hold on;
```

```
    %plot H&S coverage
    simrHS = simHS{1,i};
    [~,nNodes,~,absCov] = simrHS(1,1).getHSData;
    eh = plot(nNodes,absCov(1,:),markerHS);
    set(eh,'color',colors(i+1,:),'LineWidth',3,'DisplayName',sprintf(' H&S: %% to Fully Connected for: Exp:%d Imp:%d',expImp(i,1),expImp(i,2)));
    hold on;
end

title(sprintf('Coverage P2P vs H&S: Modifiable Systems - %d%%',modifRate),'FontSize',16,'FontWeight','bold');
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
set(get(gca,'Ylabel'),'String','Coverage','FontSize',14);
grid on;
ylim(gca,[0 1]);
set(gca,'FontSize',14)
lh=legend('show','Location','NorthWest');
set(lh,'Color','white');
end
```

CovSysBuilder

This function has the purpose of control the actions required to plot the graphics that relate the Interoperability Coverage with the percentage of modifiable systems in the scenario. It is capable of plot the data of both P2P and H&S approaches in the same graphic. If only the data of one approach is wanted, the function to plot the other approach must be commented.

```
function CovSysBuilder
%covSysBuilder Coordinates the actions required to build the graphics of
%Interoperability Coverage by percentage of modifiable systems


figure
title('% of Modifiable Systems Vs. Coverage','FontSize',18,'FontWeight','bold');
hold on
for n=20:10:100
    fprintf('Nr Systems: %d\n',n);
    sim = calcCovbyMod(n);
    % plot P2P coverage by percentage of modifiable systems
    plotCovbyModP2P(sim);
    hold on
    % plot H&S coverage by percentage of modifiable systems
    plotCovbyModHS(sim);
end
end
```

calcCovbyMod

This function performs the simulations of both interoperability approaches, ranging the percentage of systems modifiable from 0 to 100%, 5 by 5%.

```
function covSim = calcCovbyMod( nSystems )
%calcCovbyMod Executes both P2P and H&S simulations of scenarios composed
%by nSystems systems where the percentage of modifiable systems ranges from
%0 to 100


covSim = Simulations;
iterations = 100;
for ite=1:iterations
    index=1;
    modSim = Simulations;
    for modif=0:5:100
        % *note* simBuilder must be configured to execute both P2P and H&S simulations
        sim=simBuilder( nSystems, 1, nSystems, 0.5, modif * 0.01, 0.25, 0.75);
        index=index+1;
        modSim.P2PResults = [modSim.P2PResults,sim.P2PResults];
        modSim.HSResults = [modSim.HSResults,sim.HSResults];
    end
    covSim.P2PResults = [covSim.P2PResults;modSim.P2PResults];
    covSim.HSResults = [covSim.HSResults;modSim.HSResults];
end
```

plotCovbyModP2P

Function used to plot the graphic of Interoperability Coverage by percentage of modifiable systems in the P2P approach.

```
function plotCovbyModP2P( CovSim )
%plotCovbyModP2P Plot coverage by percentage of modifiable system for P2P
%simulations

[~,nNodes,Coverage,~] = getP2PData(CovSim);
xdata = 0:5:100;
ydata = Coverage(1,:);
nSystems = nNodes(1);
h_cov = plot(xdata,ydata,'o');
set(h_cov,'LineWidth',1.7,'DisplayName',sprintf(' Interoperability Coverage for %d systems',nSystems));
%---- colour selection ------
color = nSystems * 0.01;
quo = nSystems / 30;
int = fix(quo);
dec = quo - int;
if dec == 0
   index = 3;
elseif dec <0.5
   index = 1;
else
   index = 2;
end
colors = zeros(1,3);
colors(1,index) = color;
%----------------------------
set(h_cov,'color',colors,'MarkerFaceColor',colors);
set(get(gca,'Ylabel'),'String','Coverage','FontSize',16);
set(get(gca,'Xlabel'),'String','% of Modifiable Systems','FontSize',16);
ylim(gca,[0 1]);
set(gca,'ytick',(0:0.05:1)');
set(gca,'FontSize',16)
set(gca,'FontSize',16);
grid on;
if nSystems==90
   % Find x values for plotting the fit based on xlim
   axesLimits1 = xlim(gca);
   xplot1 = linspace(axesLimits1(1), axesLimits1(2));
   fitResults1 = polyfit(xdata, ydata, 2);
   yplot1 = polyval(fitResults1, xplot1);
   hold on
   plot(xplot1,yplot1,'LineWidth',2,'Color',[0 0.75 0.75]);
end
end
```

plotCovbyModHS

Function used to plot the graphic of Interoperability Coverage by percentage of modifiable systems in the H&S approach.

```
function plotCovbyModHS( CovSim )
%plotCovbyModHS Plot coverage by percentage of modifiable system for H&S
%simulations

[~,nNodes,Coverage,~] = getHSData(CovSim);
xdata = 0:5:100;
ydata = Coverage(1,:);
nSystems = nNodes(1);
h_cov = plot(xdata,ydata,'o');
set(h_cov,'LineWidth',1.7,'DisplayName',sprintf(' H&S Interoperability Coverage for %d systems',nSystems));

%---- colour selection ------
color = nSystems * 0.01;
quo = nSystems / 30;
int = fix(quo);
dec = quo - int;
if dec == 0
   index = 3;
elseif dec <0.5
   index = 1;
else
   index = 2;
end
colors = zeros(1,3);
colors(1,index) = color;
%----------------------------
```

```
set(h_cov,'color',colors,'MarkerFaceColor',colors);
set(get(gca,'Ylabel'),'String','Coverage','FontSize',16);
set(get(gca,'Xlabel'),'String','% of Modifiable Systems','FontSize',16);
ylim(gca,[0 1]);
set(gca,'ytick',(0:0.05:1)');
set(gca,'FontSize',16)
set(gca,'FontSize',16);
grid on;

if nSystems==90
    % Find x values for plotting the fit based on xlim
    axesLimits1 = xlim(gca);
    xplot1 = linspace(axesLimits1(1), axesLimits1(2));
    fitResults1 = polyfit(xdata, ydata, 2);
    yplot1 = polyval(fitResults1, xplot1);
    hold on
    plot(xplot1,yplot1,'LineWidth',2,'Color',[1 0.75 0.75]);
end
end
```

plotCoeffs

This function is designed to plot the graphic with the NIA of the P2P and H&S approaches using the corresponding coefficients.

```
function plotCoeffs
%plotCoeffs Coordinates the actions required to build the graphics of the
%NIAs using the approximate coefficients

%definition of the P2P Coefficients
coeffP2P = [0.029; 0.037; 0.030; 0.049; 0.063; 0.05; 0.067; 0.085; 0.068; ...
    0.077; 0.1; 0.084; 0.057; 0.073; 0.06; 0.034; 0.043; 0.036; ...
    0.11; 0.143; 0.117; 0.08; 0.106; 0.086; 0.048; 0.061; 0.05; ...
    0.159; 0.204; 0.164; 0.118; 0.15; 0.12; 0.068; 0.087; 0.07];

%definition of the H&S Coefficients
coeffHS = [1.497; 1.898; 2.231; ...
    2.255; 1.918;1.510; ...
    3.360; 2.852; 2.247];

%% Application of the coefficients
[sys,niaMatrixP2P] = buildCoeffGraphsP2P( coeffP2P );
[~,niaMatrixHS] = buildCoeffGraphsHS( coeffHS );
%% Graphics Plot
figure

% ------------- P2P plot -------------------------
nsize= size(niaMatrixP2P,1);
hold on
for i=2:nsize
    h=plot(sys,niaMatrixP2P(i,:));
    set(h,'LineWidth',1.5,'color',[0.59 0.76 0.93]);
    hold on
end

% ------------- H&S plot -----------------------
nsize= size(niaMatrixHS,1);
hold on
for i=2:nsize
    h=plot(sys,niaMatrixHS(i,:));
    set(h,'LineWidth',1.5,'color',[0.5529    0.3922    0.3569]);
    hold on
end

% plot of baseline of the P2P coefficients
h1=plot(sys,niaMatrixP2P(1,:),'b');
set(h1,'LineWidth',3.5);
% plot of baseline of the H&S coefficients
h1=plot(sys,niaMatrixHS(1,:),'r');
set(h1,'LineWidth',3.5);
set(get(gca,'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
grid on;
end
```

buildCoeffGraphsP2P

This function applies the coefficients that allow approximating the NIA required by the P2P Interoperability approach.

```
function [sys_array,niaMatrix] = buildCoeffGraphsP2P( coeff_array )
%buildCoeffGraphsP2P Apply the coefficients of P2P to approximate the NIA
%for scenarios whose dimension ranges from 5 to 115 systems

sys_array = 5:115;
nsys = size(sys_array,2);
ncoef = size(coeff_array,1);
niaMatrix = zeros(ncoef,nsys);
coeffBase = ones(1,nsys);
for i=1:ncoef
    % coeffBase matrix is used to make both input arguments to have the
    % same size
    niaMatrix(i,:) = arrayfun(@applyCoeff,sys_array,coeffBase*coeff_array(i,1));
end
% function to applied to each element by the arrayfun function
function nia = applyCoeff(sys,coef)
nia = coef*(sys^2);
```

buildCoeffGraphsHS

This function applies the coefficients that allow approximating the NIA required by the H&S Interoperability approach.

```
function [sys_array,niaMatrix] = buildCoeffGraphsHS( coeff_array )
%buildCoeffGraphsHS Apply the coefficients of H&S to approaximate the NIA
%for scenarios whose dimension ranges from 5 to 115 systems

sys_array = 5:115;
nsys = size(sys_array,2);
ncoef = size(coeff_array,1);
niaMatrix = zeros(ncoef,nsys);
coeffBase = ones(1,nsys);
for i=1:ncoef
    % coeffBase matrix is used to make both input arguments to have the
    % same size
    niaMatrix(i,:) = arrayfun(@applyCoeff,sys_array,coeffBase*coeff_array(i,1));
end
% function to applied to each element by the arrayfun function
function nia = applyCoeff(sys,coef)
nia = coef*sys;
```

PlotSimilarCovVSallModf

This function plots the Absolute Interoperability Coverage of both P2P and H&S approaches for a specific percentage of systems with export capabilities and percentage of systems with import capabilities.

```
function PlotSimilarCovVSallModf( expRate, impRate )
%PlotSimilarCovVSallModf Plot the Absolute Interoperability Coverage of
%both interoperability approaches for a specific expRate-Imprate pair

% Graphic formatting parameters
colors = [0 0 0; 0.7 0.6 0; 1 0 0; 0 0.6 0;...
    0 0 0.4];
markerP2P = '-';
markerHS = '--';

modf = [25, 50,75];
simP2P = cell(1,4);
simHS = loadSimulations( -1, expRate, impRate);
for i=1:3
    simP2P(1,i) = {loadSimulations( modf(1,i), expRate, impRate)};
end
figure;
[~,nNodes,~,absCov] = simHS(1,1).getHSData;
eh = plot(nNodes,absCov(1,:),markerHS);
set(eh,'color',colors(1,:),'LineWidth',3,'DisplayName',' H&S: %% to Fully Connected');
hold on;
for i=1:3
    simrP2P = simP2P{1,i};
    [~,nNodes,~,absCov] = simrP2P(1,1).getP2PData;
    eh = plot(nNodes,absCov(1,:),markerP2P);
    set(eh,'color',colors(i+1,:),'LineWidth',3,'DisplayName',sprintf(' P2P: %% to Fully Connected for %d%% Modifiable systems',modf(1,i)));
    hold on;
end
title(sprintf('Coverage P2P vs H&S: Exporters - %d%% Importers - %d%%',expRate, impRate),'FontSize',16,'FontWeight','bold');
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
set(get(gca,'Ylabel'),'String','Coverage','FontSize',14);
```

```
grid on;
ylim(gca,[0 1]);
set(gca,'FontSize',14)
lh=legend('show','Location','NorthWest');
set(lh,'Color','white');
end
```

PlotSimilarMaxCovVS

Function used to plot the all Interoperability Coverage values of both P2P and H&S approaches.

```
function PlotSimilarMaxCovVS
%PlotSimilarMaxCovVS Plots the interoperability coverage of the P2P and
%H&S approaches for 25%, 50% and 75% of modifiable systems
modf = [25 50 75];
% Graphic formatting parameters
colors = [0 0 0; 0.7 0.6 0; 1 0 0; 0 0.6 0;...
    0 0 0.4];
markerP2P = '-';
markerHS = '--';
%load of the simulation data
simP2P = cell(1,3);
simHS = loadSimulations( -1, 50,50);
for i=1:3
    simP2P(1,i) = {loadSimulations( modf(1,i), 50, 50)};
end
%% plot interoperability coverage
figure;
[~,nNodes,Coverage,~] = simHS(1,1).getHSData;
eh = plot(nNodes,Coverage(1,:),markerHS);
set(eh,'color',colors(1,:),'LineWidth',3,'DisplayName',' H&S: % to Max Possible');
hold on;
for i=1:3
    simrP2P = simP2P{1,i};
    [~,nNodes,Coverage,~] = simrP2P(1,1).getP2PData;
    eh = plot(nNodes,Coverage(1,:),markerP2P);
    set(eh,'color',colors(1+i,:),'LineWidth',3,'DisplayName',sprintf(' P2P: %% to Max Possible for %d%% Modifiable Systems',modf(1,i)));
    hold on;
end
title('Coverage P2P vs H&S','FontSize',16,'FontWeight','bold');
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
set(get(gca,'Ylabel'),'String','Coverage','FontSize',14);
grid on;
ylim(gca,[0 1.01]);
set(gca,'FontSize',14)
lh=legend('show','Location','SouthEast');
set(lh,'Color','white');
end
```

plotAllData

This function plots all the metrics obtained from the simulations of both P2P and H&S approaches. Each metric is plotted in a different graphic.

```
function plotAllData
%plotAll Plot the NIA, Interoperability Coverage and Absolute
%Interoperability Coverage of obtained in all simulations of both
%interoperability approaches. Each metric is plotted in a separated graphic
[niaP2P,maxCovP2P,absCovP2P,sys] = getP2P;
[niaHS,maxCovHS,absCovHS,~] = getHS;
%% Plot Max Coverage Graphic
figure
% P2P plot
nsize= size(maxCovP2P,1);
hold on
for i=1:nsize
    h=plot(sys,maxCovP2P(i,:));
    set(h,'LineWidth',1.5,'color',[0.59 0.76 0.93]);
    hold on
end
% H&S plot
nsize= size(maxCovHS,1);
hold on
for i=1:nsize
    h=plot(sys,maxCovHS(i,:));
    set(h,'LineWidth',1.5,'color',[0.5529    0.3922    0.3569]);
    hold on
end

ylim(gca,[0 1.01]);
```

```
set(gca,'ytick',(0:0.05:1)');
set(get(gca,'Ylabel'),'String','Coverage','FontSize',14);
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
title('P2P vs H&S Interoperability Coverage','FontSize',16,'FontWeight','bold');
grid on;
%%  Plot Abs Coverage Graphic
figure
% P2P plot
nsize= size(absCovP2P,1);
hold on
for i=1:nsize
   h=plot(sys,absCovP2P(i,:));
   set(h,'LineWidth',1.5,'color',[0.59 0.76 0.93]);
   hold on
end
% H&S plot
nsize= size(absCovHS,1);
hold on
for i=1:nsize
   h=plot(sys,absCovHS(i,:));
   set(h,'LineWidth',1.5,'color',[0.5529    0.3922    0.3569]);
   hold on
end

ylim(gca,[0 1.01]);
set(gca,'ytick',(0:0.05:1)');
set(get(gca,'Ylabel'),'String','Coverage','FontSize',14);
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
title('P2P vs H&S Interoperability Absolute Coverage','FontSize',16,'FontWeight','bold');
grid on;
%% Plot NIA Graphic
figure
% P2P plot
nsize= size(niaP2P,1);
hold on
for i=1:nsize
   h=plot(sys,niaP2P(i,:));
   set(h,'LineWidth',1.5,'color',[0.59 0.76 0.93]);
   hold on
end
% H&S plot
nsize= size(niaHS,1);
hold on
for i=1:nsize
   h=plot(sys,niaHS(i,:));
   set(h,'LineWidth',1.5,'color',[0.5529    0.3922    0.3569]);
   hold on
end

set(get(gca,'Ylabel'),'String','Number of Interoperability Artefacts (NIA)','FontSize',14);
set(get(gca,'Xlabel'),'String','Number of systems','FontSize',14);
title('P2P vs H&S Number of Interoperability Artefacts (NIA)','FontSize',16,'FontWeight','bold');
grid on;
end
```

getP2P

This function is used to get the data of all P2P Interoperability simulations.

```
function [Nia,maxCov,absCov,nodes] = getP2P
%getP2P Get the data obtained in all simulations of the P2P approach
expImp = [25 75; 50 50; 50 75; 75 75];
nEI = size(expImp,1);
modf = [25 50 75];
nModf = size(modf,2);
tSim = nEI * nModf * 3;% 3 corresponds to the percentage of systems with equal data formats
sim = cell(nEI,nModf);
for i=1:nEI
   for j=1:nModf
      sim(i,j) = {loadSimulations( (modf(1,j)), expImp(i,1), expImp(i,2))};
   end
end
simP2P = sim{1,1};
[~,nodes,~,~] = simP2P(1,1).getP2PData;
nNodes = size(nodes,2);
absCov=zeros(tSim,nNodes);
maxCov=zeros(tSim,nNodes);
Nia=zeros(tSim,nNodes);
for i=1:nEI
   for j=1:nModf
      for eq=1:3
         simP2P = sim{i,j};
         [nia,~,maxC,absC] = simP2P(1,eq).getP2PData;
         nSimulation = (9*(i-1))+(3*(j-1))+eq;
         absCov(nSimulation,:) = absC(1,:);
         maxCov(nSimulation,:) = maxC(1,:);
         Nia(nSimulation,:) = nia(1,:);
      end
   end
end
end
```

getHS

This function is used to get the data of all H&S Interoperability simulations.

```
function [Nia,maxCov,absCov,nodes] = getHS
%getHS Get the data obtained in all simulations of the H&S approach
expImp = [25 75; 50 50; 50 75; 75 75];
nEI = size(expImp,1);
sim = cell(nEI,1);
for i=1:nEI
    sim(i,1) = {loadSimulations( -1, expImp(i,1), expImp(i,2))};
end
simHS = sim{1,1};
[~,nodes,~,~] = simHS(1,1).getHSData;
nNodes = size(nodes,2);
absCov = zeros(nEI,nNodes);
maxCov = zeros(nEI,nNodes);
Nia = zeros(nEI * 3,nNodes);
for i=1:nEI
    simHS = sim{i,1};
    [nia,~,maxC,absC] = simHS(1,1).getHSData;
    absCov(i,:) = absC(1,:);
    maxCov(i,:) = maxC(1,:);
    Nia((3*(i-1))+1,:) = nia(1,:);
    for eq=2:3
        [nia,~,~,~] = simHS(1,eq).getHSData;
        Nia((3*(i-1))+eq,:) = nia(1,:);
    end
end
end
```

Appendix B:
P2P Interoperability Simulation Graphics

In this appendix are presented the 54 graphics obtained as result from the simulations of the P2P Interoperability approach. These graphics result from the combination of five parameters: number of systems, percentage of modifiable systems, percentage of systems with same formats, percentage of systems with export capabilities and percentage of systems with import capabilities. The number of systems varies from 5 to 115 with a step of five. All the other four parameters assume the values 25%, 50% and 75%, being performed all the possible combinations, assuring that the sum of the percentage of Exporters and Importers is always equal or greater than 100%. The variation of parameters and corresponding graphics are identified in the table below.

Exporters/Importers (%)	Systems with Same Formats (%)								
	25	50	75	25	50	75	25	50	75
25/75	Image 4	Image 10	Image 16	Image 22	Image 28	Image 34	Image 40	Image 46	Image 52
50/50	Image 5	Image 11	Image 17	Image 23	Image 29	Image 35	Image 41	Image 47	Image 53
50/75	Image 6	Image 12	Image 18	Image 24	Image 30	Image 36	Image 42	Image 48	Image 54
75/25	Image 7	Image 13	Image 19	Image 25	Image 31	Image 37	Image 43	Image 49	Image 55
75/50	Image 8	Image 14	Image 20	Image 26	Image 32	Image 38	Image 44	Image 50	Image 56
75/75	Image 9	Image 15	Image 21	Image 27	Image 33	Image 39	Image 45	Image 51	Image 57
	25	25	25	50	50	50	75	75	75
	Modifiable Systems (%)								

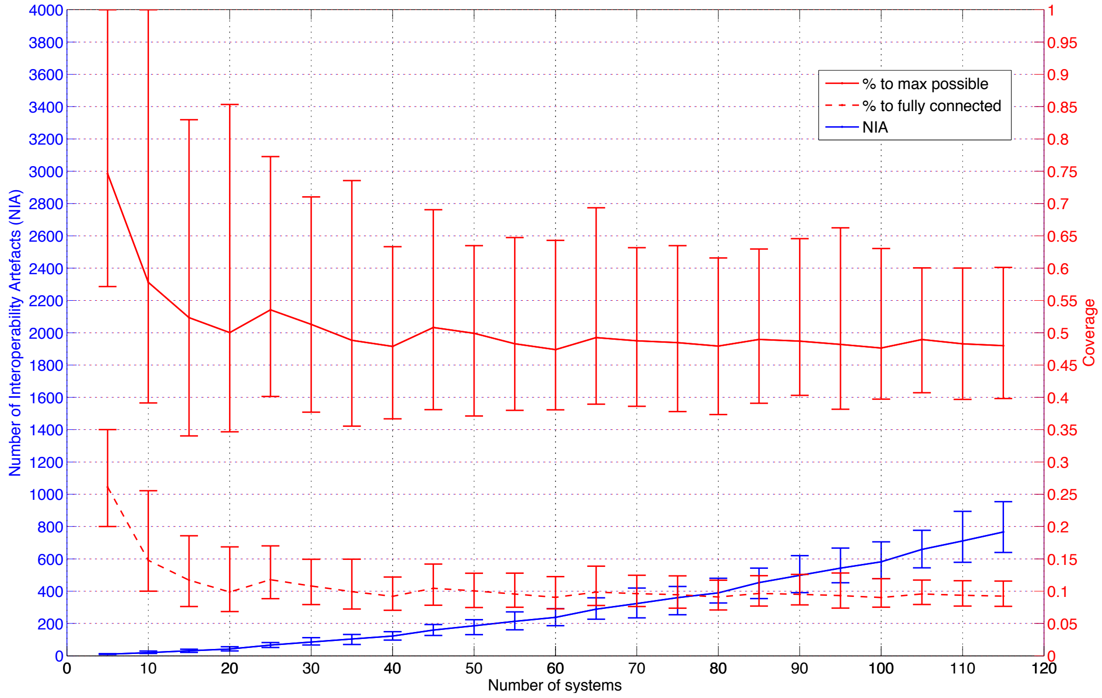


Image 4: Simulation result for 25% of modifiable systems, 25% of same formats, 25% of exporters and 75% of importers

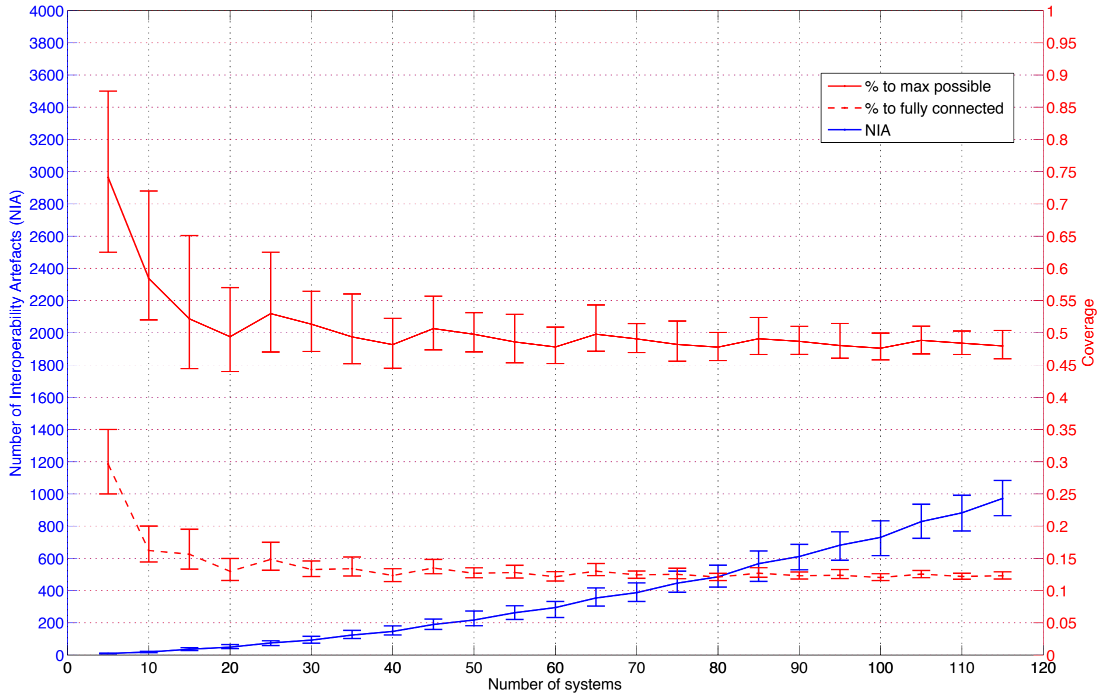


Image 5: Simulation result for 25% of modifiable systems, 25% of same formats, 50% of exporters and 50% of importers

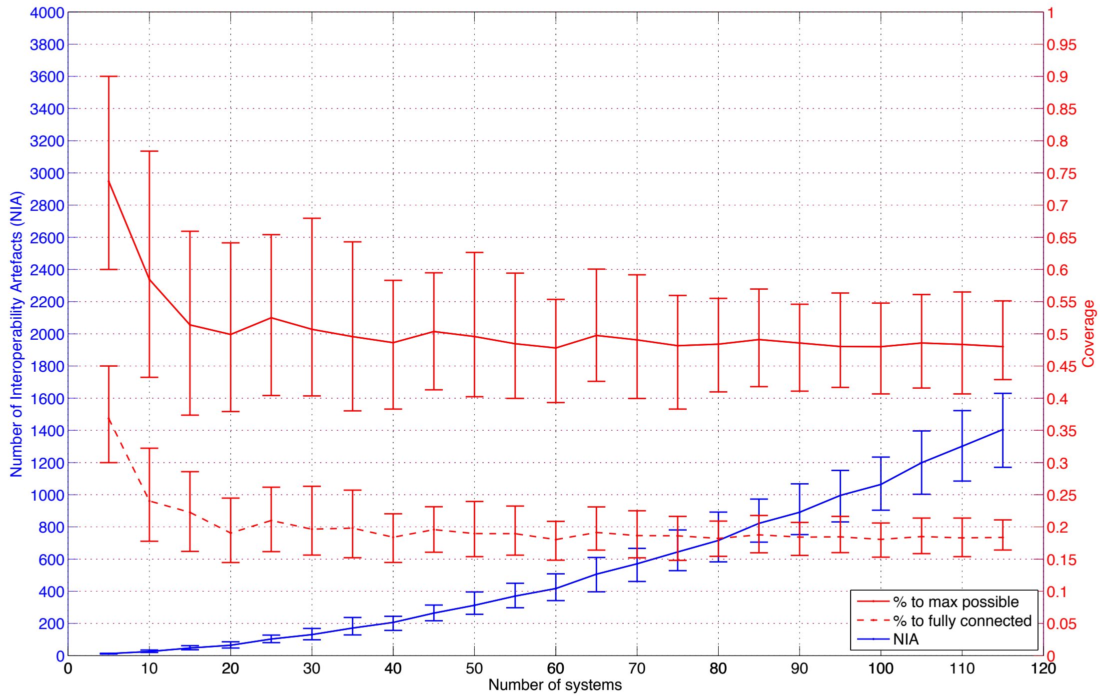


Image 6: Simulation result for 25% of modifiable systems, 25% of same formats, 50% of exporters and 75% of importers

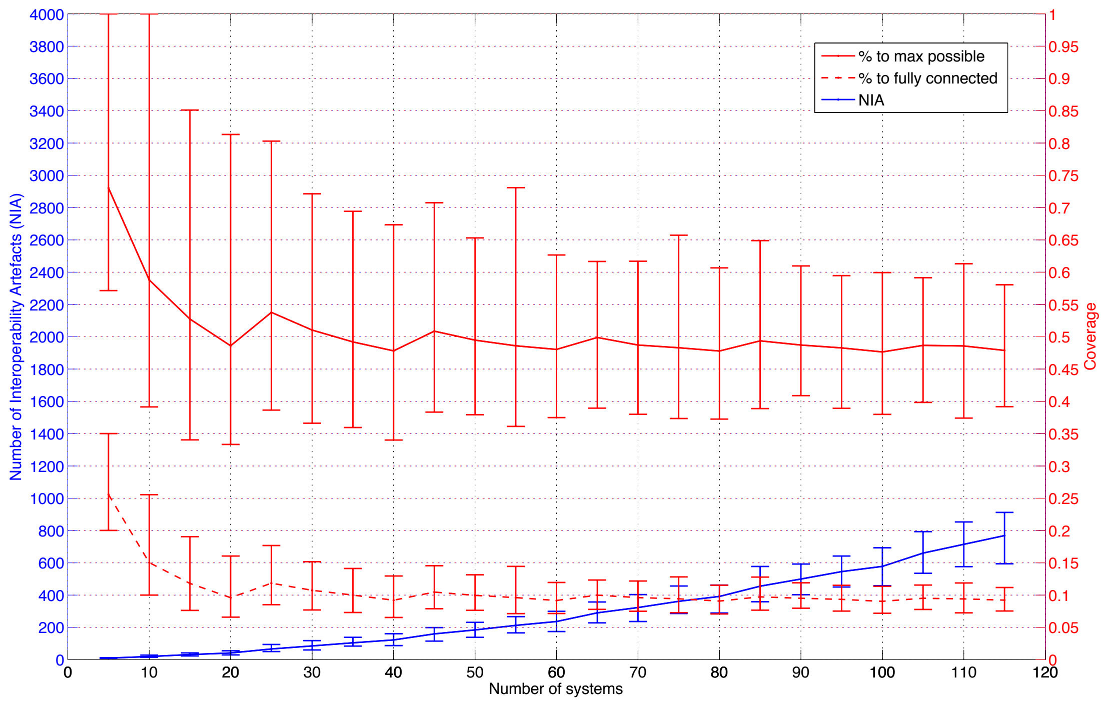


Image 7: Simulation result for 25% of modifiable systems, 25% of same formats, 75% of exporters and 25% of importers

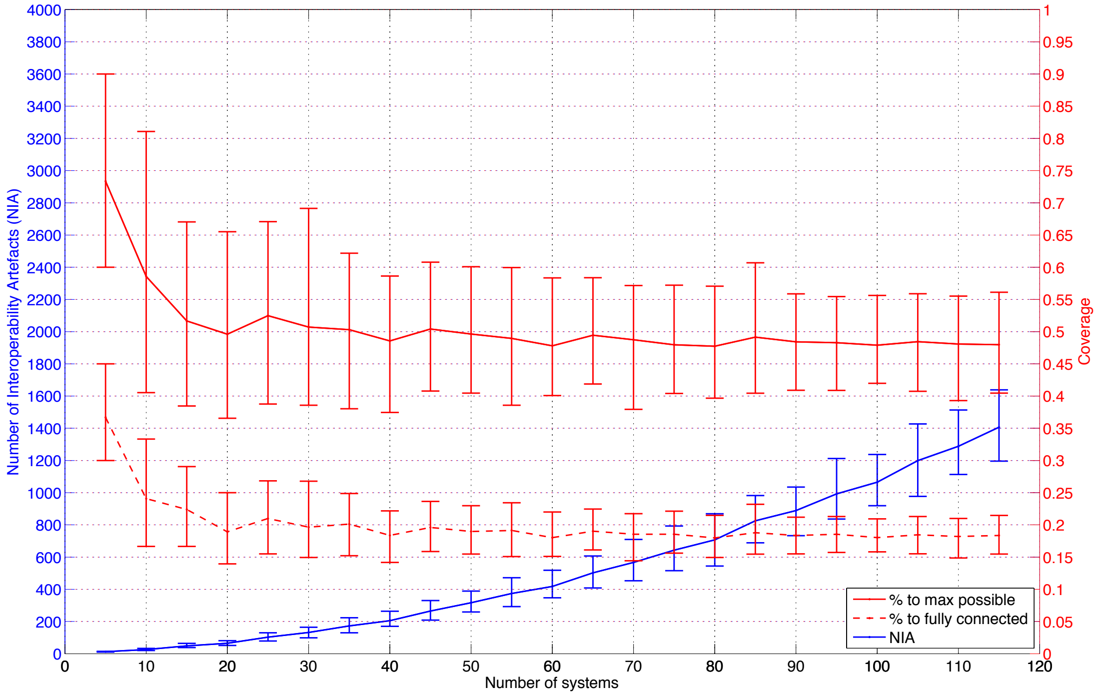


Image 8: Simulation result for 25% of modifiable systems, 25% of same formats, 75% of exporters and 50% of importers

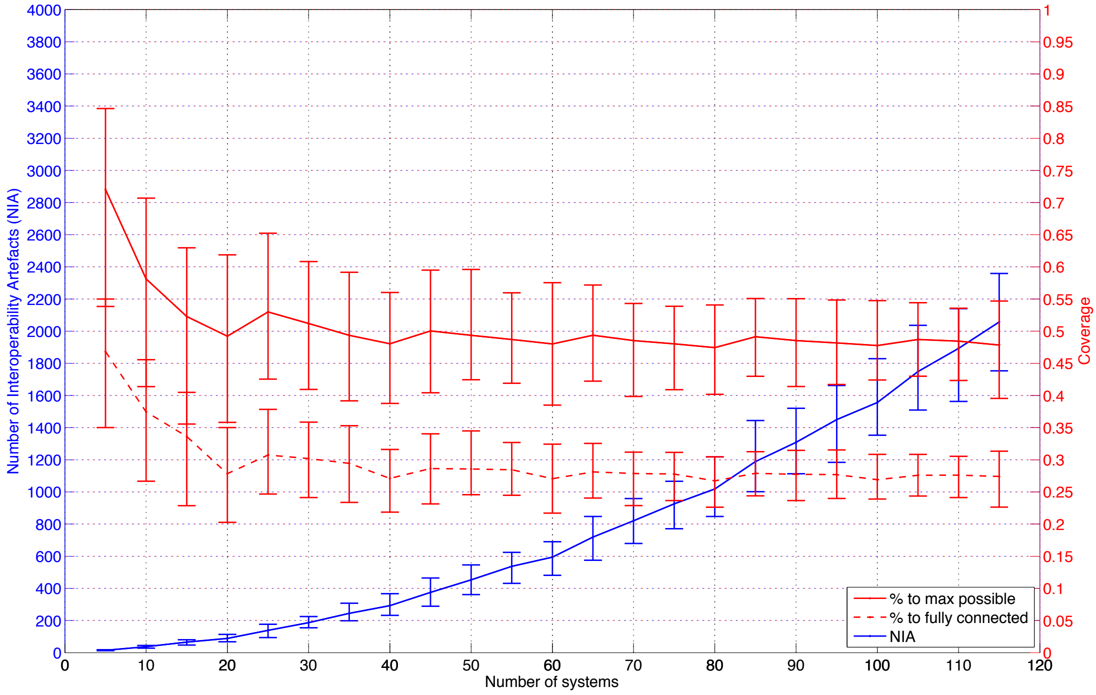


Image 9: Simulation result for 25% of modifiable systems, 25% of same formats, 75% of exporters and 75% of importers

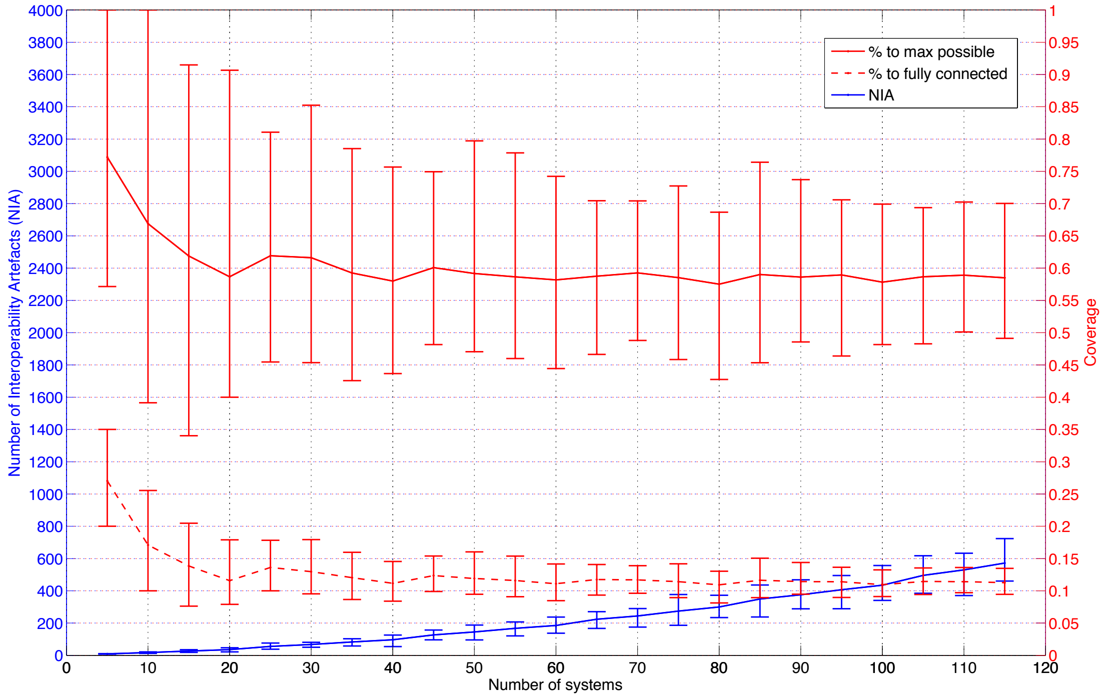


Image 10: Simulation result for 25% of modifiable systems, 50% of same formats, 25% of exporters and 75% of importers

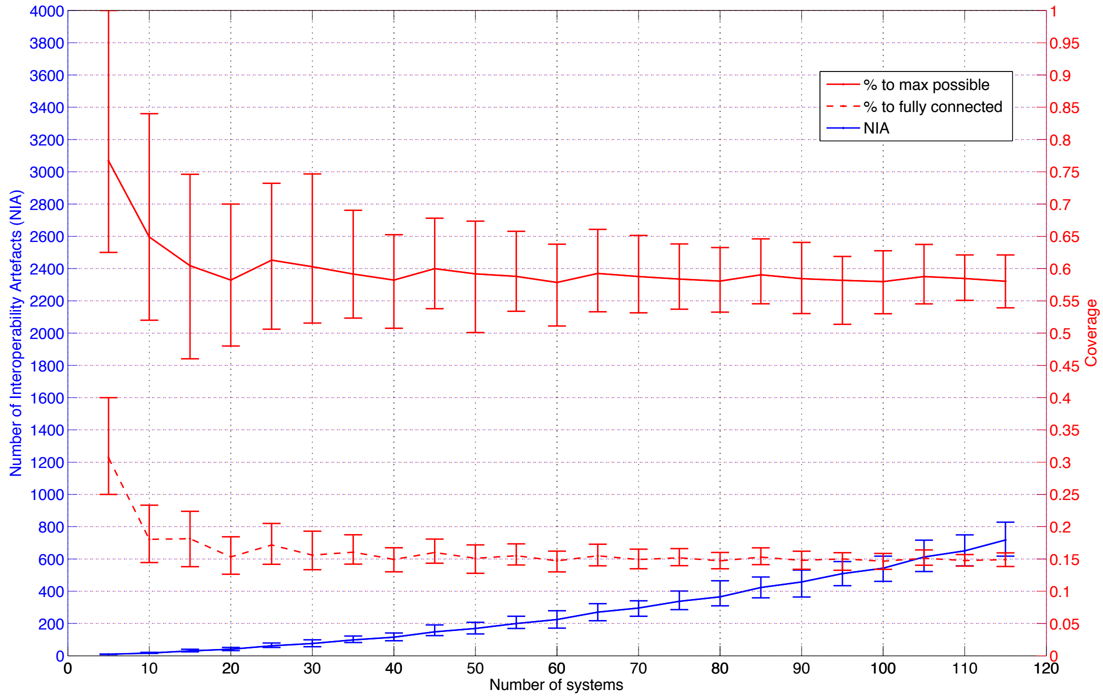


Image 11: Simulation result for 25% of modifiable systems, 50% of same formats, 50% of exporters and 50% of importers

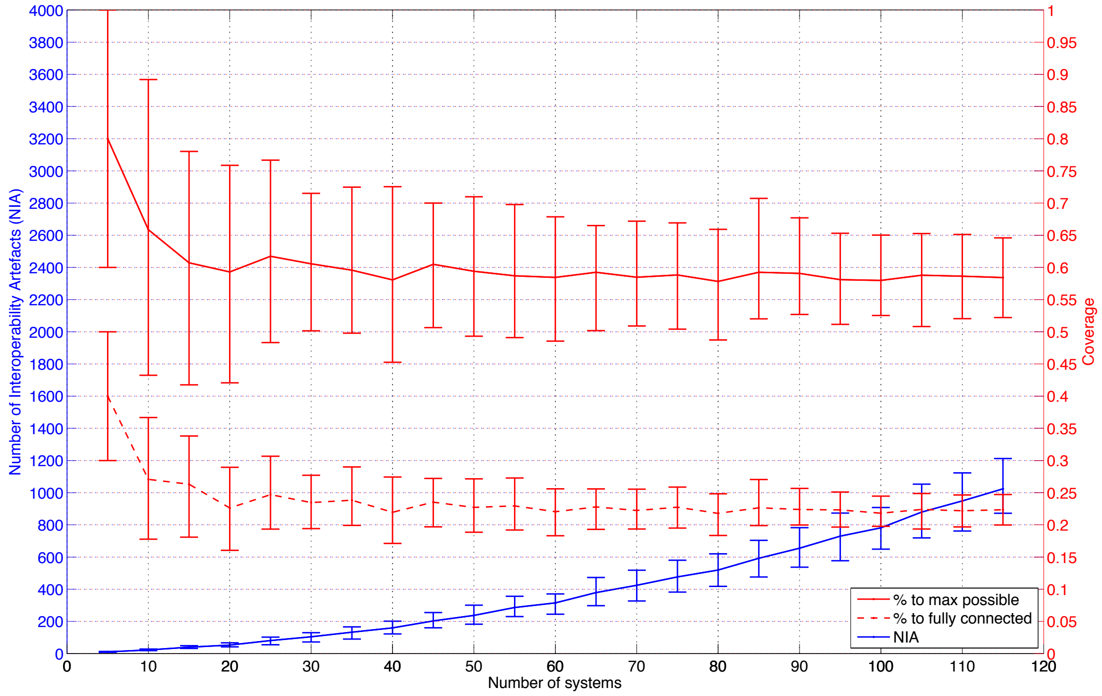


Image 12: Simulation result for 25% of modifiable systems, 50% of same formats, 50% of exporters and 75% of importers

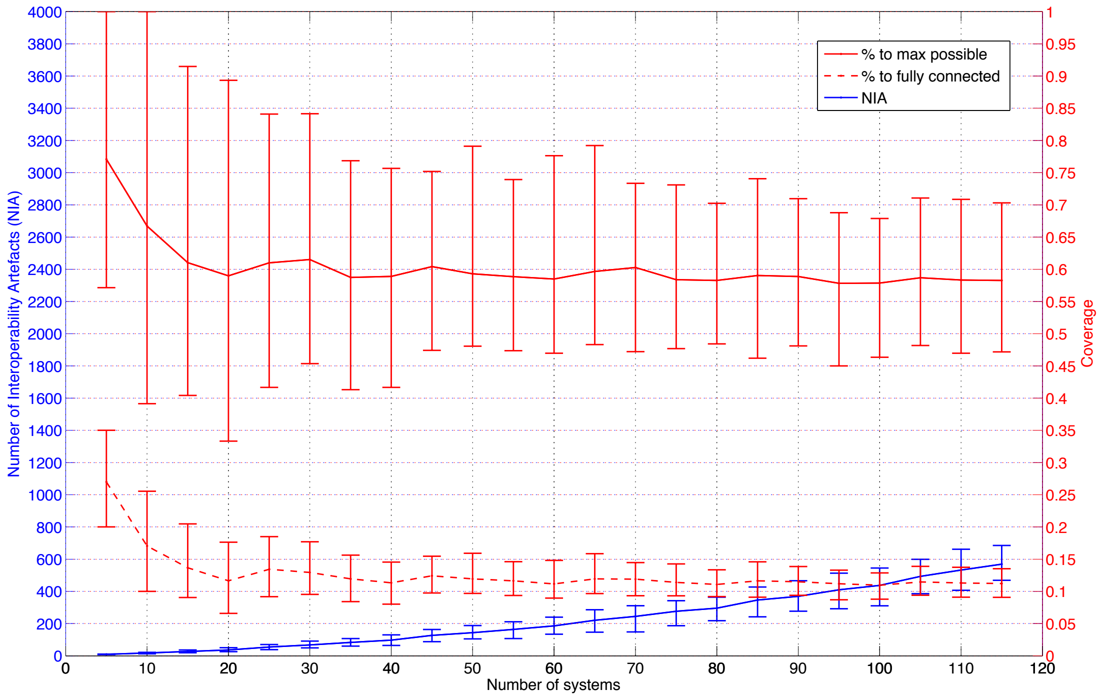


Image 13: Simulation result for 25% of modifiable systems, 50% of same formats, 75% of exporters and 25% of importers

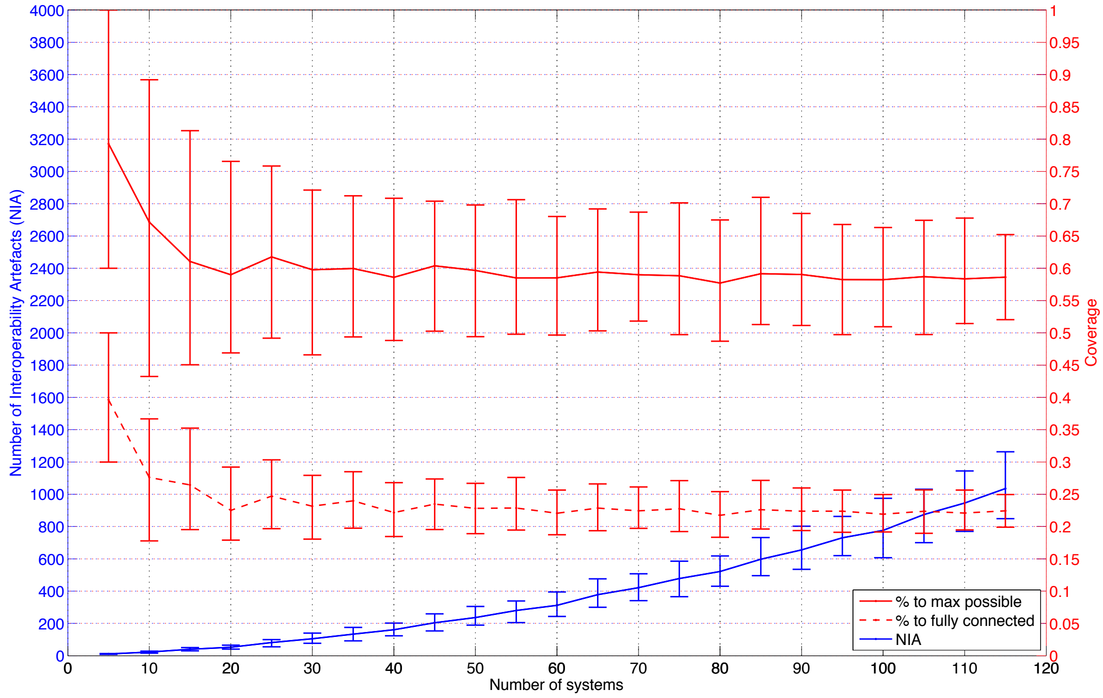


Image 14: Simulation result for 25% of modifiable systems, 50% of same formats, 75% of exporters and 50% of importers

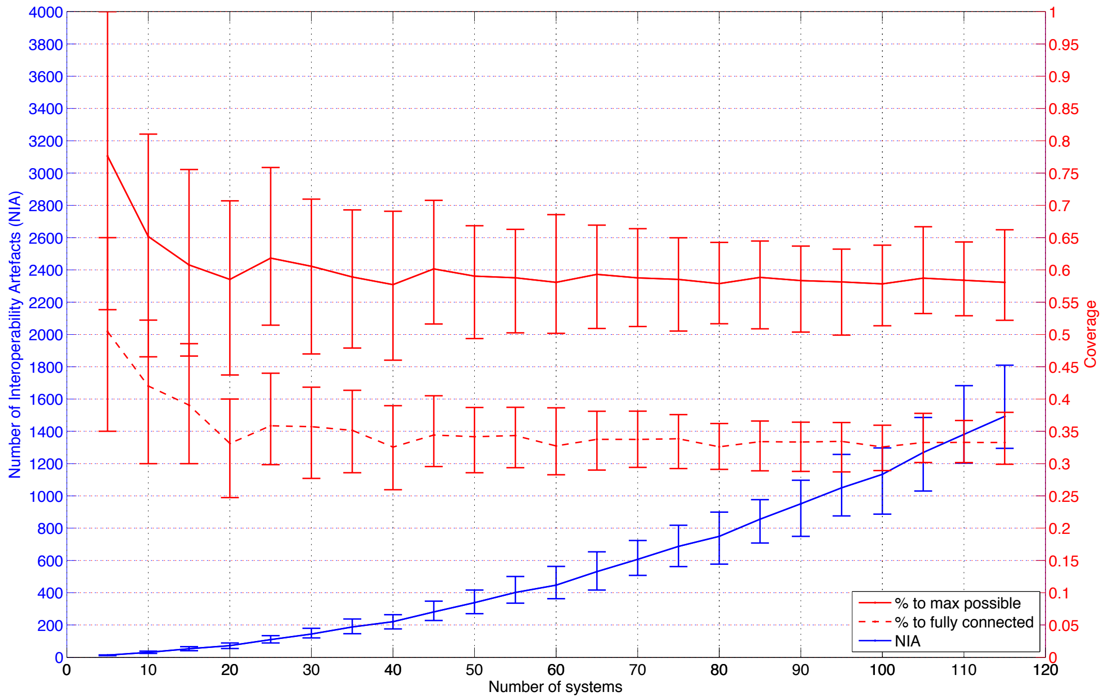


Image 15: Simulation result for 25% of modifiable systems, 50% of same formats, 75% of exporters and 75% of importers

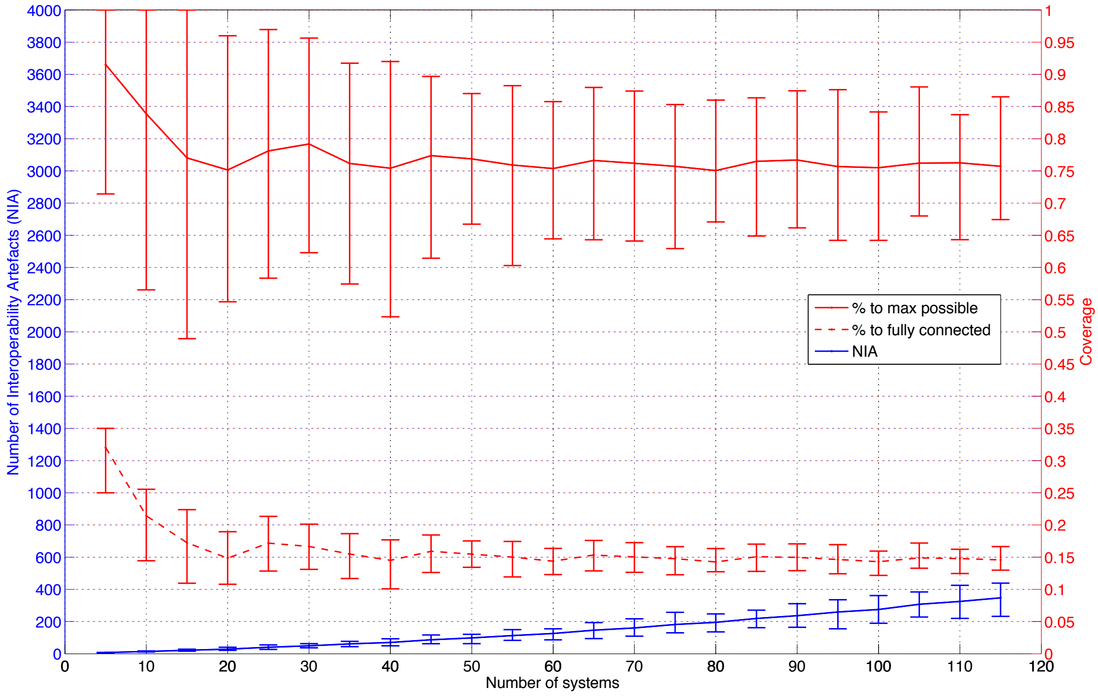


Image 16: Simulation result for 25% of modifiable systems, 75% of same formats, 25% of exporters and 75% of importers

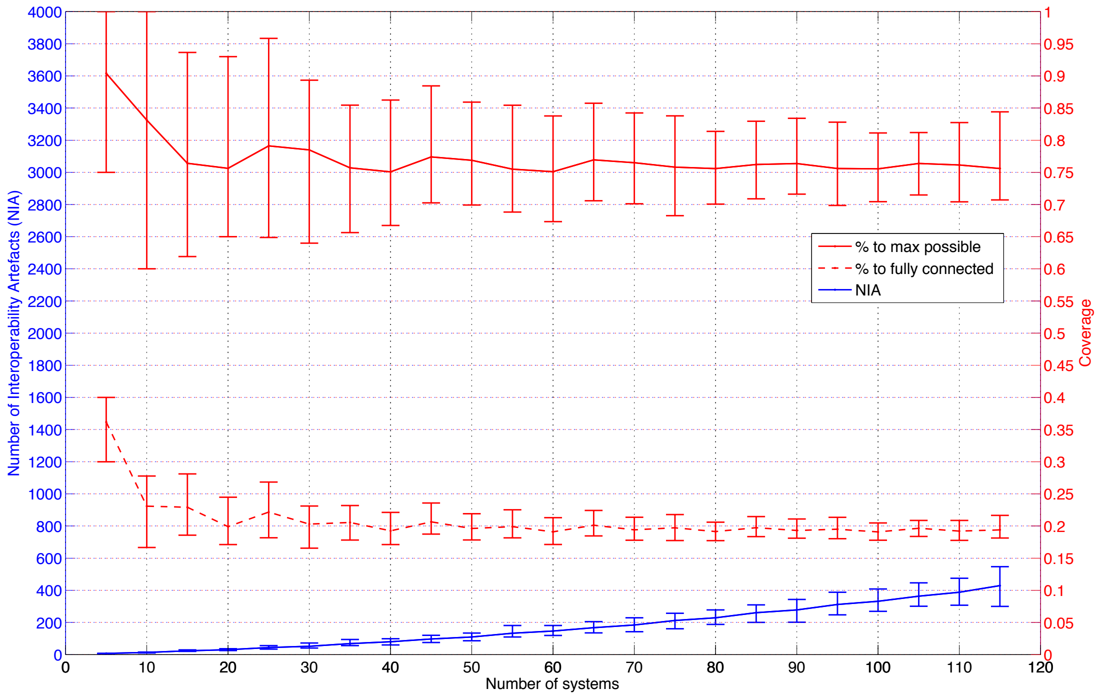


Image 17: Simulation result for 25% of modifiable systems, 75% of same formats, 50% of exporters and 50% of importers

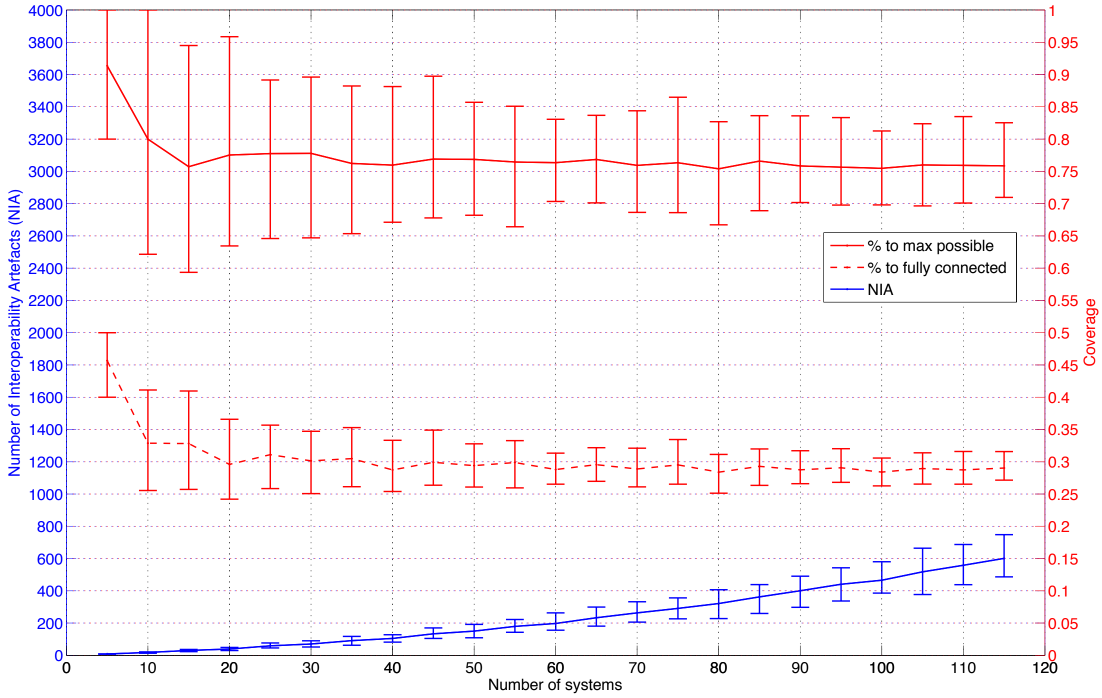


Image 18: Simulation result for 25% of modifiable systems, 75% of same formats, 50% of exporters and 75% of importers

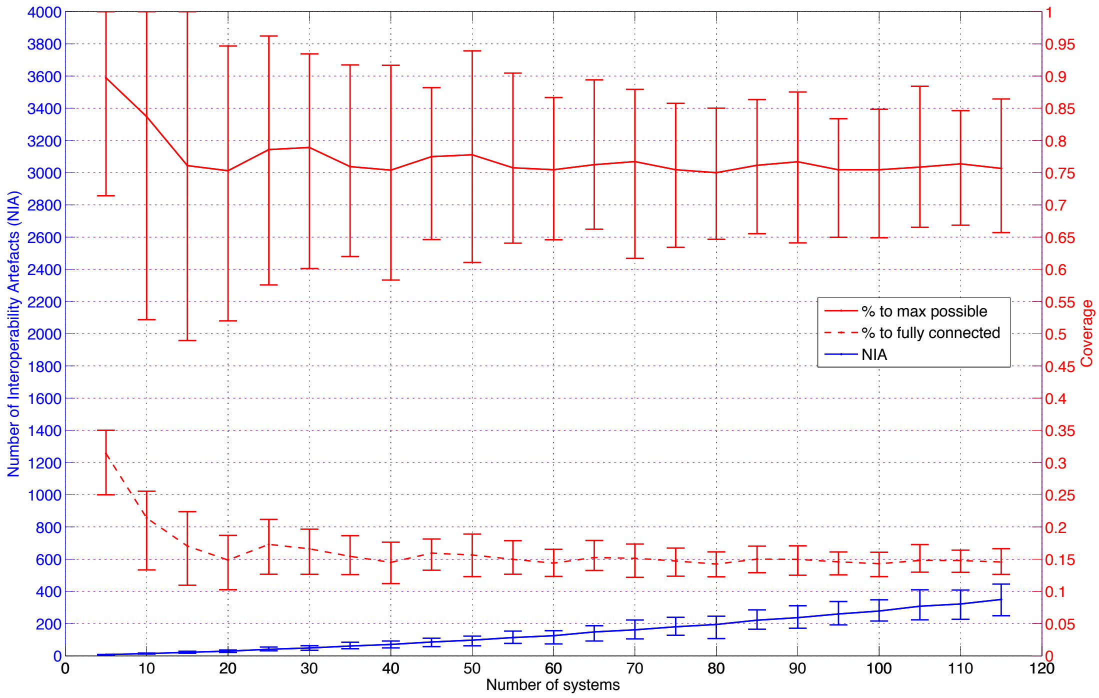


Image 19: Simulation result for 25% of modifiable systems, 75% of same formats, 75% of exporters and 25% of importers

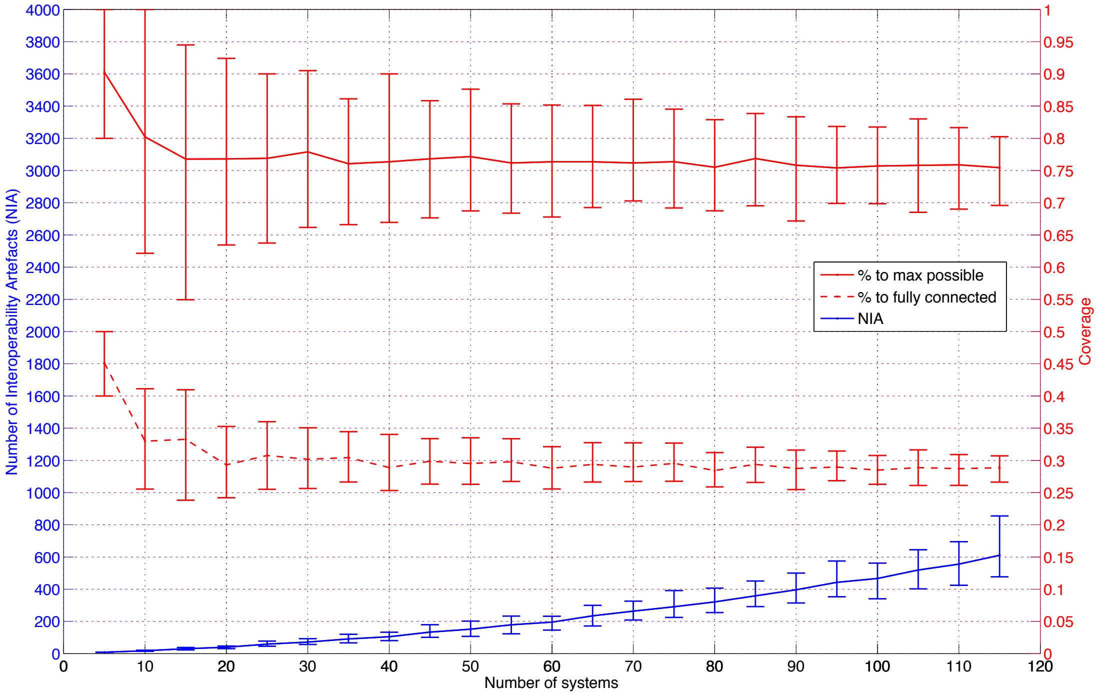


Image 20: Simulation result for 25% of modifiable systems, 75% of same formats, 75% of exporters and 50% of importers

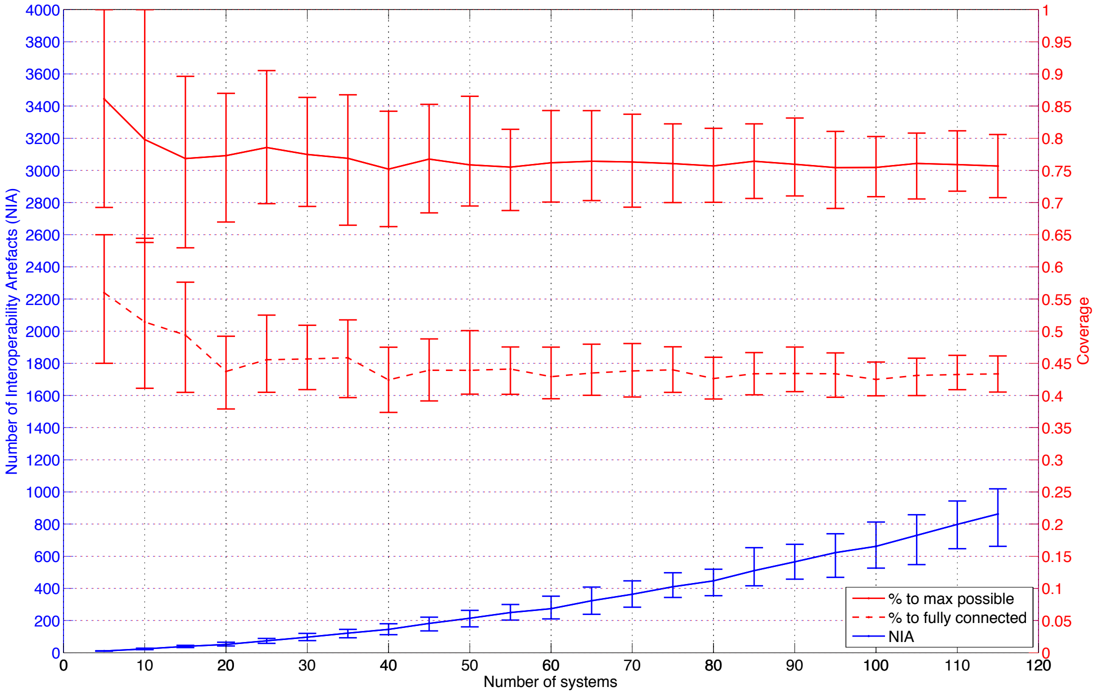


Image 21: Simulation result for 25% of modifiable systems, 75% of same formats, 75% of exporters and 75% of importers

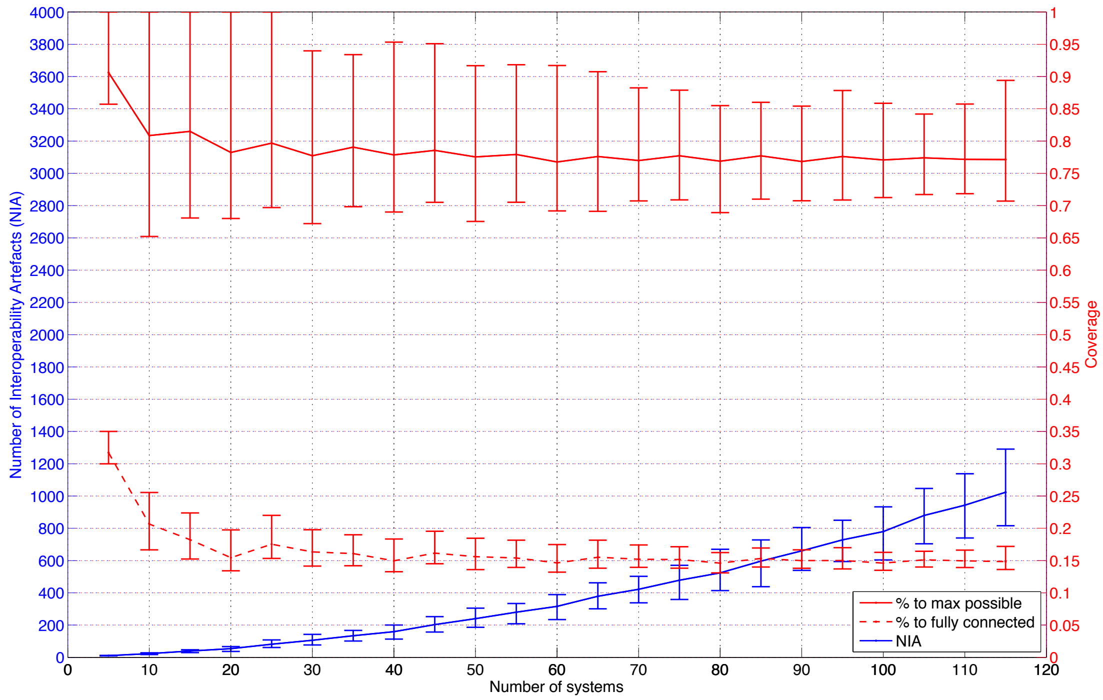


Image 22: Simulation result for 50% of modifiable systems, 25% of same formats, 25% of exporters and 75% of importers

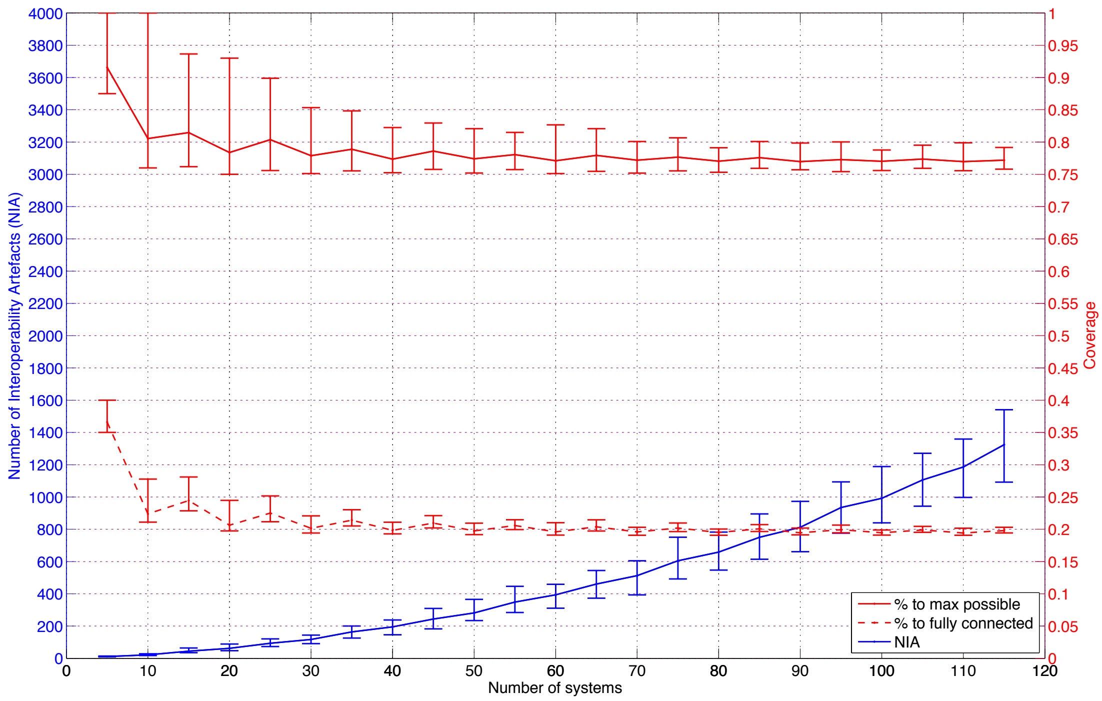


Image 23: Simulation result for 50% of modifiable systems, 25% of same formats, 50% of exporters and 50% of importers

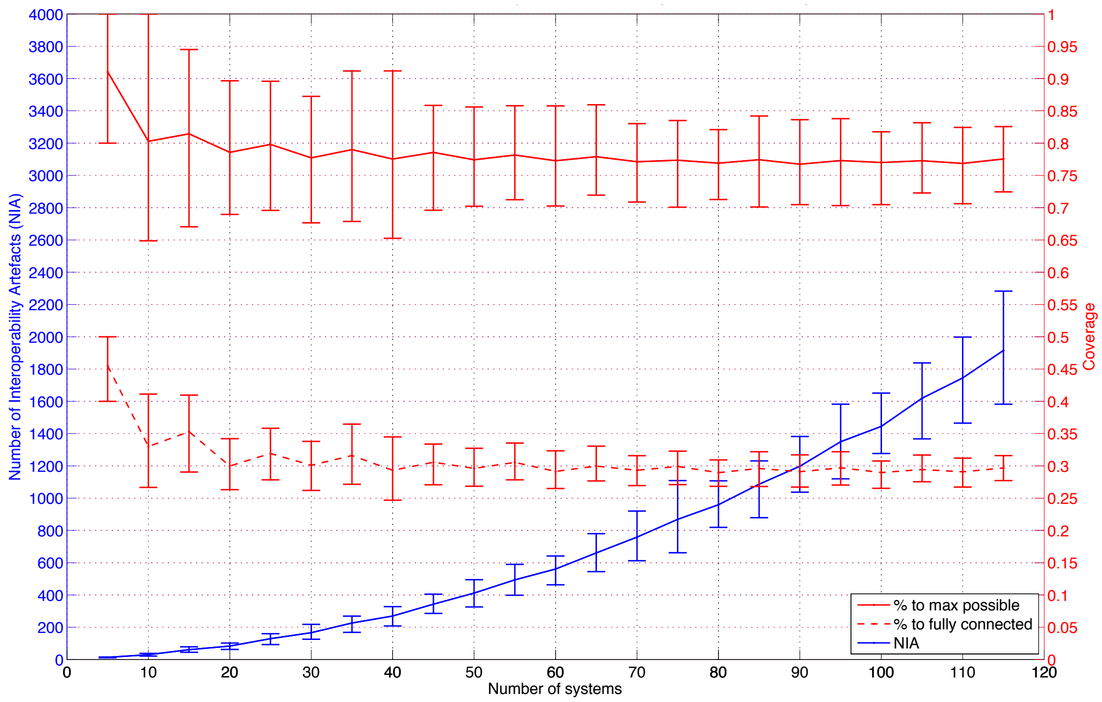


Image 24: Simulation result for 50% of modifiable systems, 25% of same formats, 50% of exporters and 75% of importers

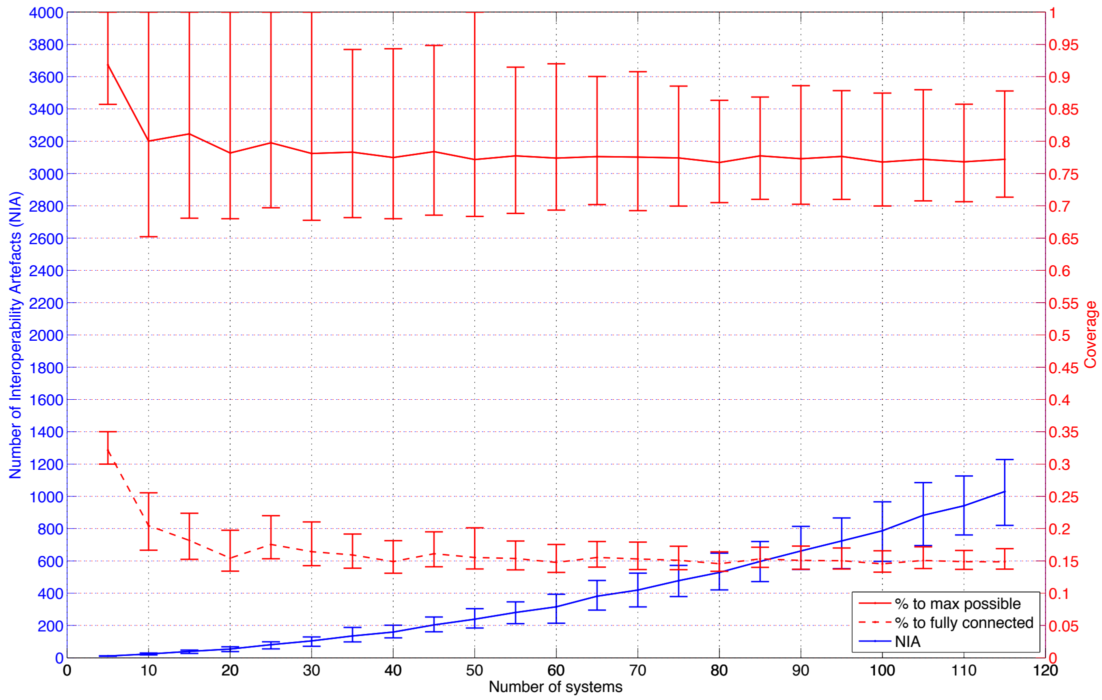


Image 25: Simulation result for 50% of modifiable systems, 25% of same formats, 75% of exporters and 25% of importers

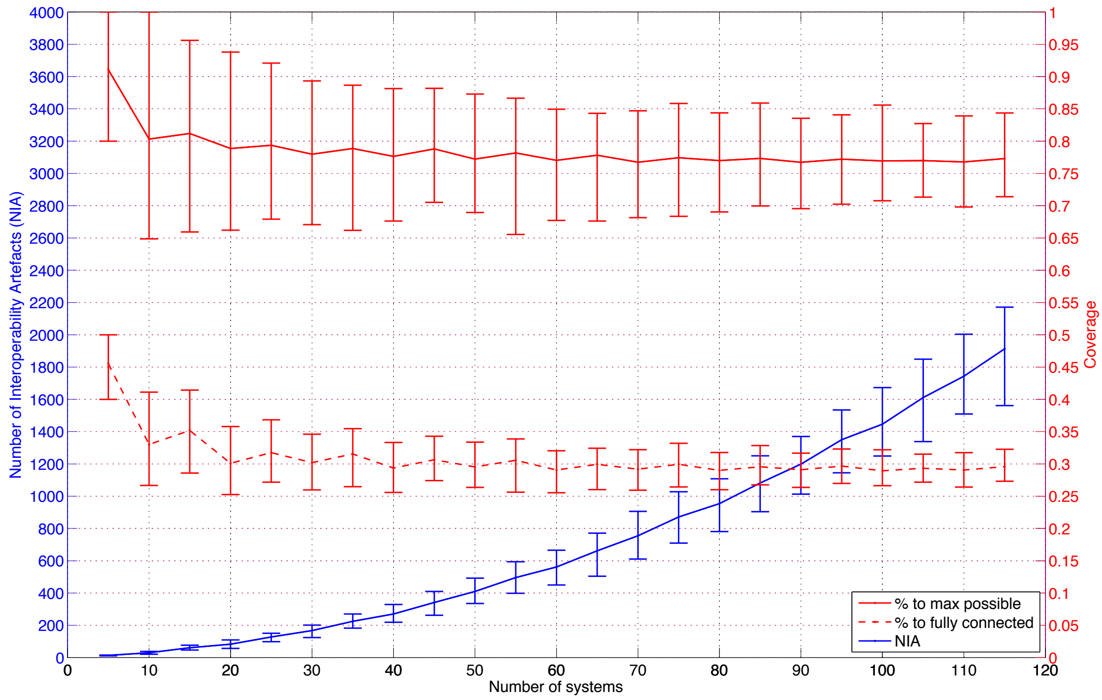


Image 26: Simulation result for 50% of modifiable systems, 25% of same formats, 75% of exporters and 50% of importers

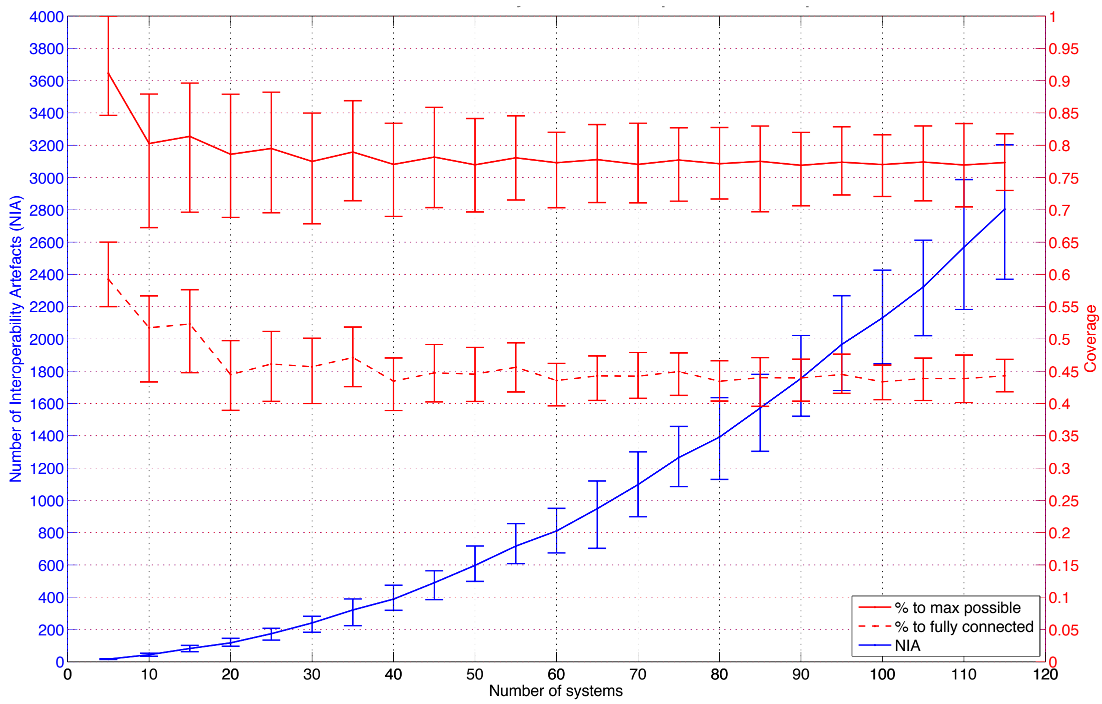


Image 27: Simulation result for 50% of modifiable systems, 25% of same formats, 75% of exporters and 75% of importers

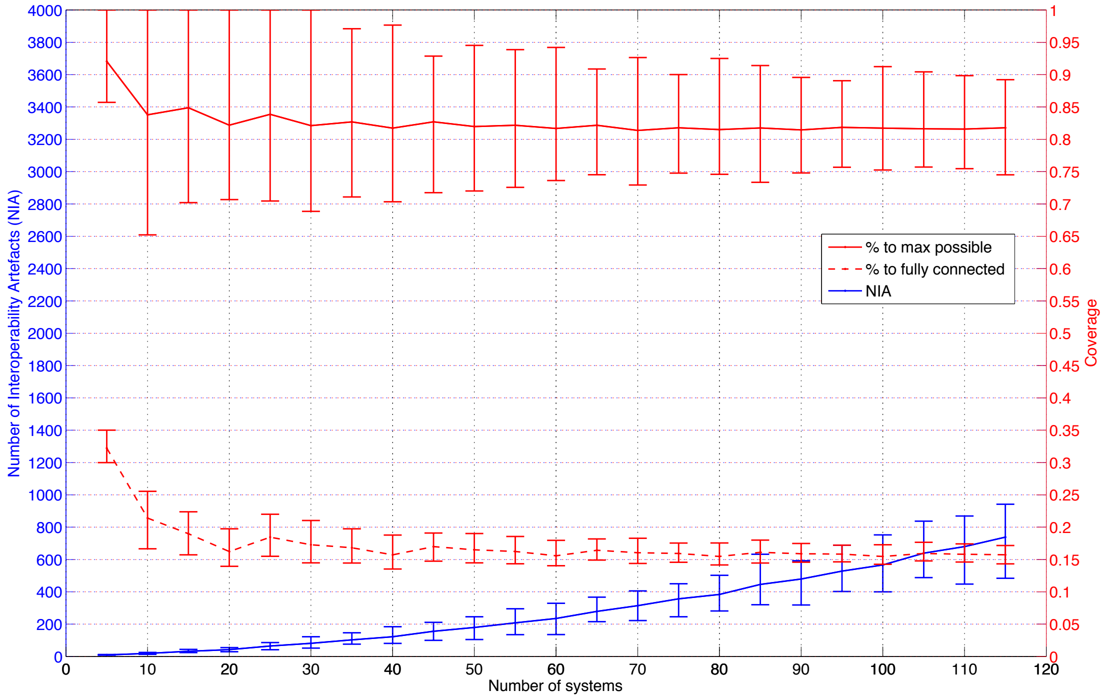


Image 28: Simulation result for 50% of modifiable systems, 50% of same formats, 25% of exporters and 75% of importers

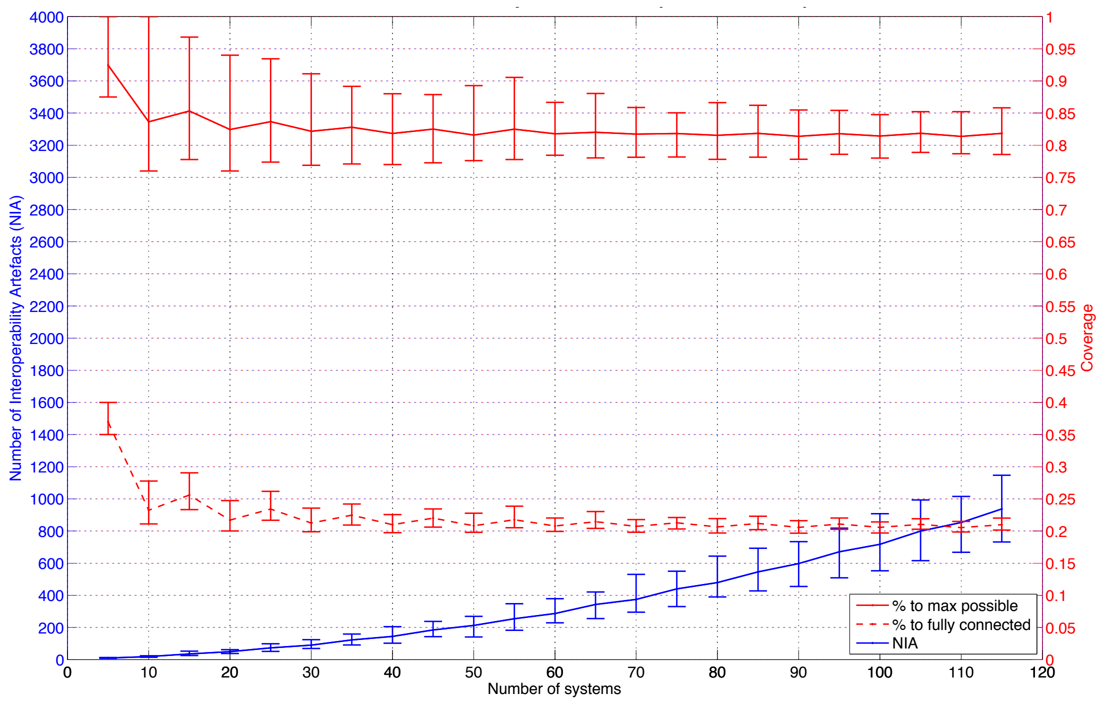


Image 29: Simulation result for 50% of modifiable systems, 50% of same formats, 50% of exporters and 50% of importers

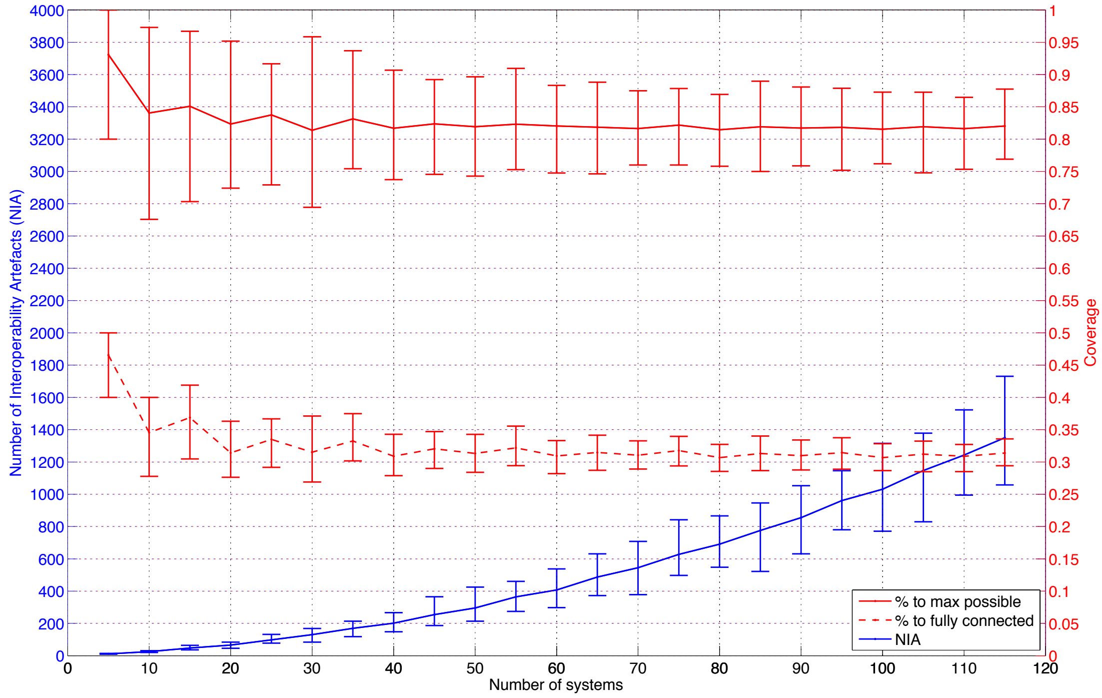


Image 30: Simulation result for 50% of modifiable systems, 50% of same formats, 50% of exporters and 75% of importers

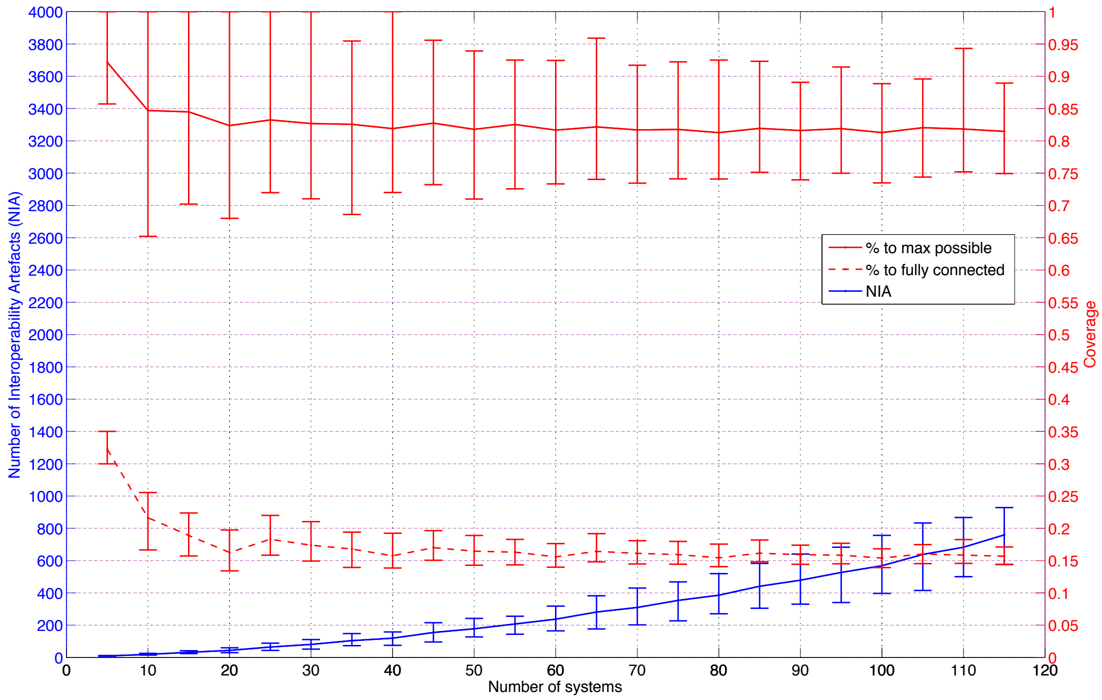


Image 31: Simulation result for 50% of modifiable systems, 50% of same formats, 75% of exporters and 25% of importers

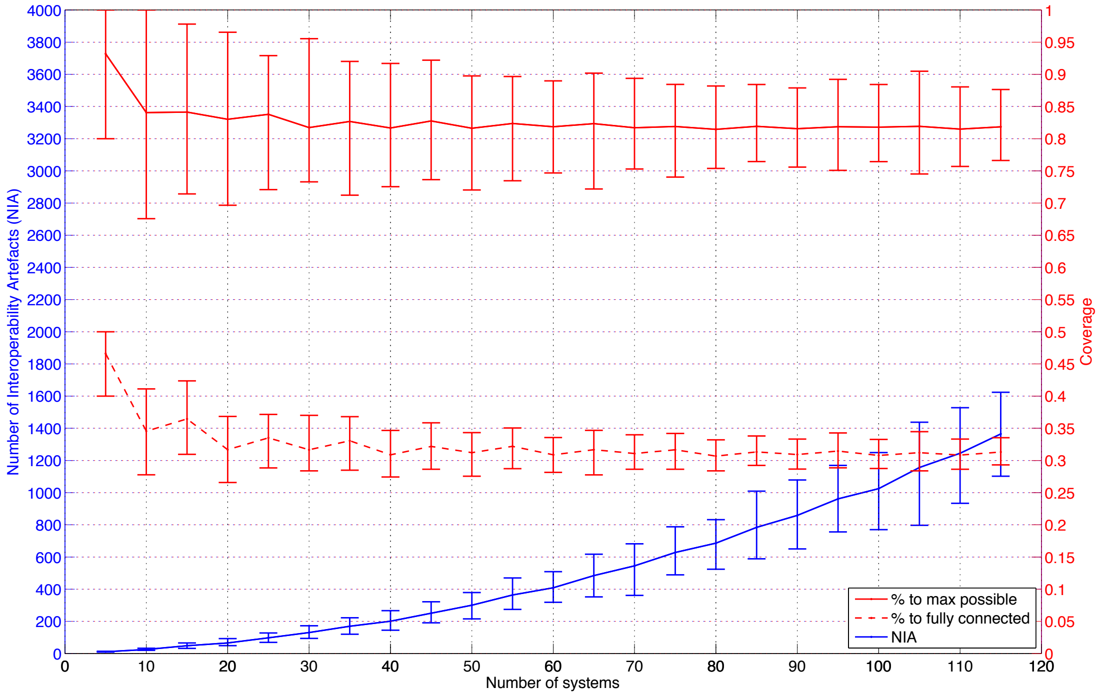


Image 32: Simulation result for 50% of modifiable systems, 50% of same formats, 75% of exporters and 50% of importers

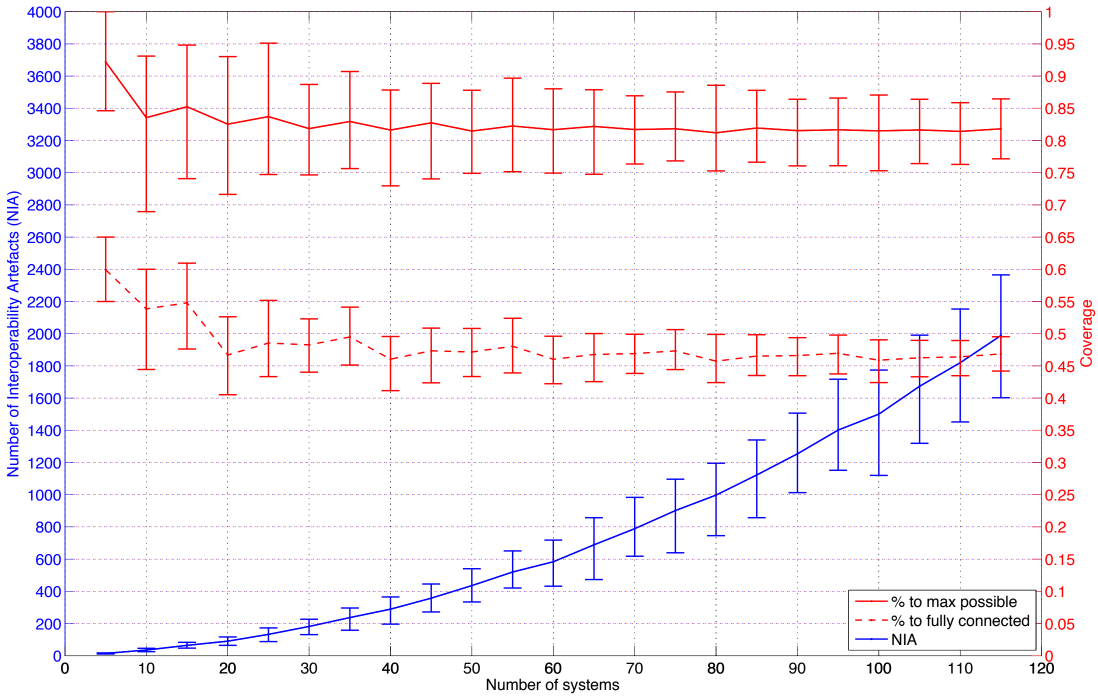


Image 33: Simulation result for 50% of modifiable systems, 50% of same formats, 75% of exporters and 75% of importers

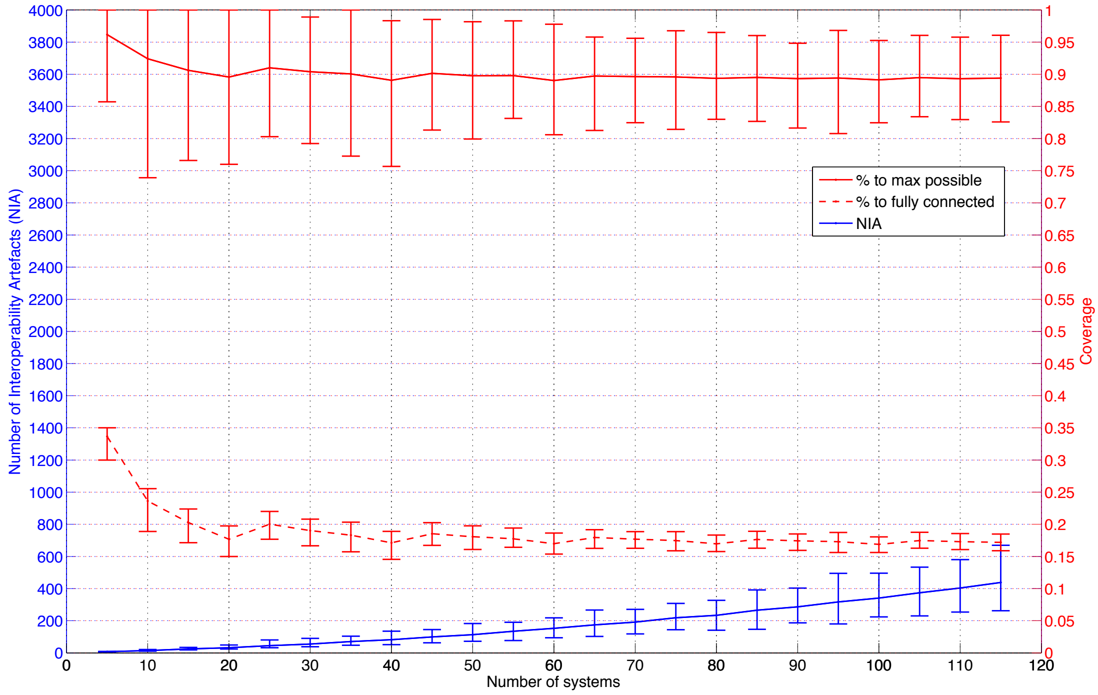


Image 34: Simulation result for 50% of modifiable systems, 75% of same formats, 25% of exporters and 75% of importers

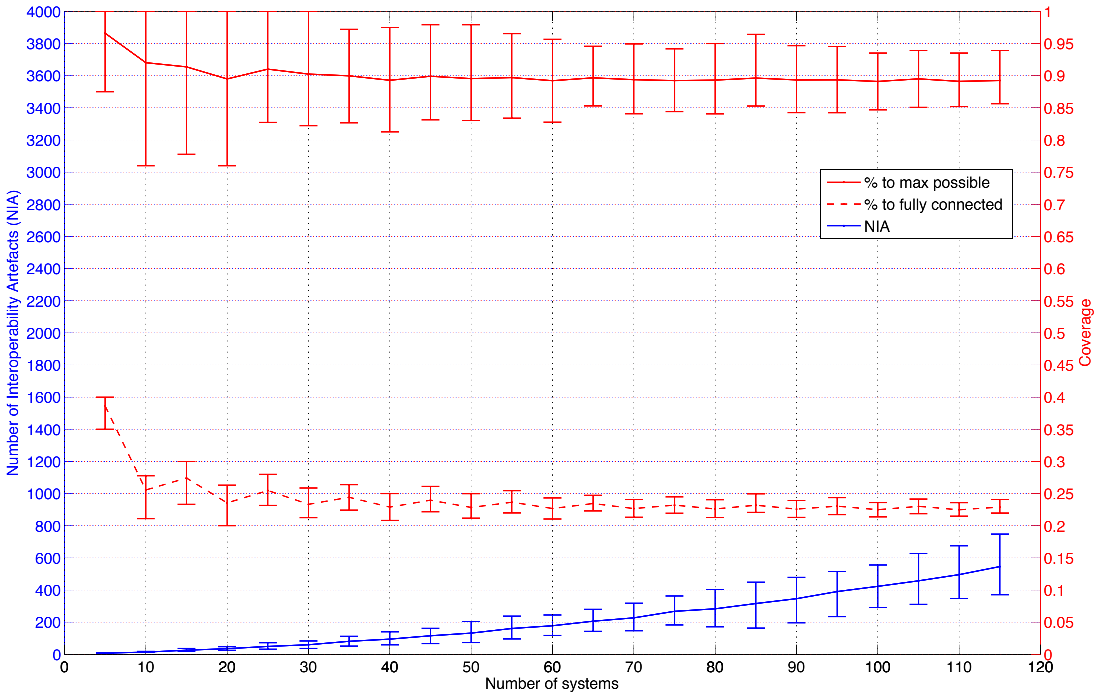


Image 35: Simulation result for 50% of modifiable systems, 75% of same formats, 50% of exporters and 50% of importers

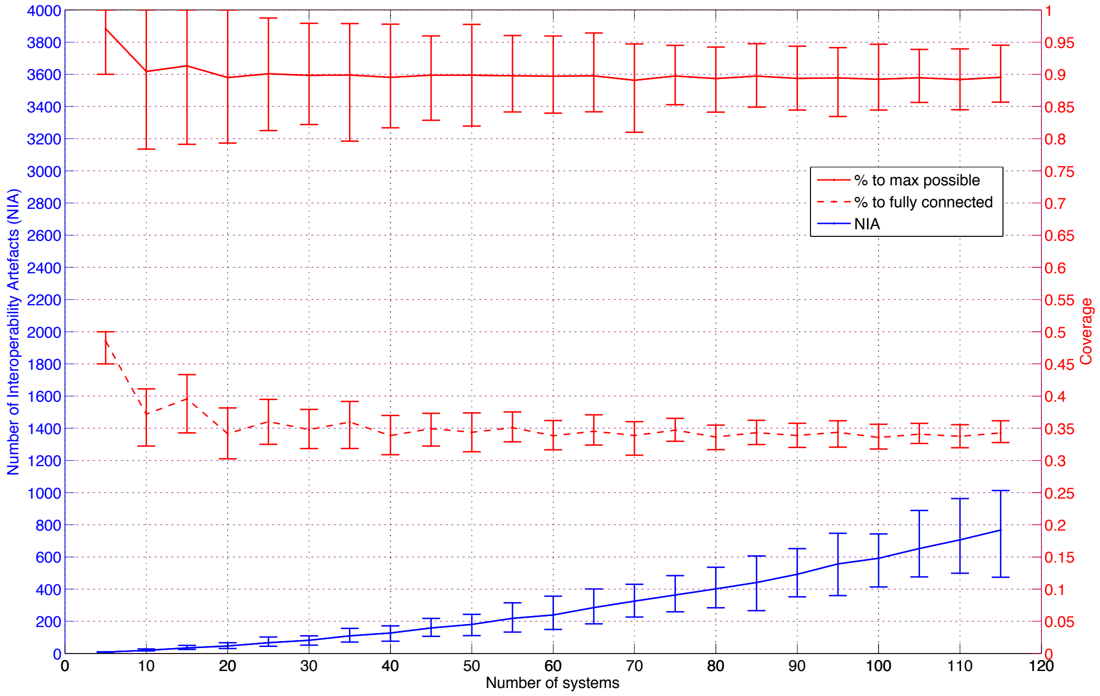


Image 36: Simulation result for 50% of modifiable systems, 75% of same formats, 50% of exporters and 75% of importers

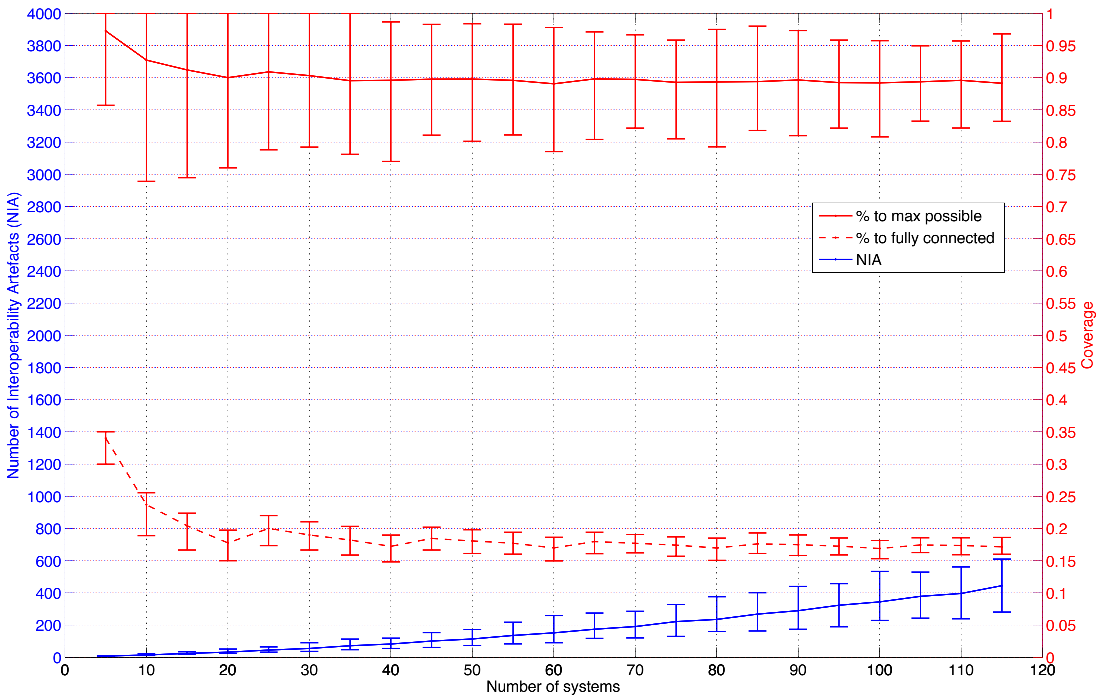


Image 37: Simulation result for 50% of modifiable systems, 75% of same formats, 75% of exporters and 25% of importers

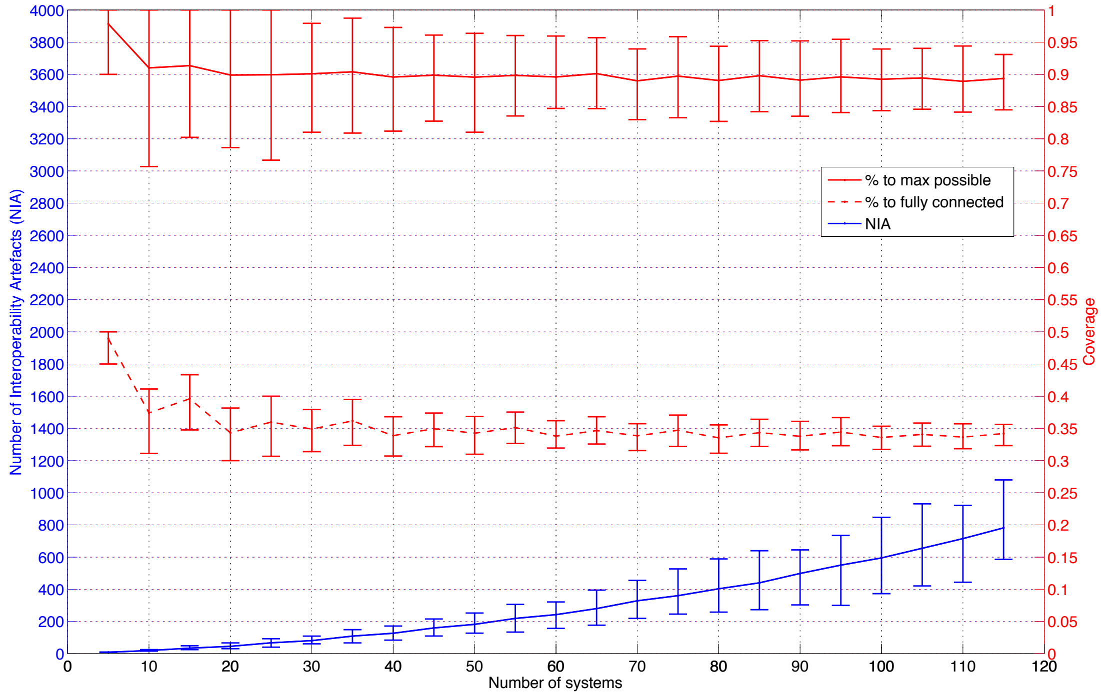


Image 38: Simulation result for 50% of modifiable systems, 75% of same formats, 75% of exporters and 50% of importers

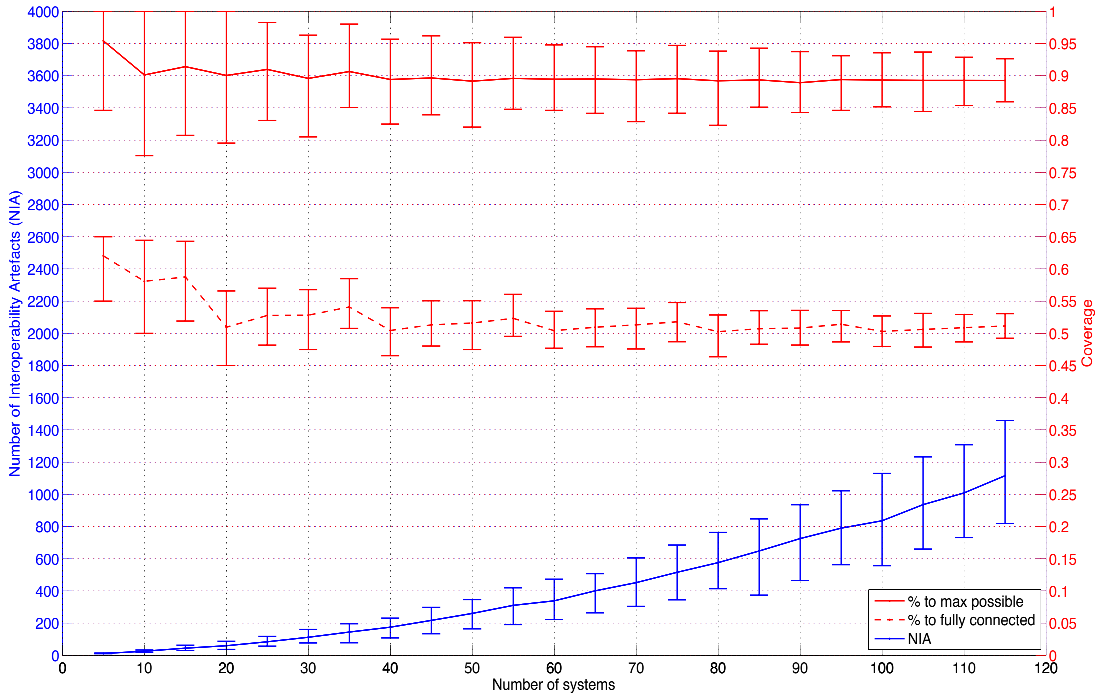


Image 39: Simulation result for 50% of modifiable systems, 75% of same formats, 75% of exporters and 75% of importers

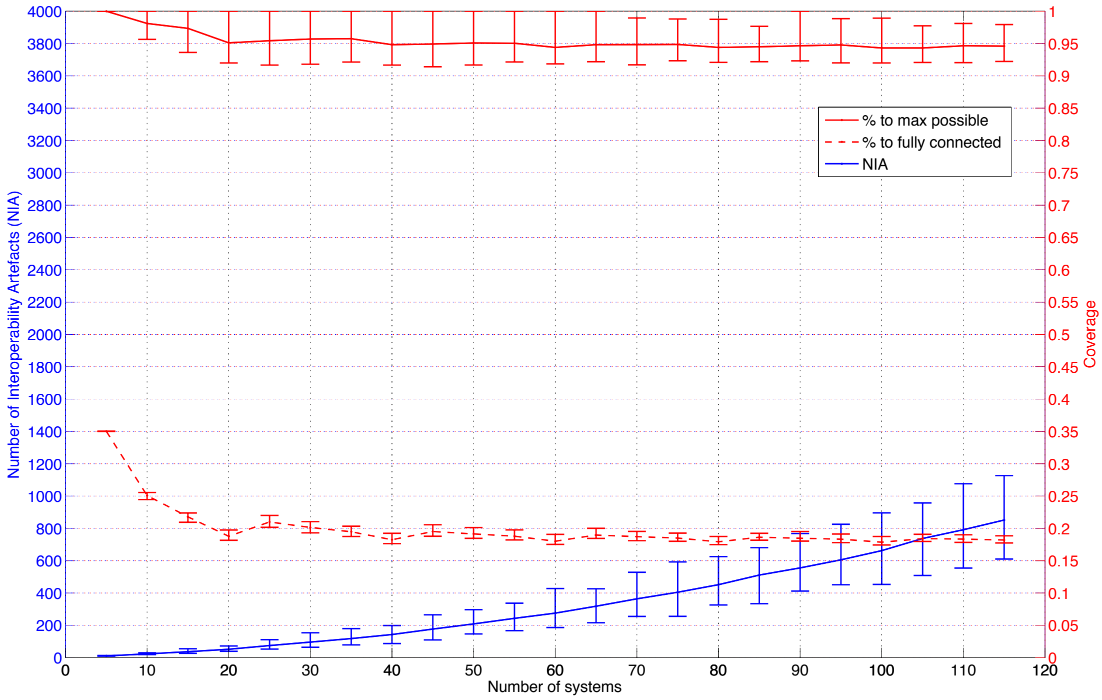


Image 40: Simulation result for 75% of modifiable systems, 25% of same formats, 25% of exporters and 75% of importers

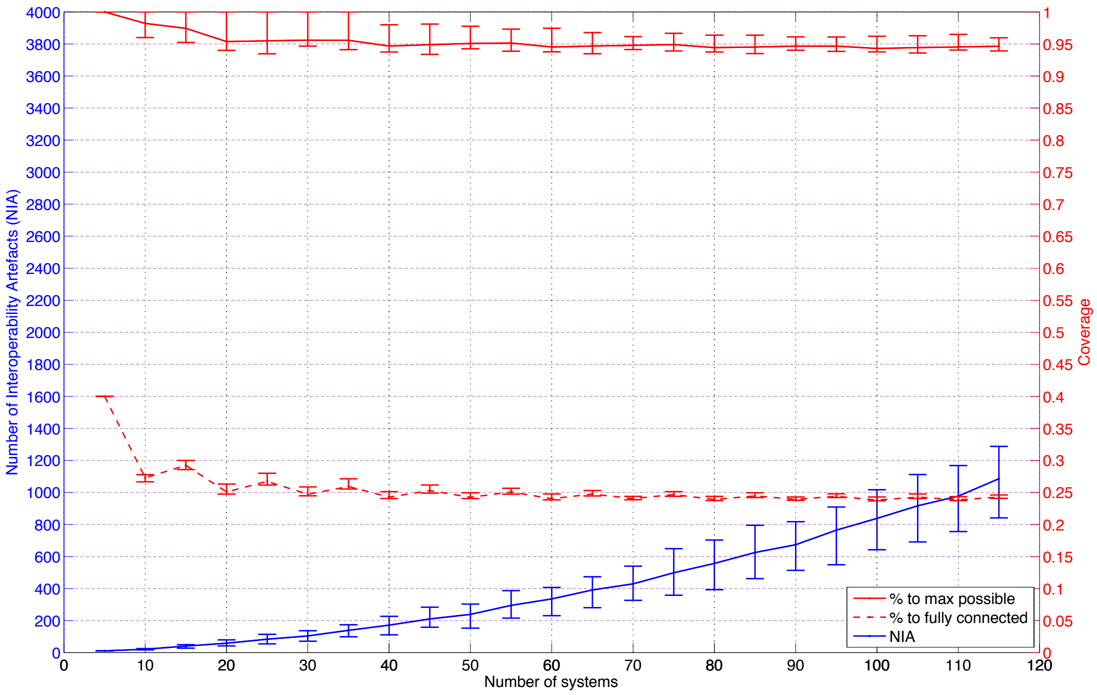


Image 41: Simulation result for 75% of modifiable systems, 25% of same formats, 50% of exporters and 50% of importers

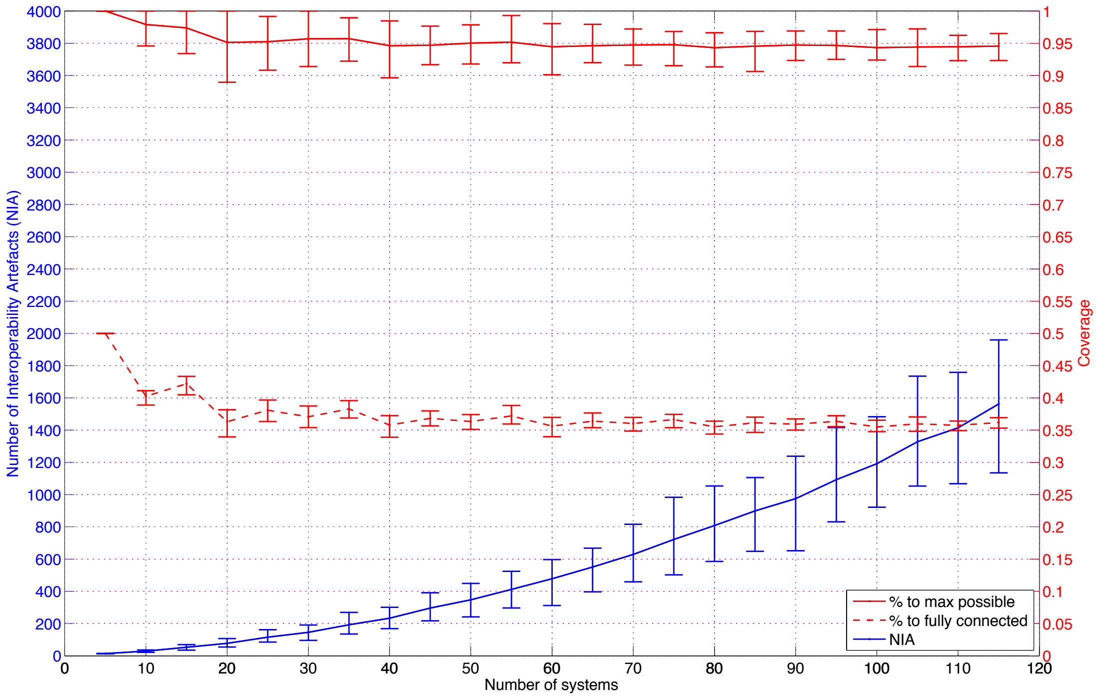


Image 42: Simulation result for 75% of modifiable systems, 25% of same formats, 50% of exporters and 75% of importers

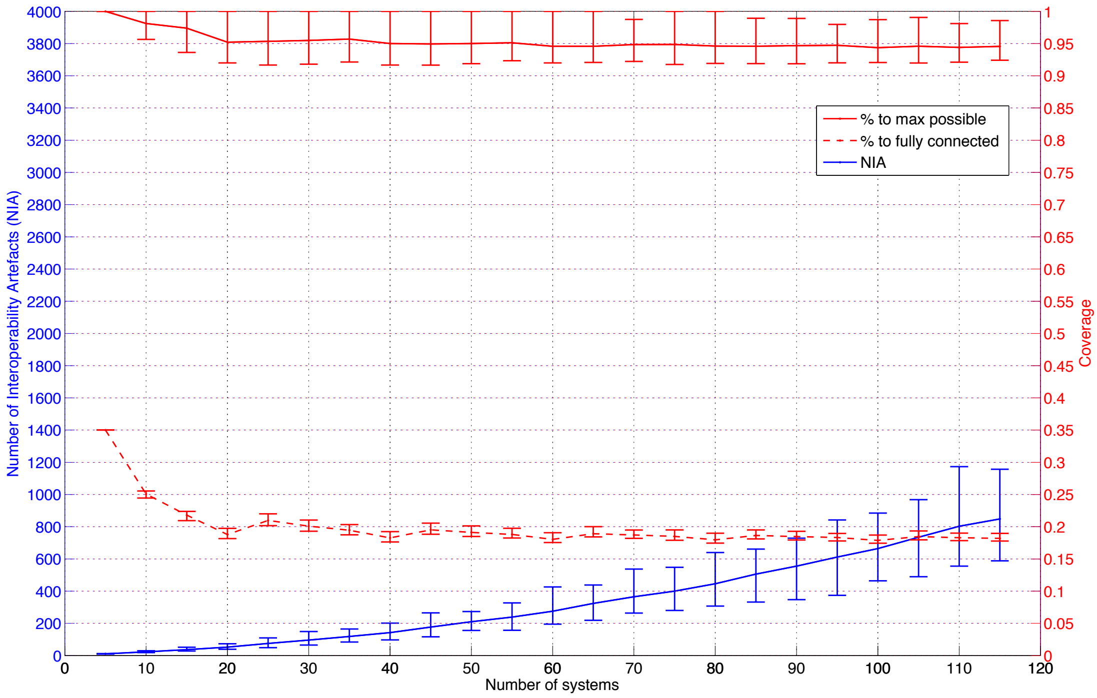


Image 43: Simulation result for 75% of modifiable systems, 25% of same formats, 75% of exporters and 25% of importers

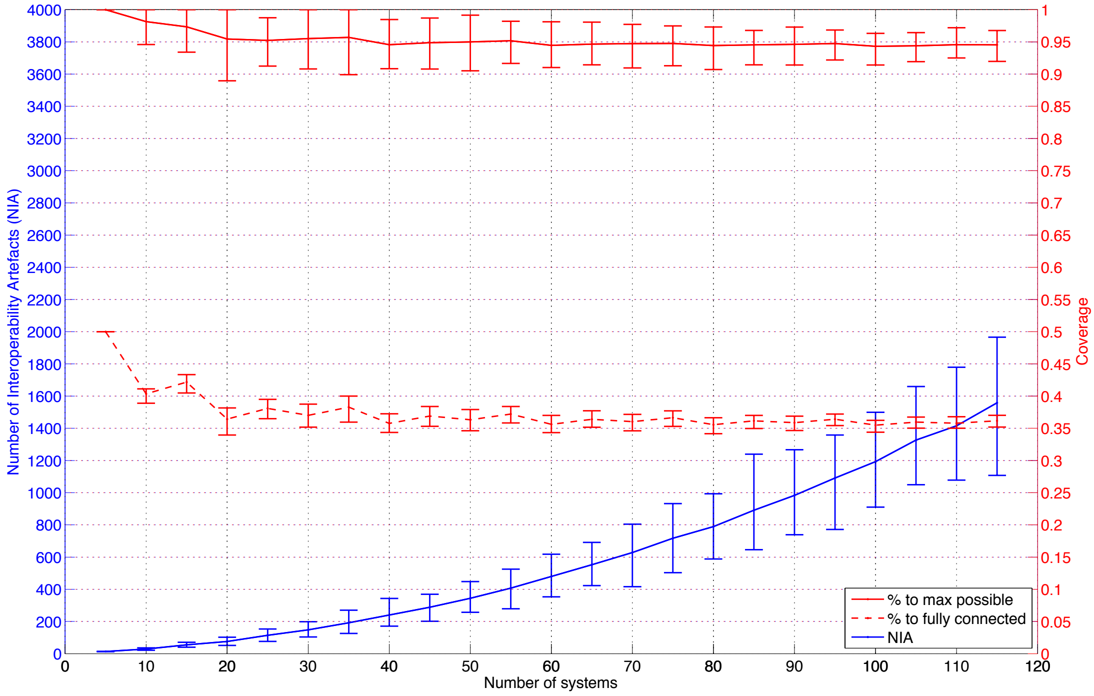


Image 44: Simulation result for 75% of modifiable systems, 25% of same formats, 75% of exporters and 50% of importers

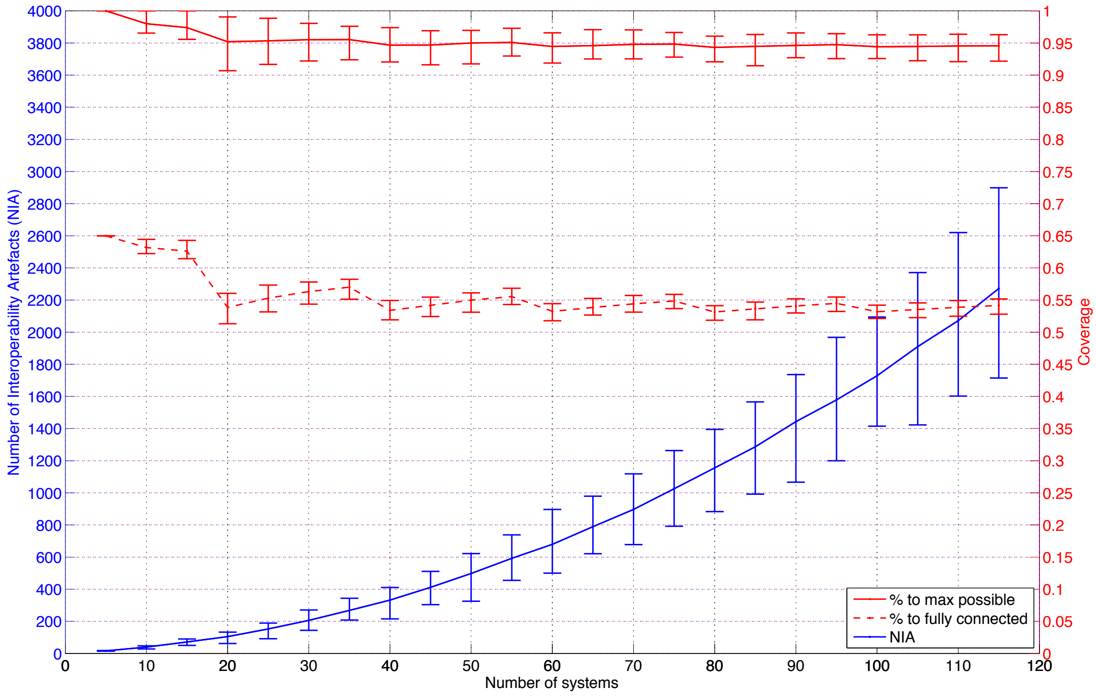


Image 45: Simulation result for 75% of modifiable systems, 25% of same formats, 75% of exporters and 75% of importers

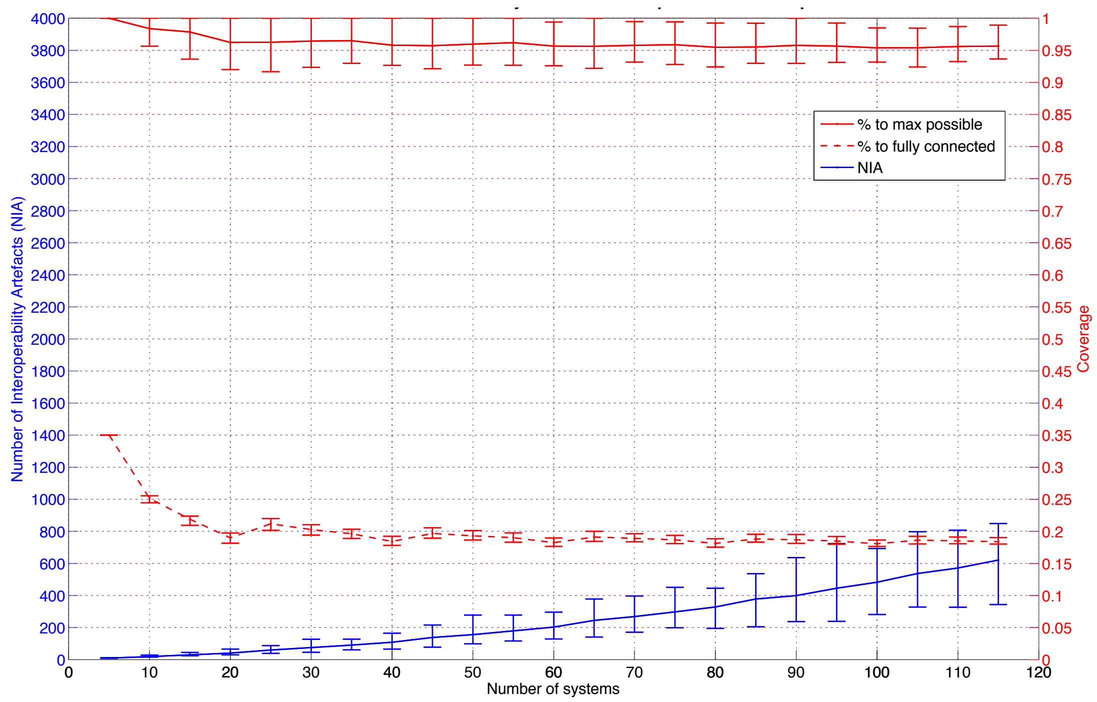


Image 46: Simulation result for 75% of modifiable systems, 50% of same formats, 25% of exporters and 75% of importers

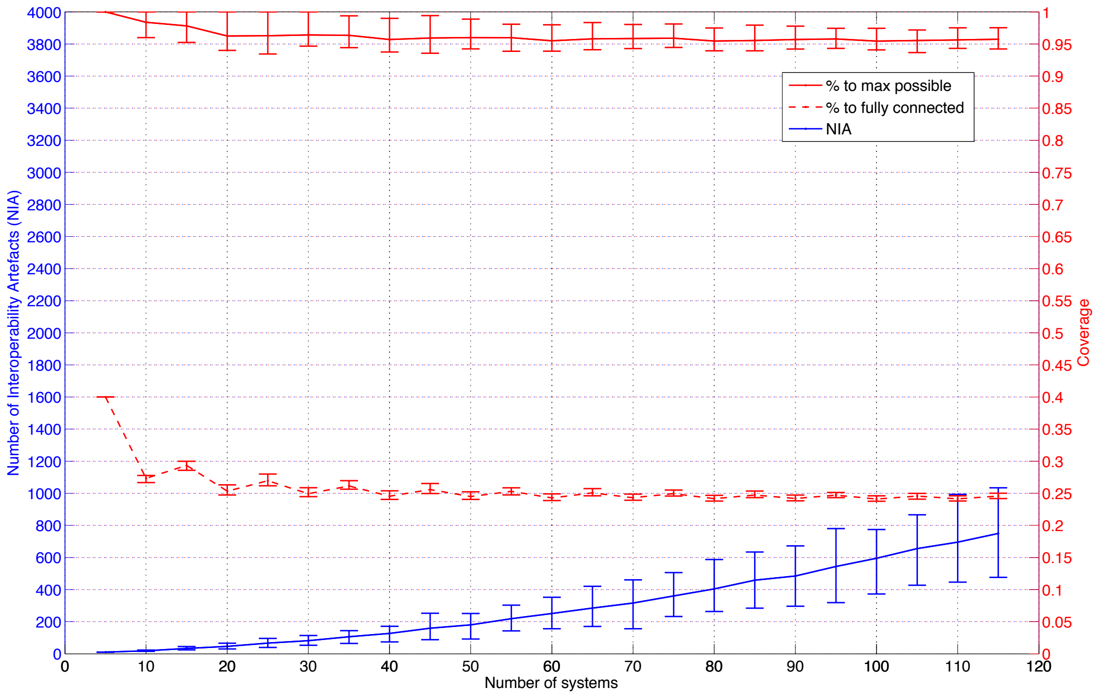


Image 47: Simulation result for 75% of modifiable systems, 50% of same formats, 50% of exporters and 50% of importers

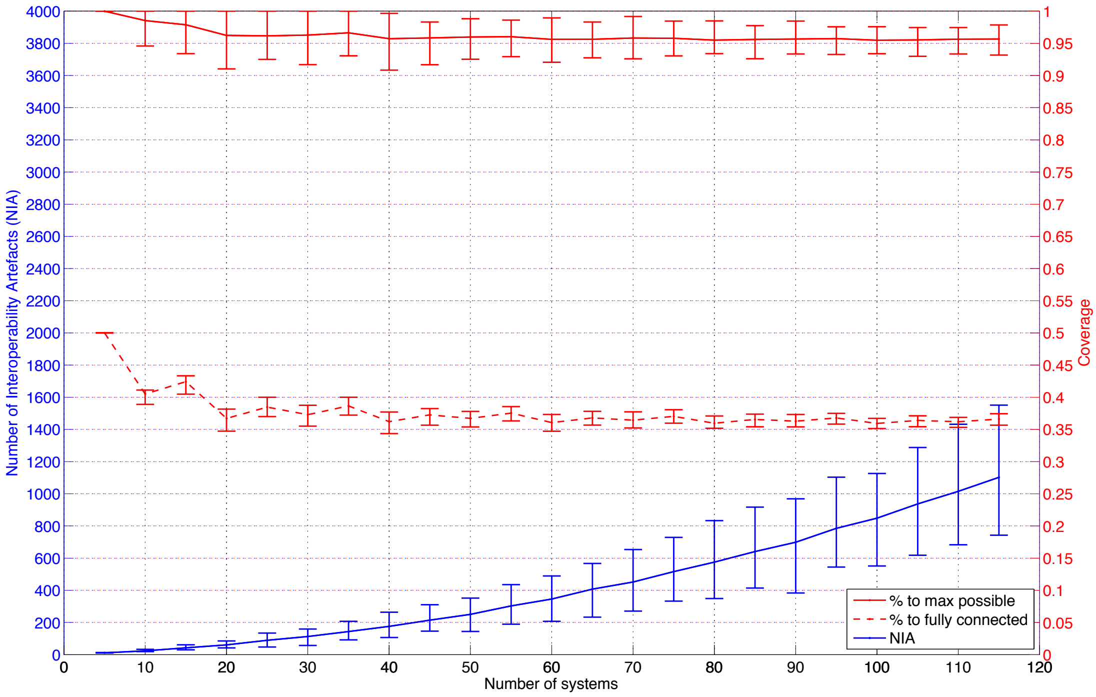


Image 48: Simulation result for 75% of modifiable systems, 50% of same formats, 50% of exporters and 75% of importers

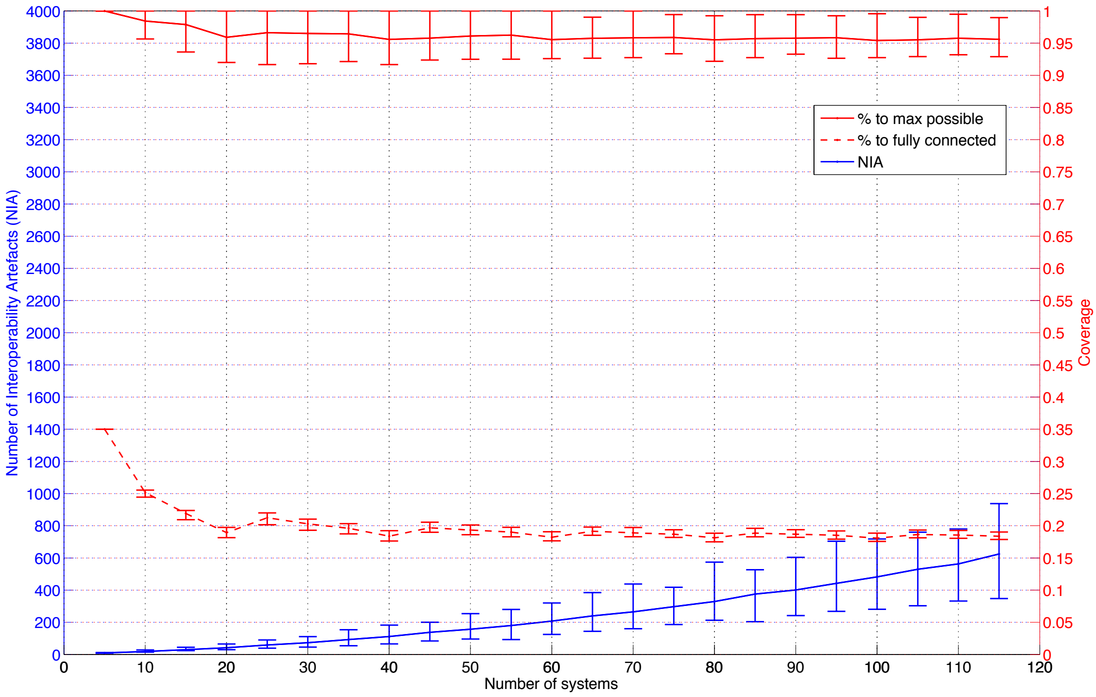


Image 49: Simulation result for 75% of modifiable systems, 50% of same formats, 75% of exporters and 25% of importers

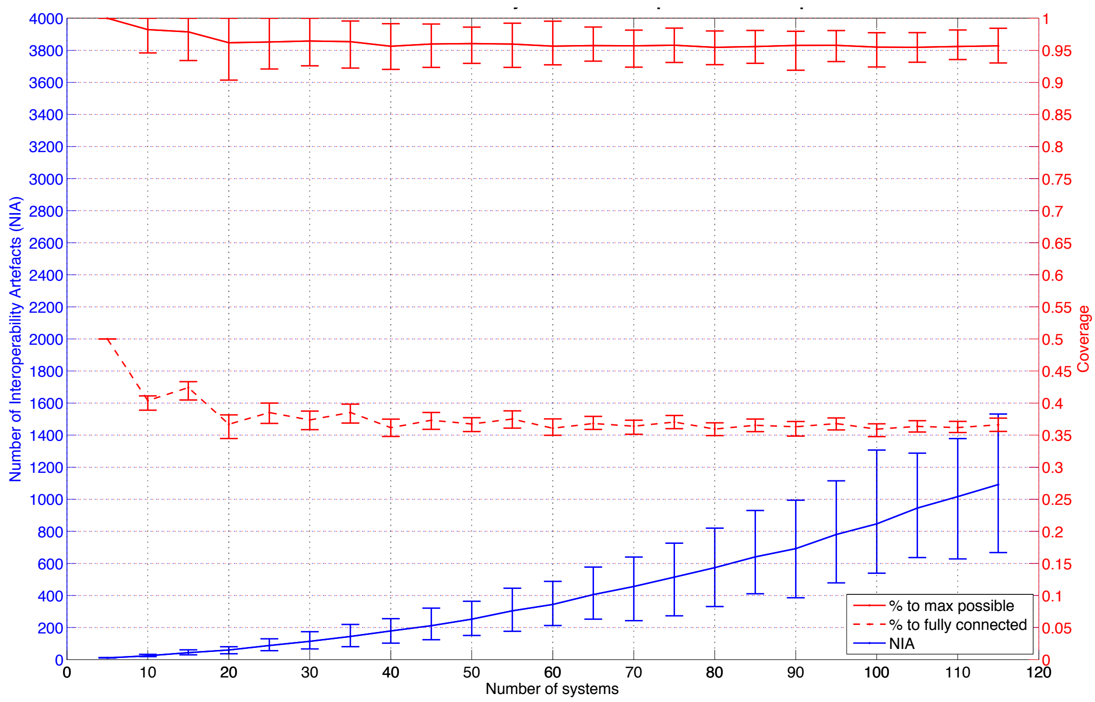


Image 50: Simulation result for 75% of modifiable systems, 50% of same formats, 75% of exporters and 50% of importers

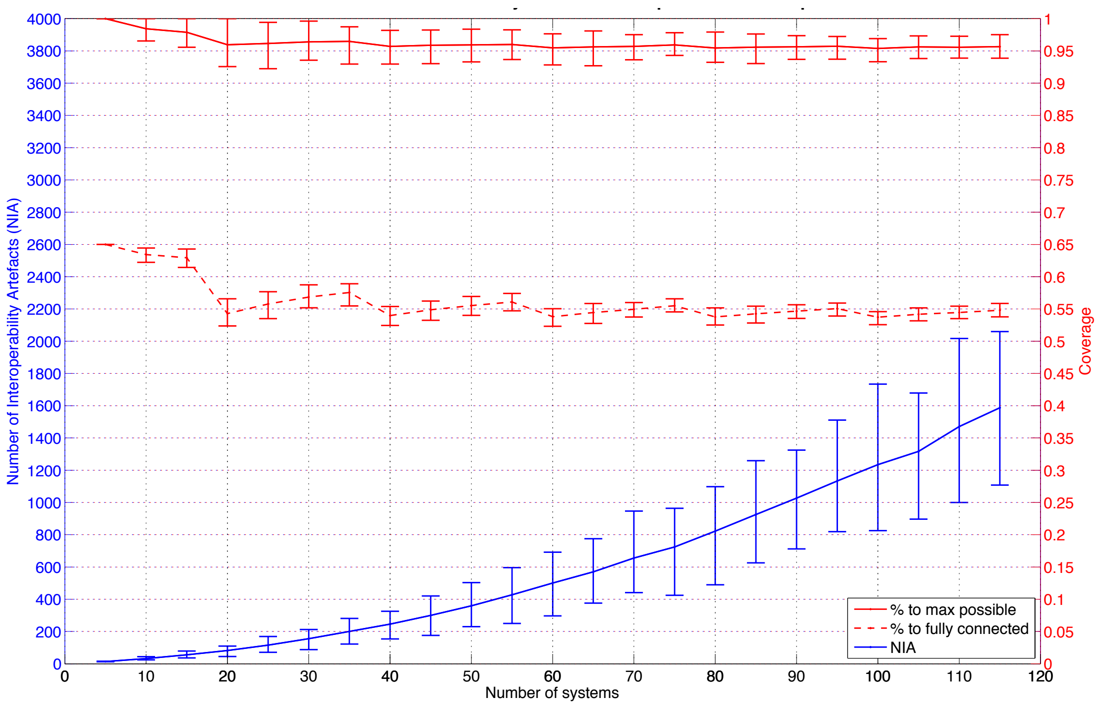


Image 51: Simulation result for 75% of modifiable systems, 50% of same formats, 75% of exporters and 75% of importers

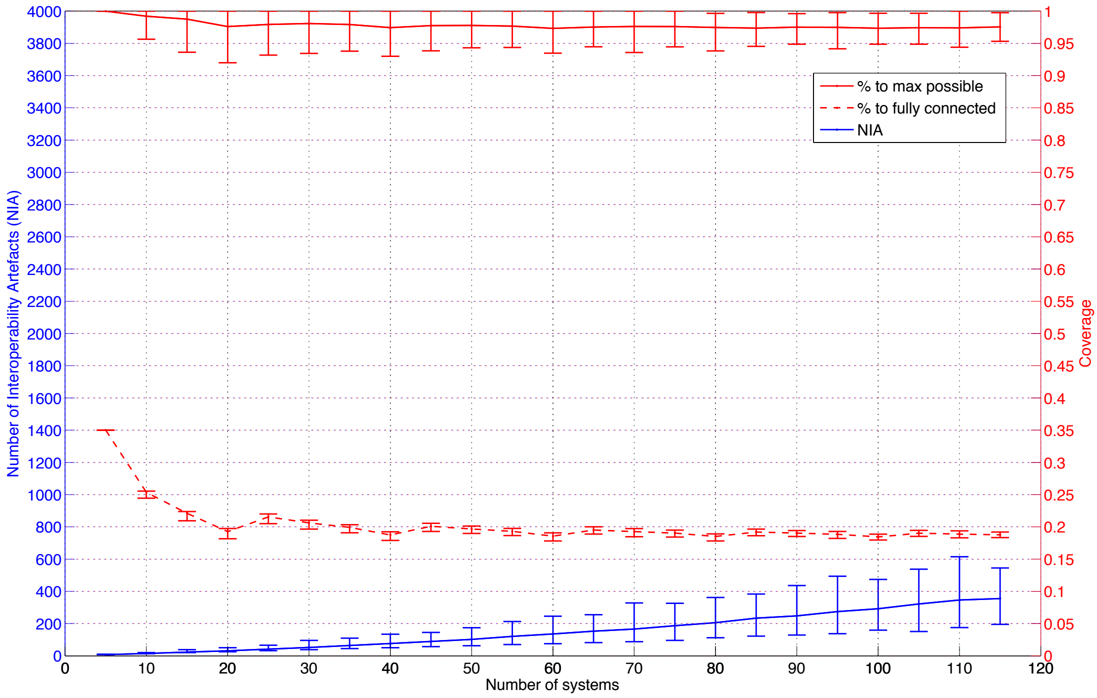


Image 52: Simulation result for 75% of modifiable systems, 75% of same formats, 25% of exporters and 75% of importers

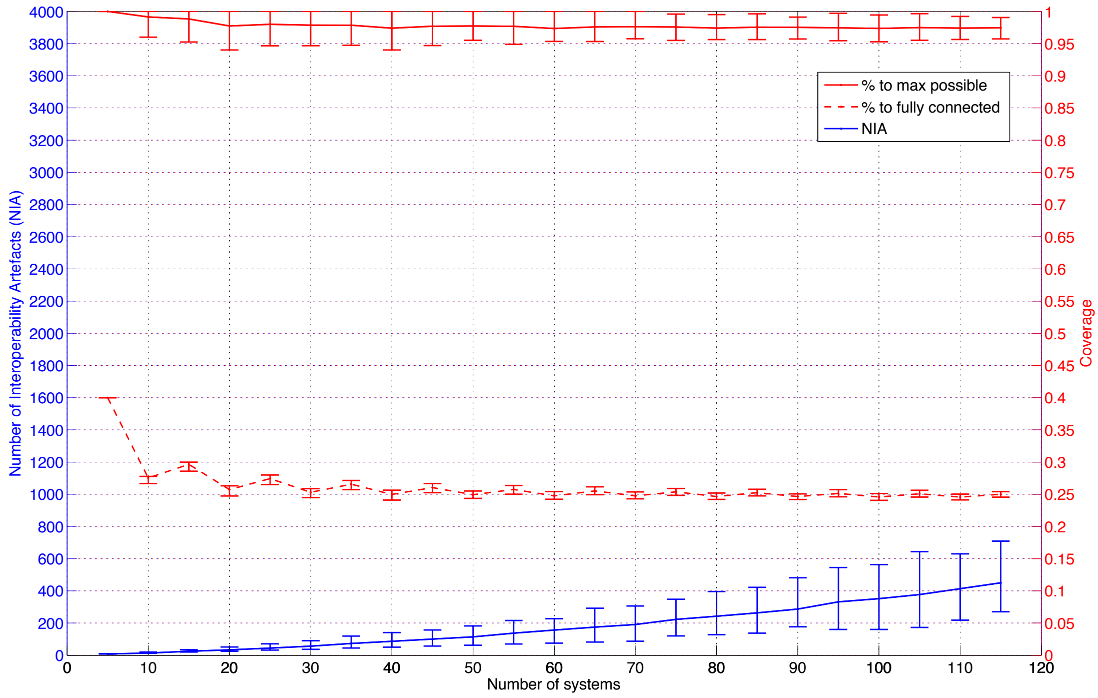


Image 53: Simulation result for 75% of modifiable systems, 75% of same formats, 50% of exporters and 50% of importers

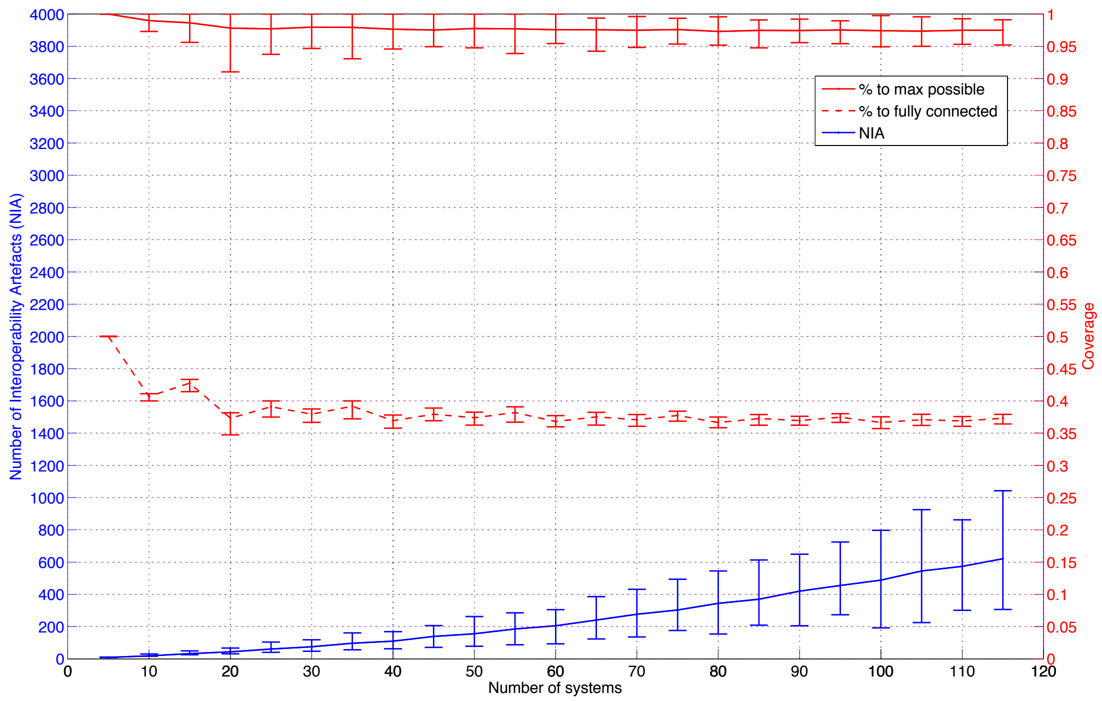


Image 54: Simulation result for 75% of modifiable systems, 75% of same formats, 50% of exporters and 75% of importers

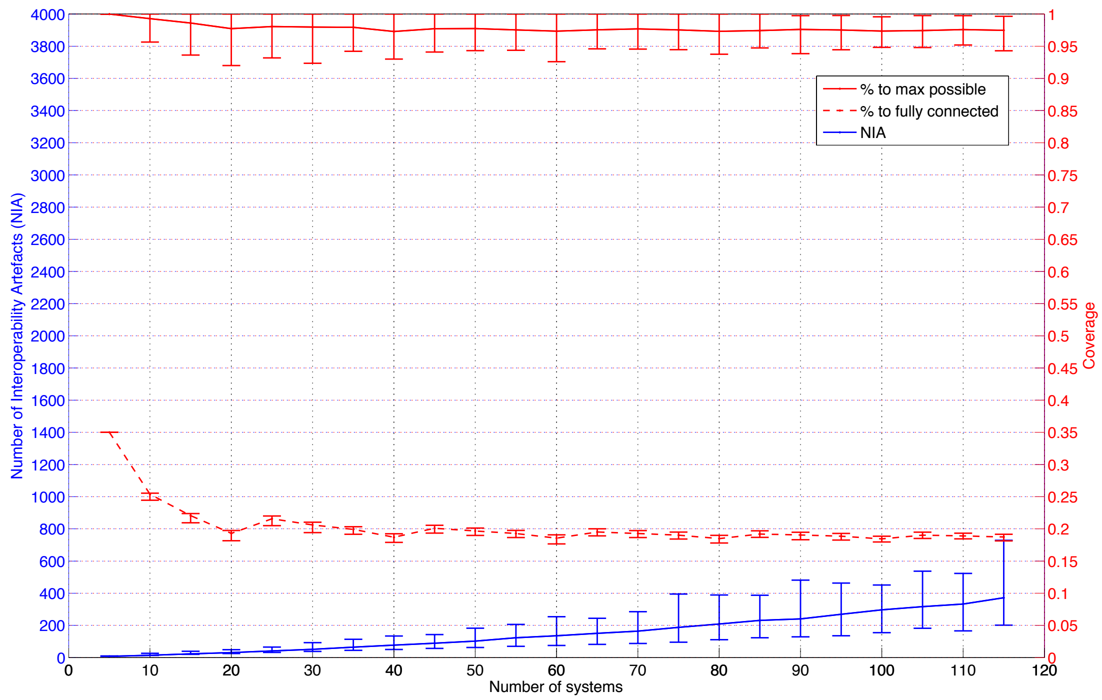


Image 55: Simulation result for 75% of modifiable systems, 75% of same formats, 75% of exporters and 25% of importers

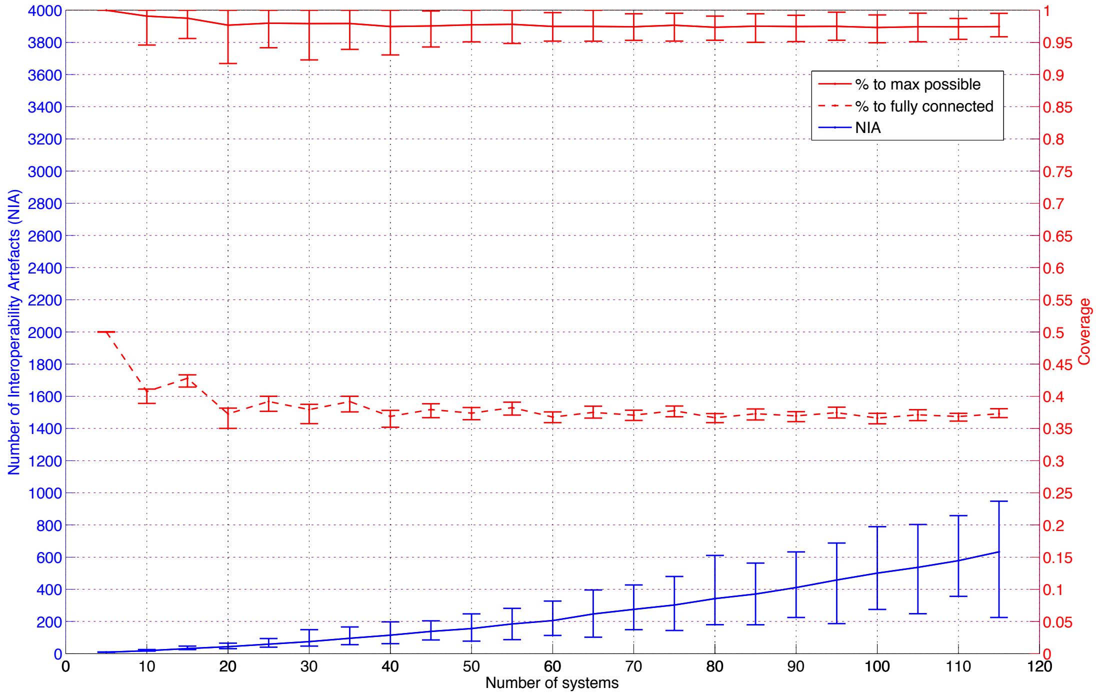


Image 56: Simulation result for 75% of modifiable systems, 75% of same formats, 75% of exporters and 50% of importers

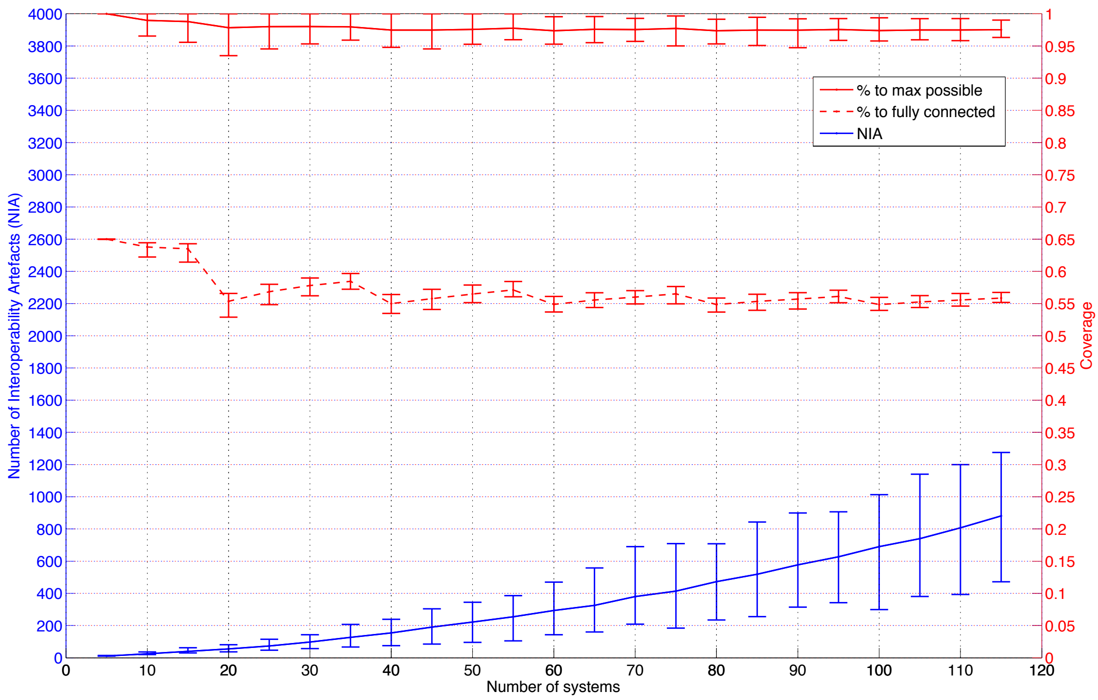


Image 57: Simulation result for 75% of modifiable systems, 75% of same formats, 75% of exporters and 75% of importers

Appendix C: Hub&Spoke Interoperability Simulation Graphics

In this appendix are presented the 18 graphics obtained as result from the simulations of the Hub-and-Spoke Interoperability approach. These graphics result from the combination of four parameters: number of systems, percentage of systems with same formats, percentage of systems with export capabilities and percentage of systems with import capabilities. The number of systems varies from 5 to 115 with a step of five. All the other four parameters assume the values 25%, 50% and 75%, being performed all the possible combinations, assuring that the sum of the percentage of Exporters and Importers is always same or greater than 100%. The variation of the parameters and the corresponding graphic are identified in the table below.

		Systems with Same Formats (%)		
		25	50	75
Exporters/Importers (%)	25/75	Image 58	Image 64	Image 70
	50/50	Image 59	Image 65	Image 71
	50/75	Image 60	Image 66	Image 72
	75/25	Image 61	Image 67	Image 73
	75/50	Image 62	Image 68	Image 74
	75/75	Image 63	Image 69	Image 75

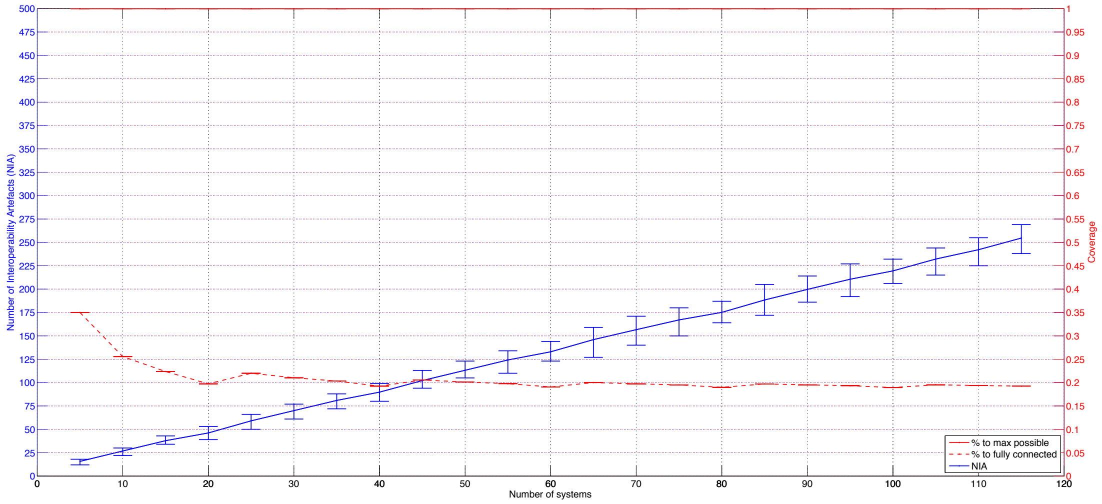


Image 58: Simulation result for 25% of same formats, 25% of exporters and 75% of importers

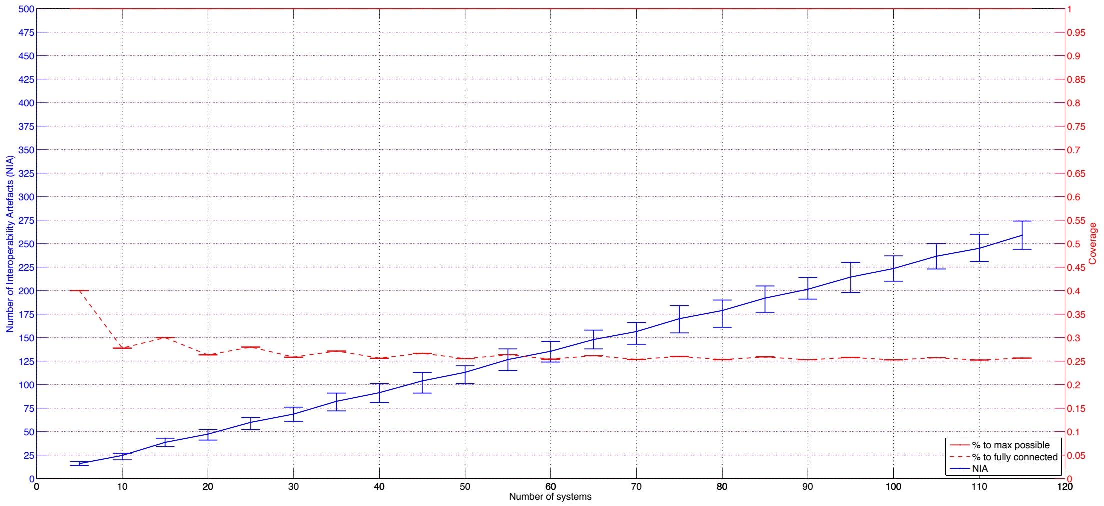


Image 59: Simulation result for 25% of same formats, 50% of exporters and 50% of importers

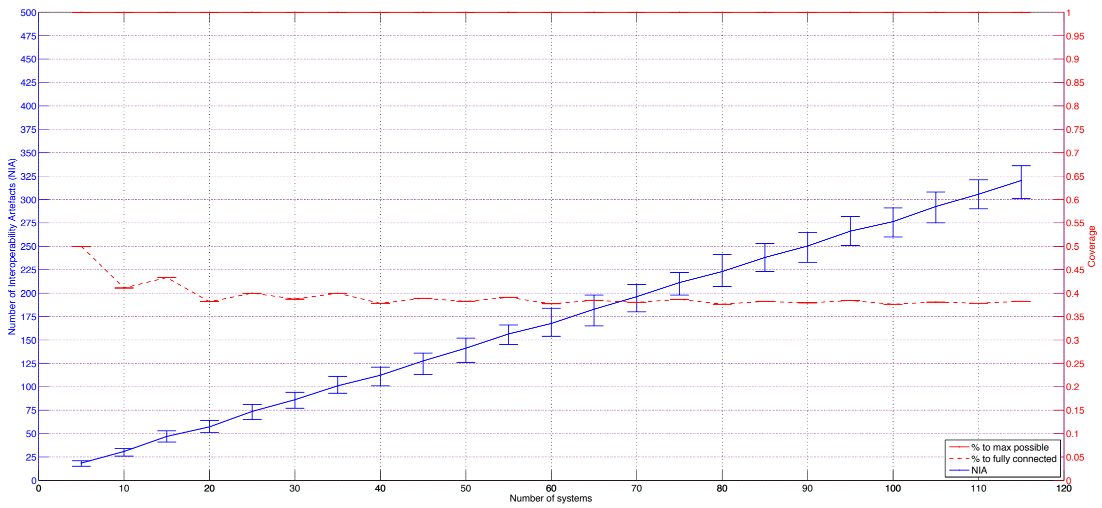


Image 60: Simulation result for 25% of same formats, 50% of exporters and 75% of importers

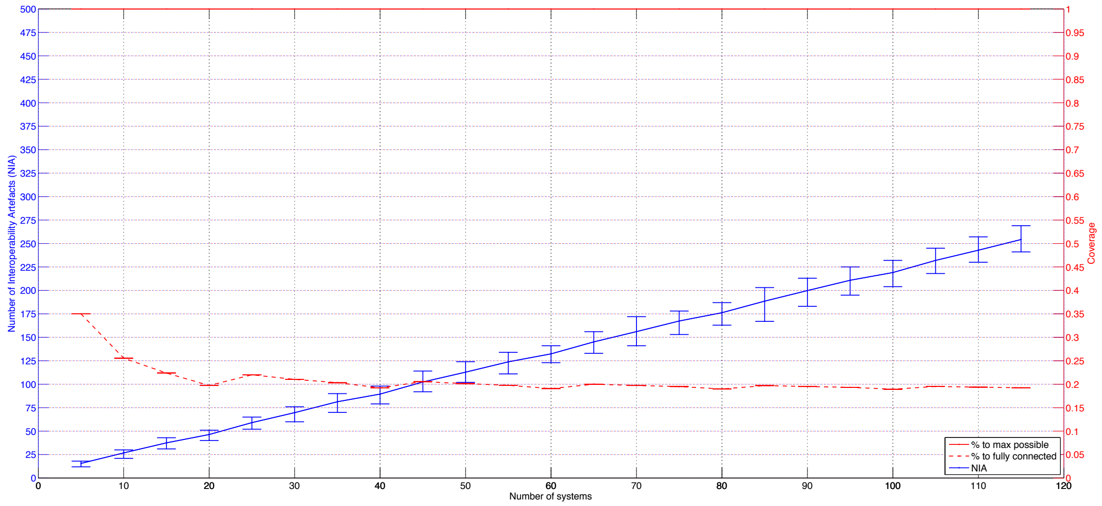


Image 61: Simulation result for 25% of same formats, 75% of exporters and 25% of importers

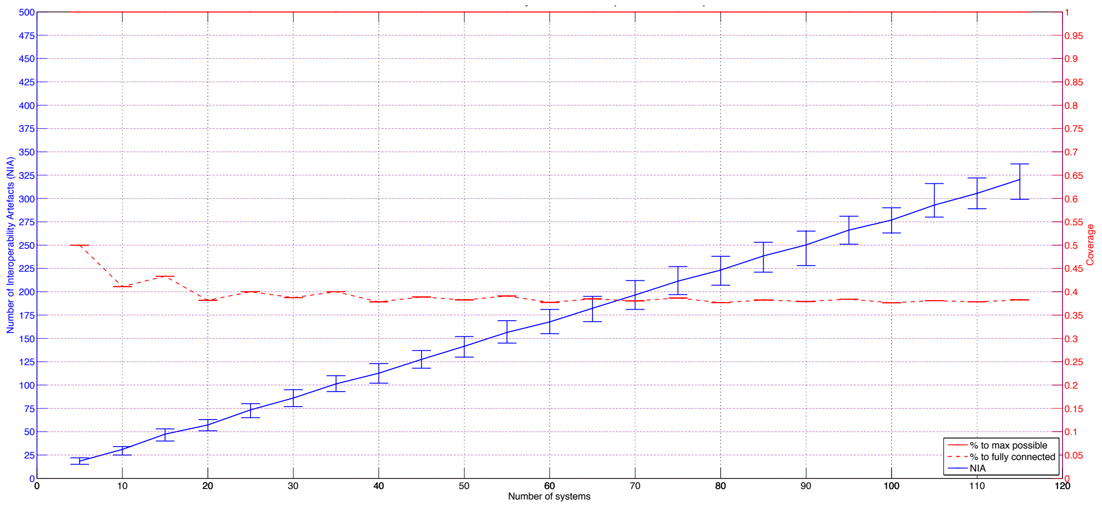


Image 62: Simulation result for 25% of same formats, 75% of exporters and 50% of importers

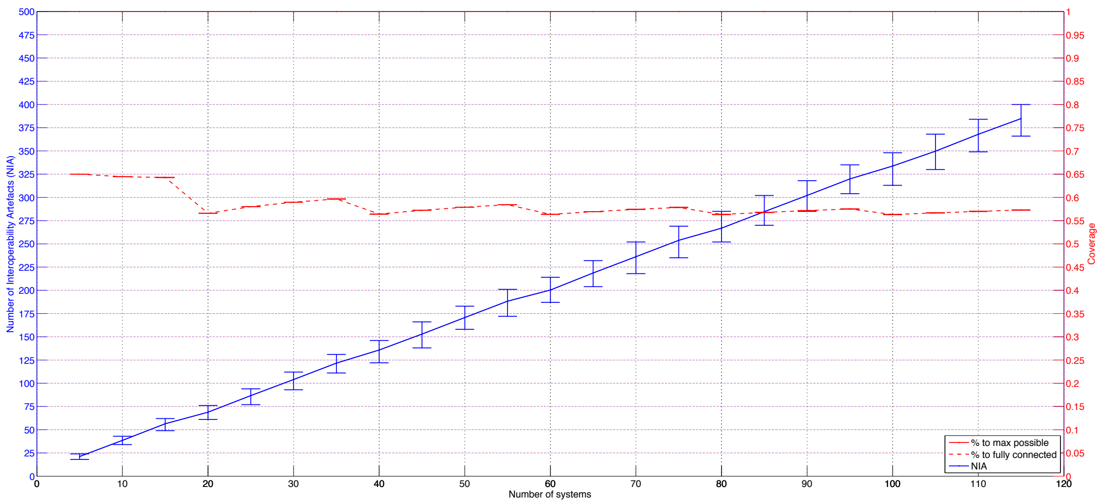


Image 63: Simulation result for 25% of same formats, 75% of exporters and 75% of importers

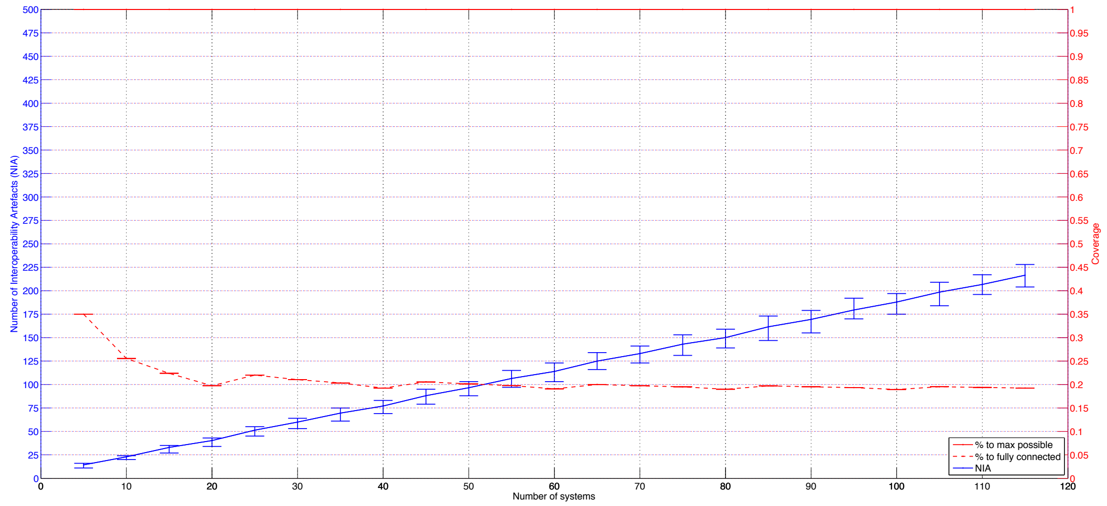


Image 64: Simulation result for 50% of same formats, 25% of exporters and 75% of importers

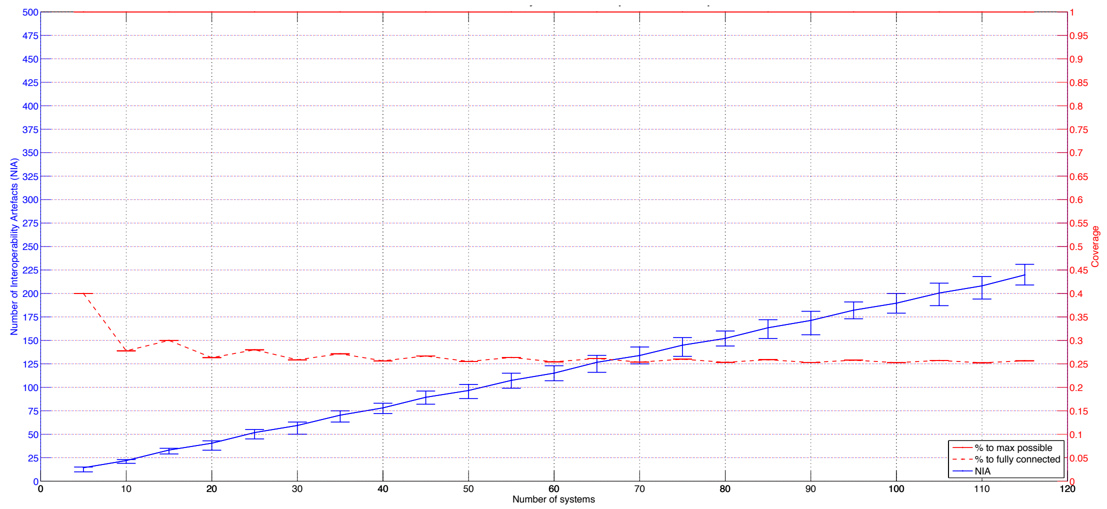


Image 65: Simulation result for 50% of same formats, 50% of exporters and 50% of importers

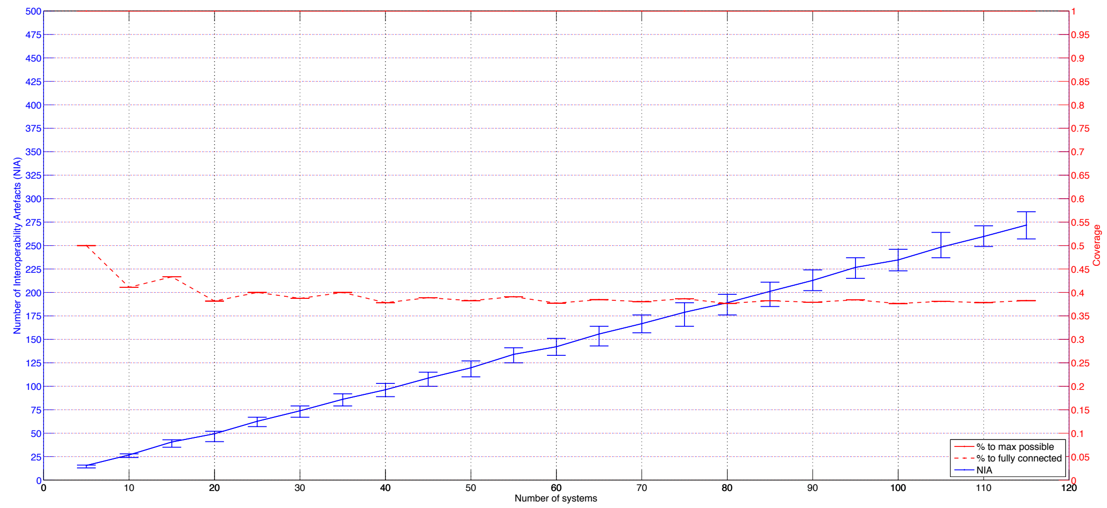


Image 66: Simulation result for 50% of same formats, 50% of exporters and 75% of importers

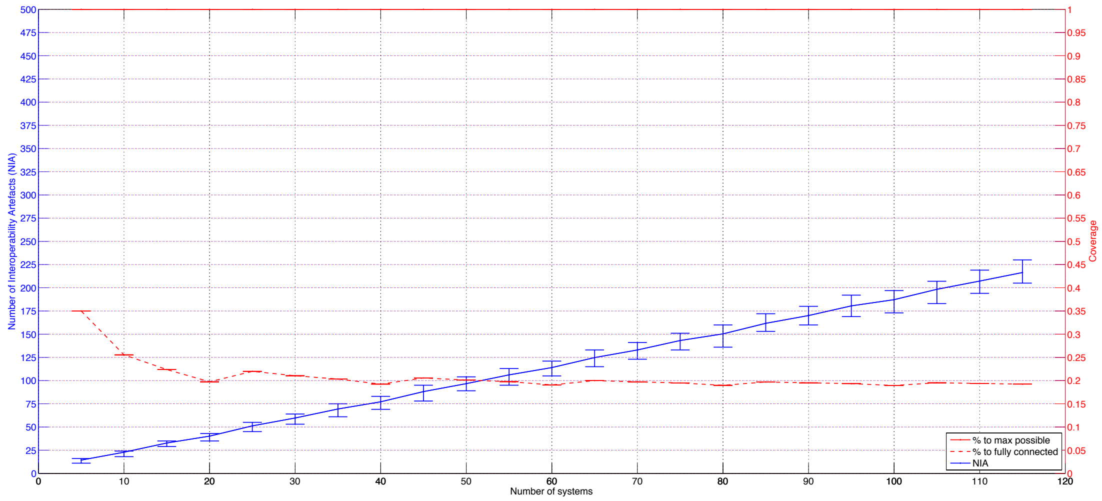


Image 67: Simulation result for 50% of same formats, 75% of exporters and 25% of importers

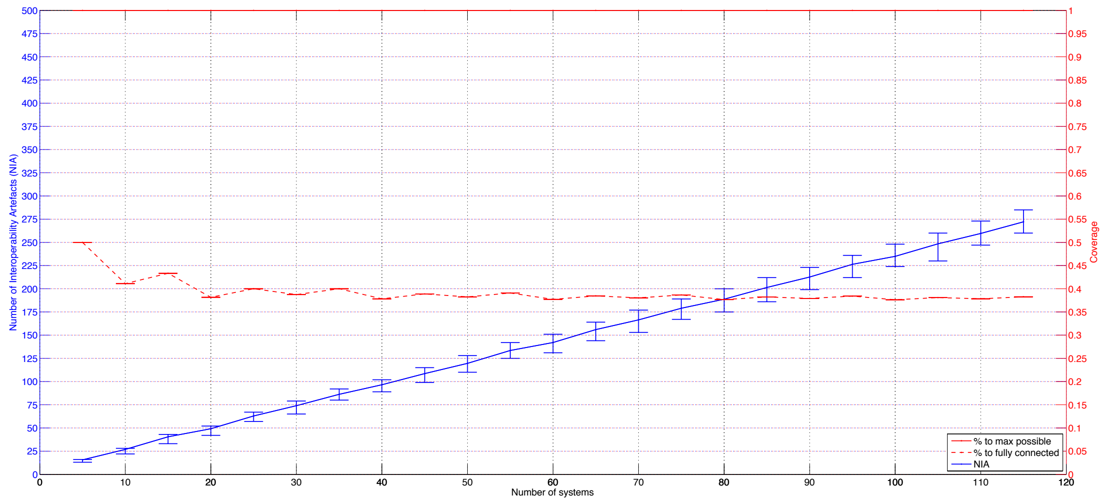


Image 68: Simulation result for 50% of same formats, 75% of exporters and 50% of importers

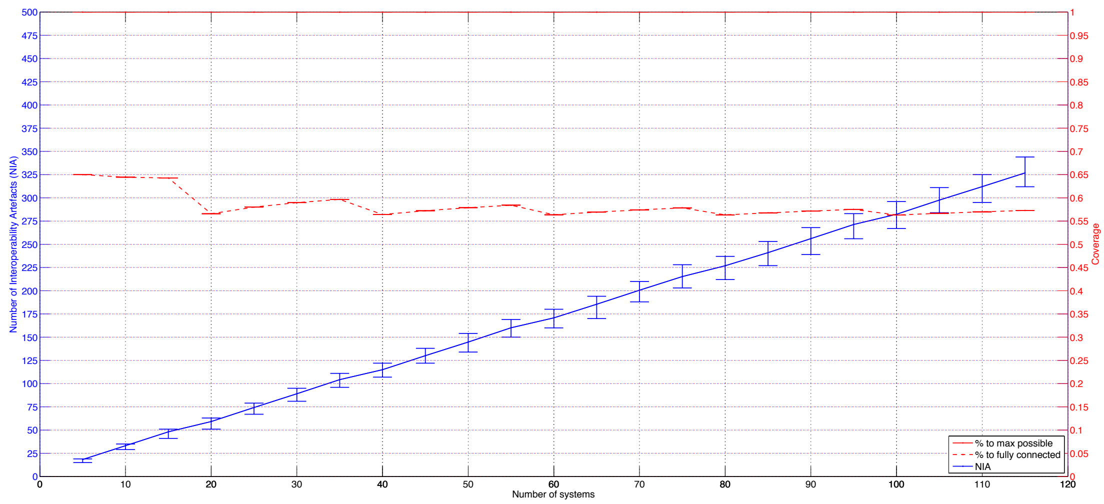


Image 69: Simulation result for 50% of same formats, 75% of exporters and 75% of importers

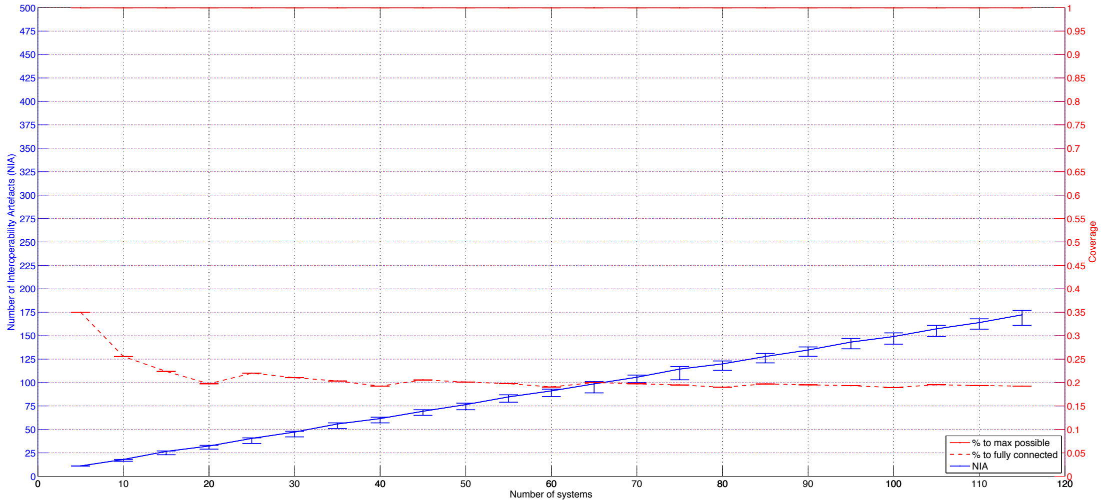


Image 70: Simulation result for 75% of same formats, 25% of exporters and 75% of importers

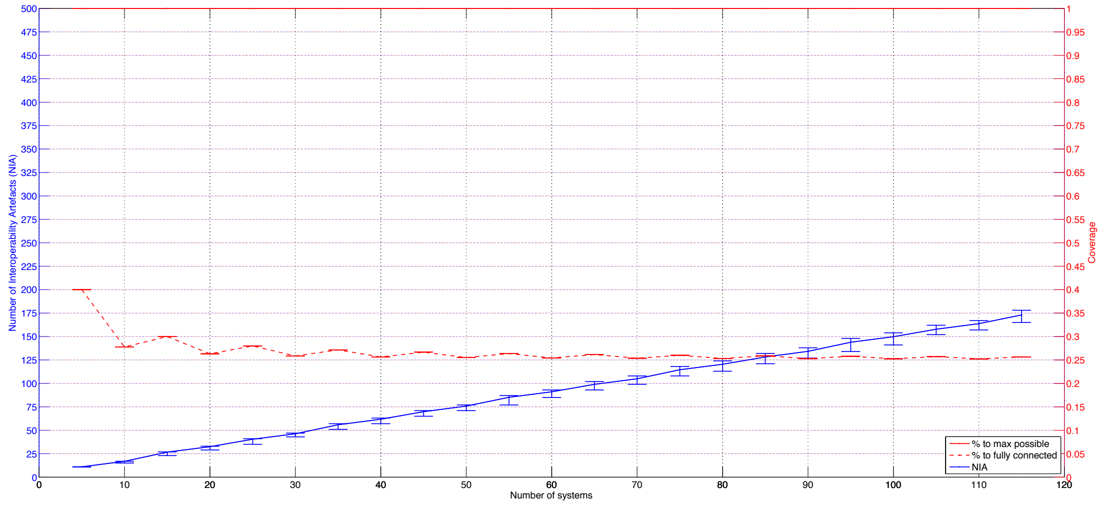


Image 71: Simulation result for 75% of same formats, 50% of exporters and 50% of importers

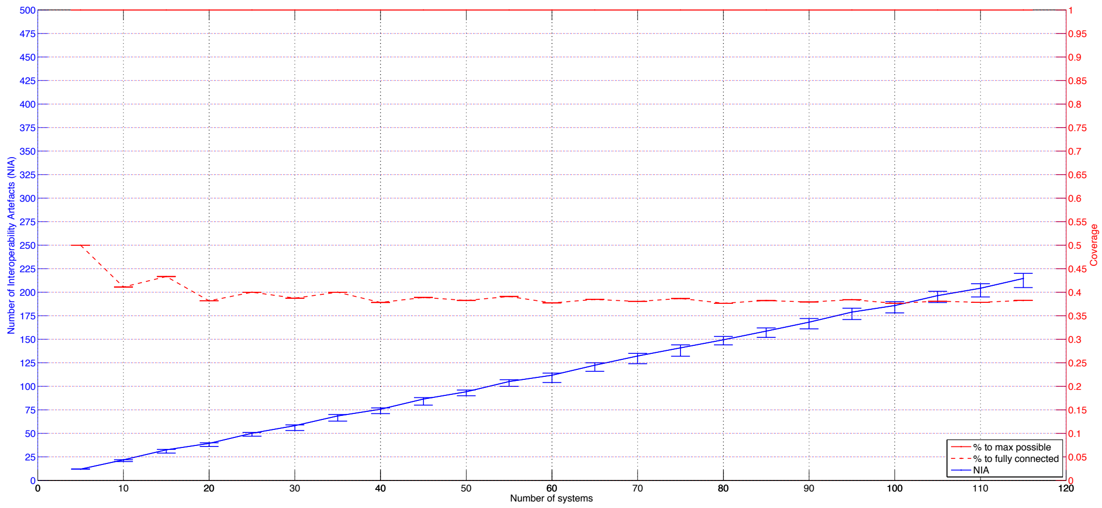


Image 72: Simulation result for 75% of same formats, 50% of exporters and 75% of importers

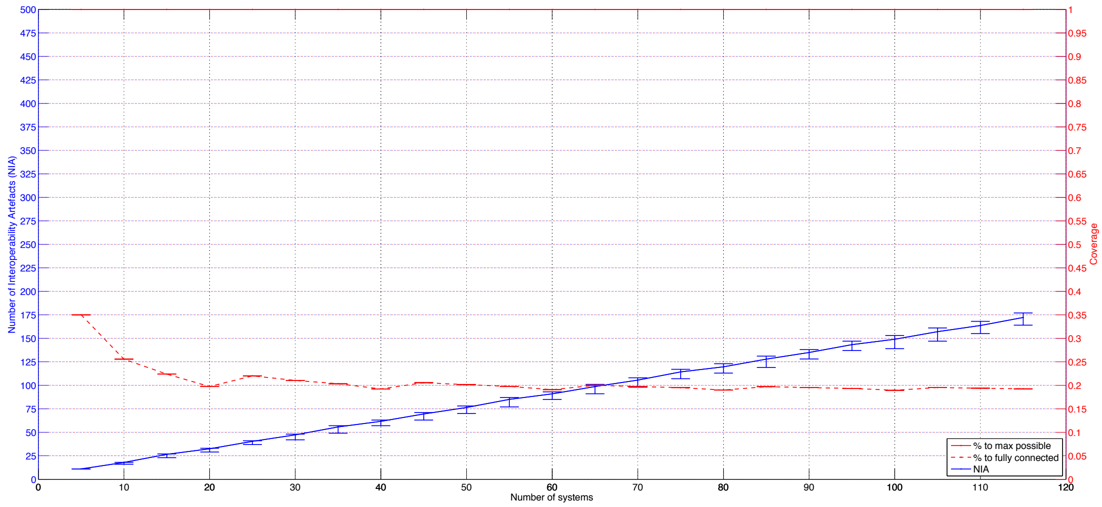


Image 73: Simulation result for 75% of same formats, 75% of exporters and 25% of importers

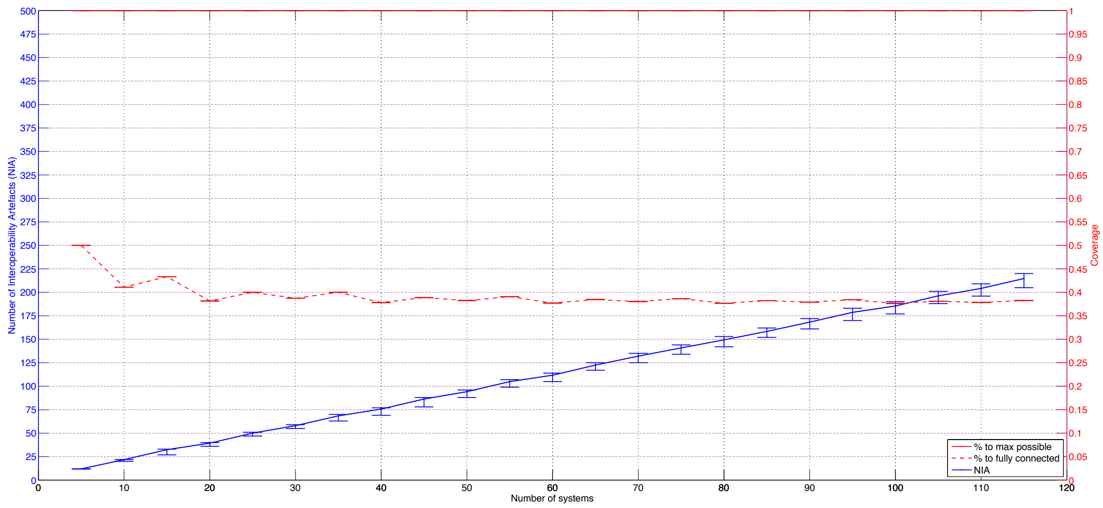


Image 74: Simulation result for 75% of same formats, 75% of exporters and 50% of importers

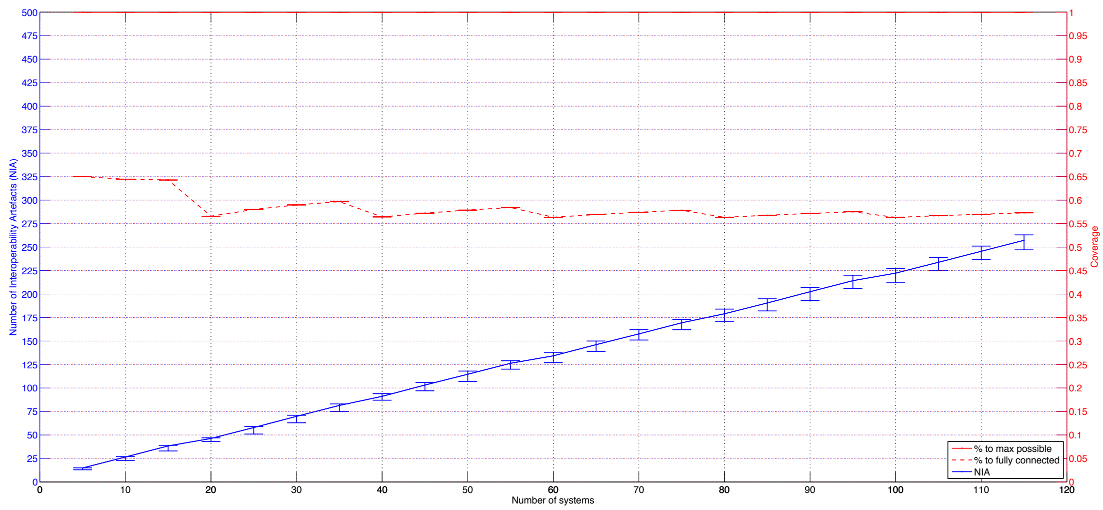


Image 75: Simulation result for 75% of same formats, 75% of exporters and 75% of importers

Appendix D: H&S vs. P2P Interoperability Small-Scale Graphs

In this appendix are presented the 12 graphics used to support the study of the small-scale performance of both Point-to-Point Interoperability and Hub-and-Spoke Interoperability approaches. These graphics result from the combination of five parameters: number of systems, percentage of modifiable systems, percentage of systems with same formats, percentage of systems with export capabilities and percentage of systems with import capabilities; being the results grouped by equal values of percentage of modifiable systems and by the pair of percentage of export and import capabilities. The number of systems in the scenarios simulates ranges from 2 to 30 with a step of 2 systems. All the other four parameters assume the values 25%, 50% and 75%, being performed all the possible combinations, assuring that the sum of the percentage of Exporters and Importers is always equal or greater than 100%. The variation of parameters and corresponding graphics are identified in the table below.

		Modifiable Systems (%)		
		25	50	75
Exporters/Importers (%)	25/75	Image 76	Image 80	Image 84
	50/50	Image 77	Image 81	Image 85
	50/75	Image 78	Image 82	Image 86
	75/75	Image 79	Image 83	Image 87

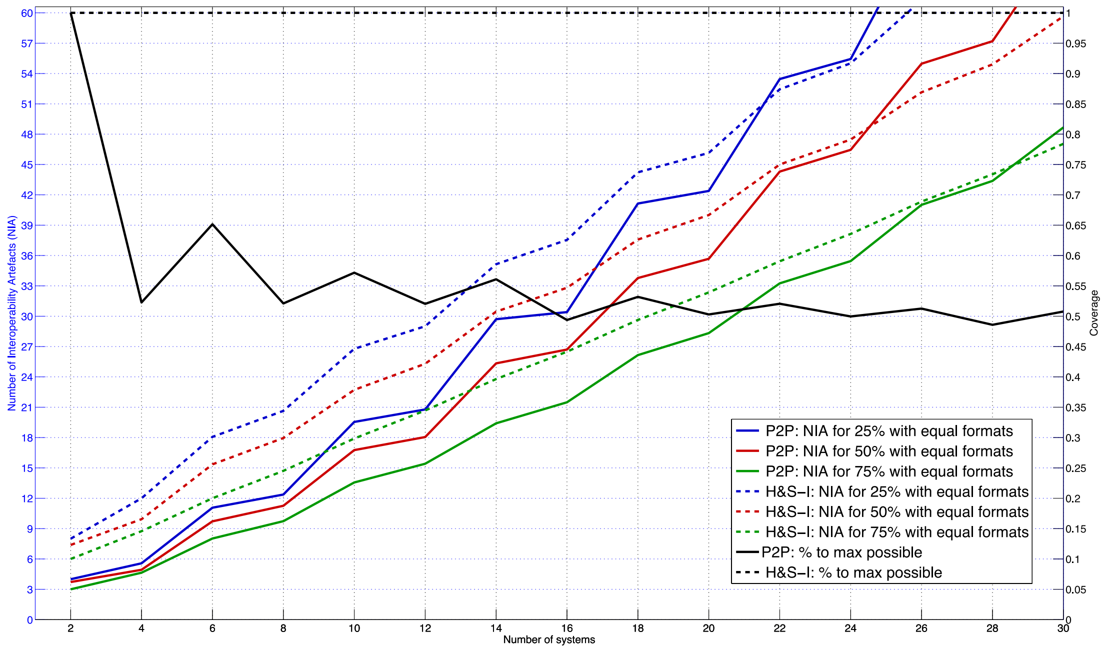


Image 76: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 25% of exporters and 75% of importers

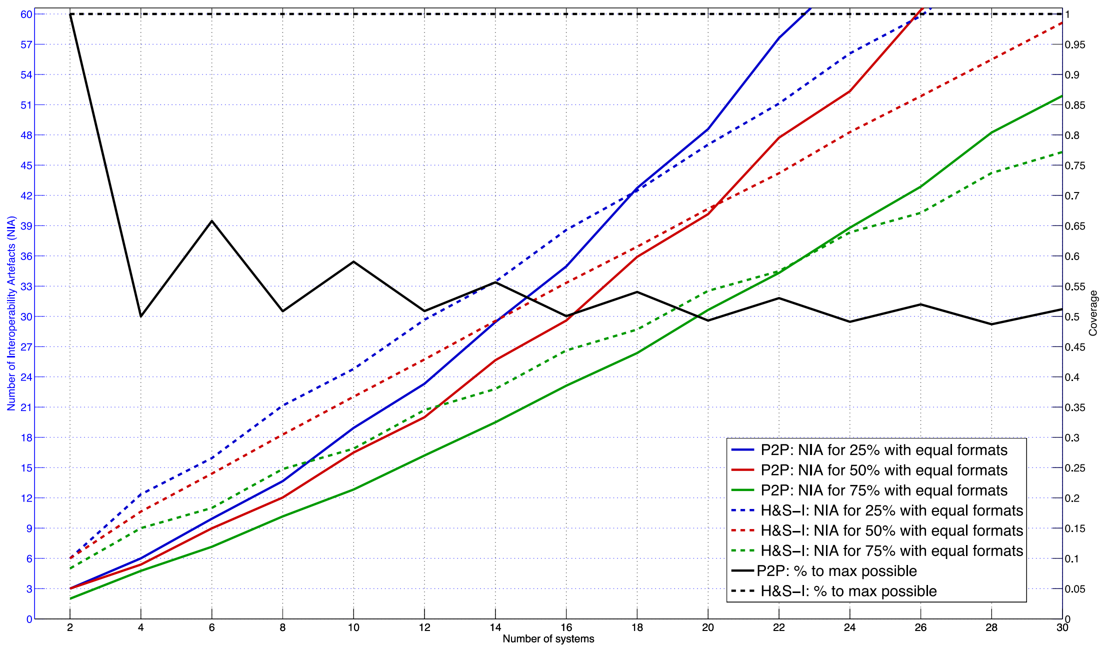


Image 77: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 50% of exporters and 50% of importers

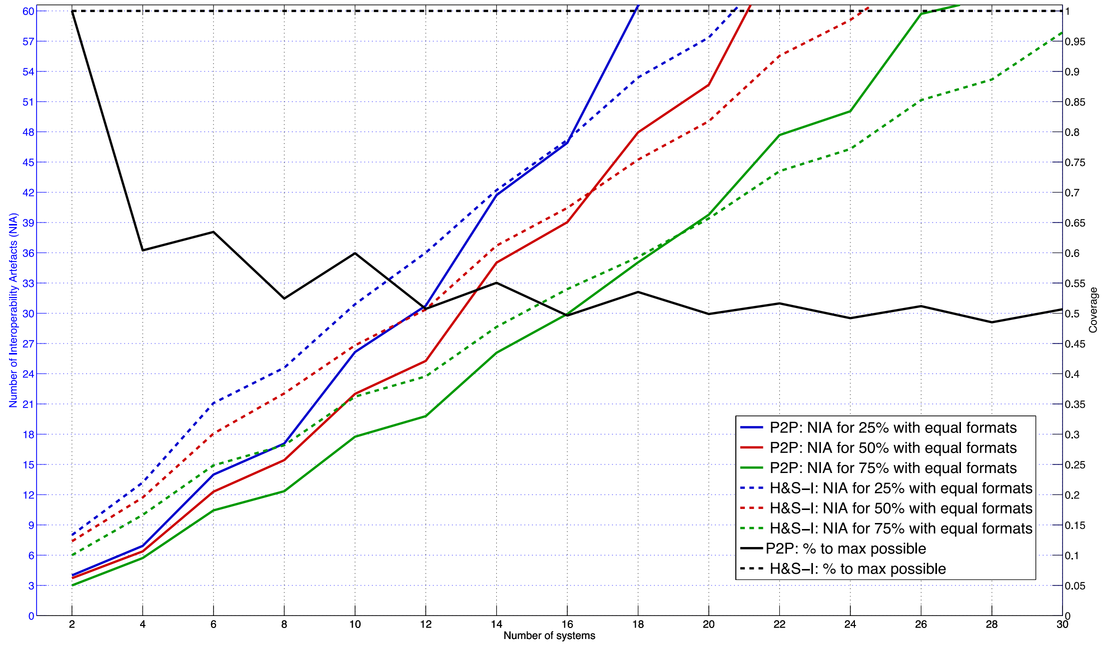


Image 78: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 50% of exporters and 75% of importers

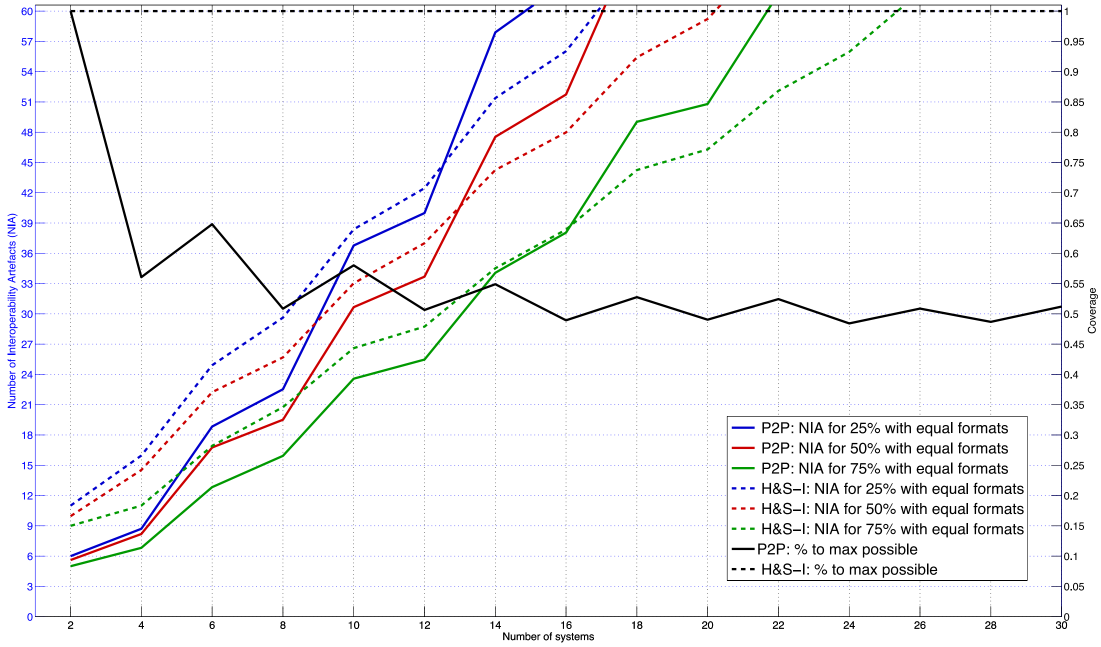


Image 79: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 75% of exporters and 75% of importers

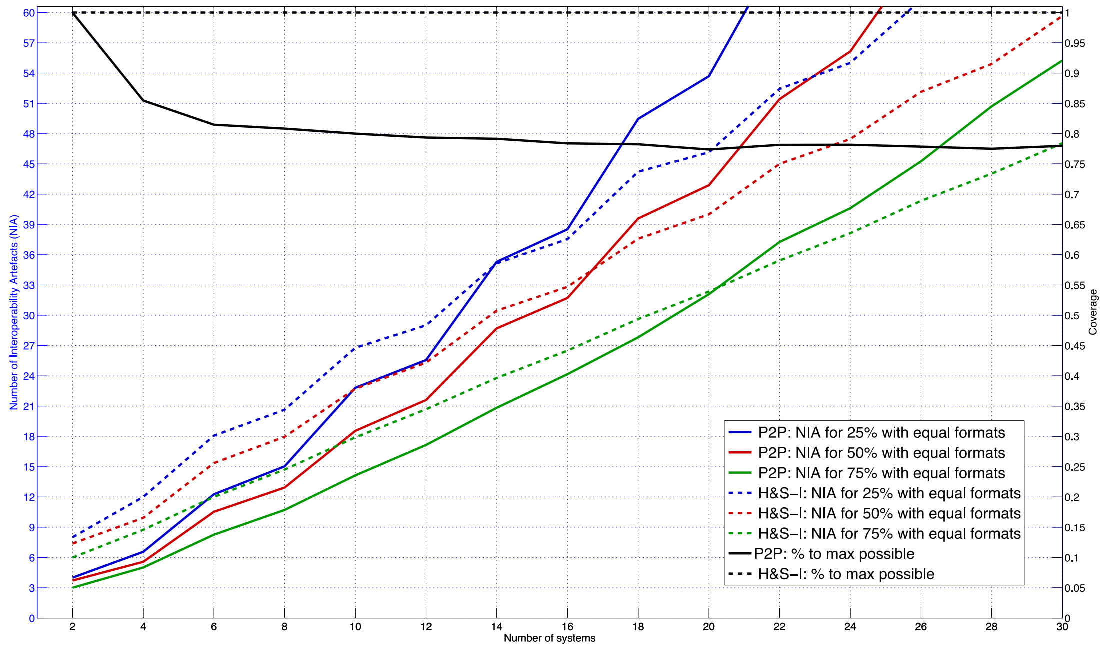


Image 80: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 25% of exporters and 75% of importers

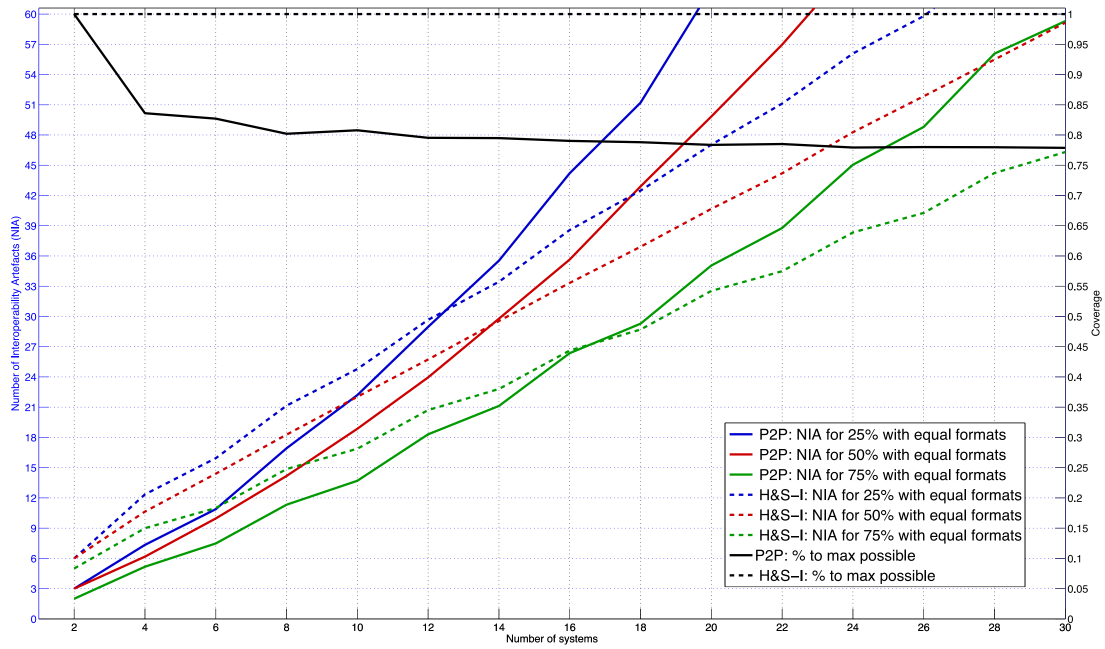


Image 81: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 50% of exporters and 50% of importers

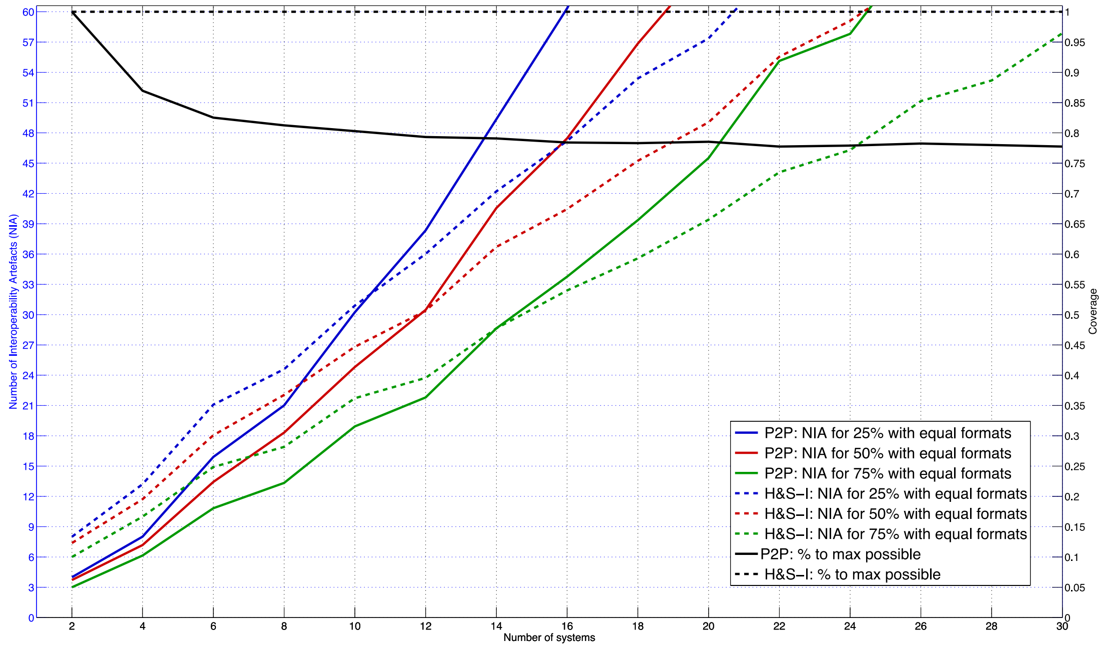


Image 82: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 50% of exporters and 75% of importers

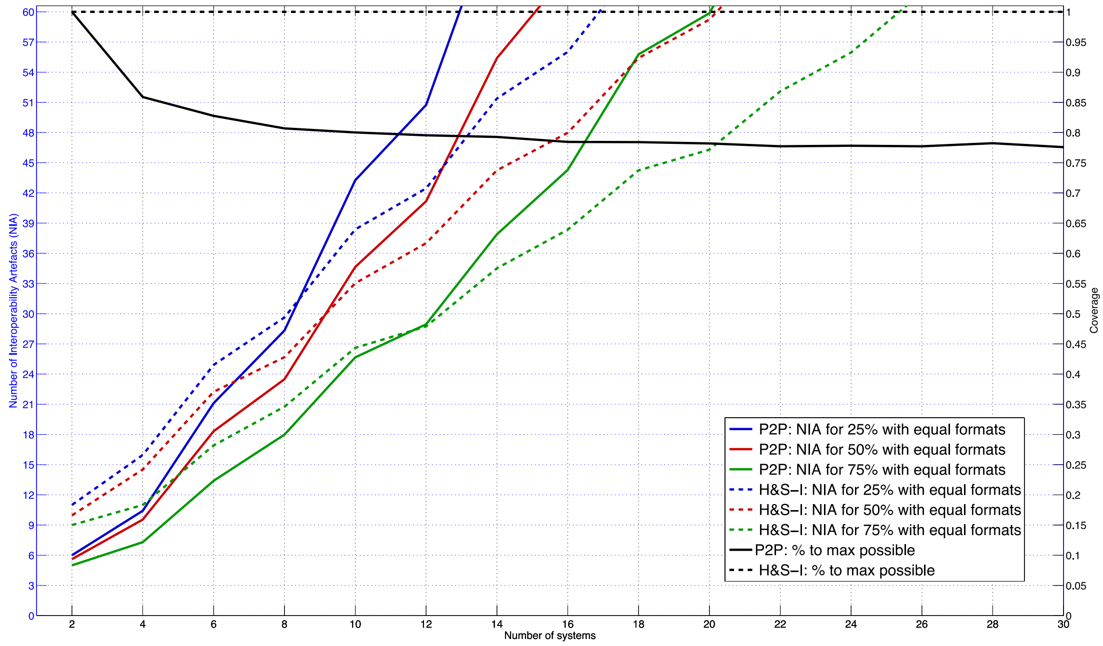


Image 83: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 75% of exporters and 75% of importers

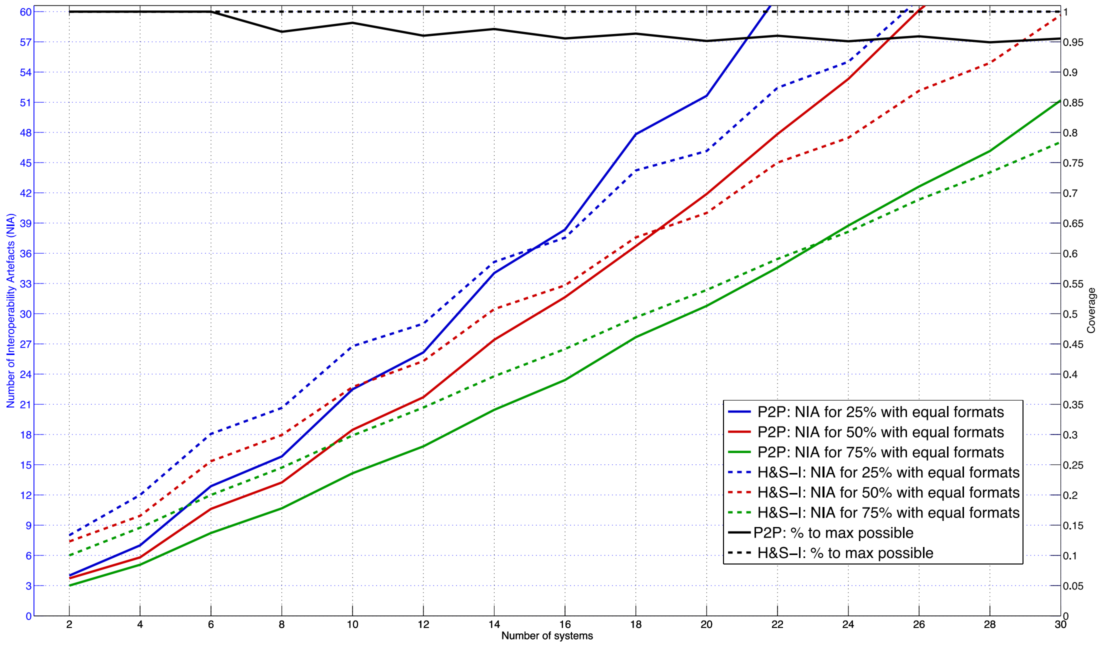


Image 84: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 25% of exporters and 75% of importers

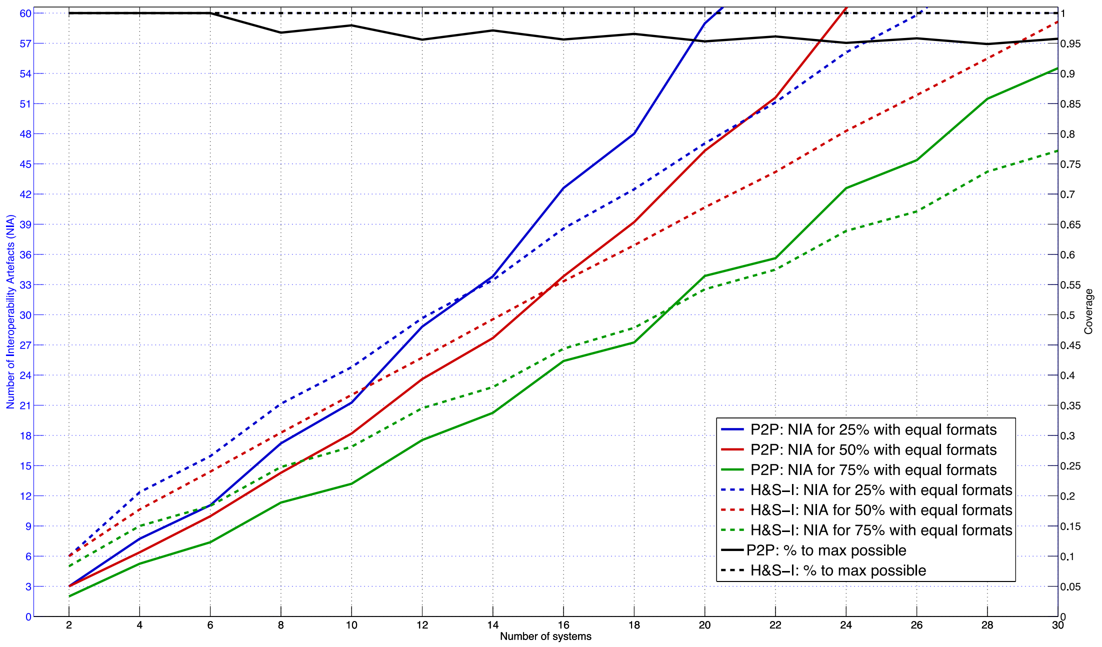


Image 85: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 50% of exporters and 50% of importers

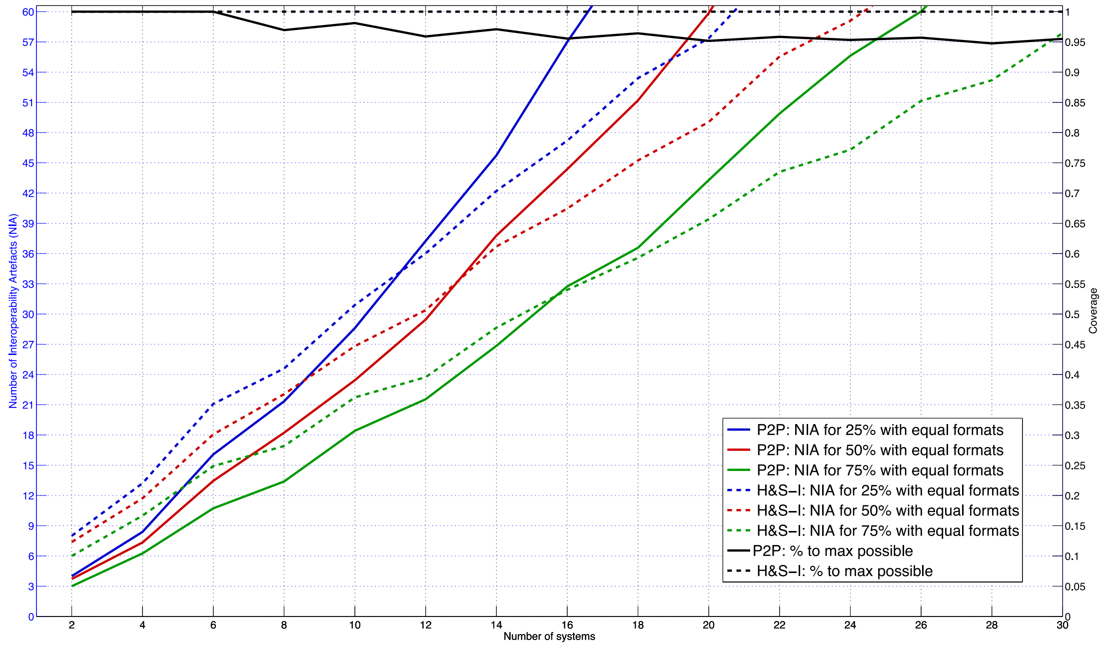


Image 86: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 50% of exporters and 75% of importers

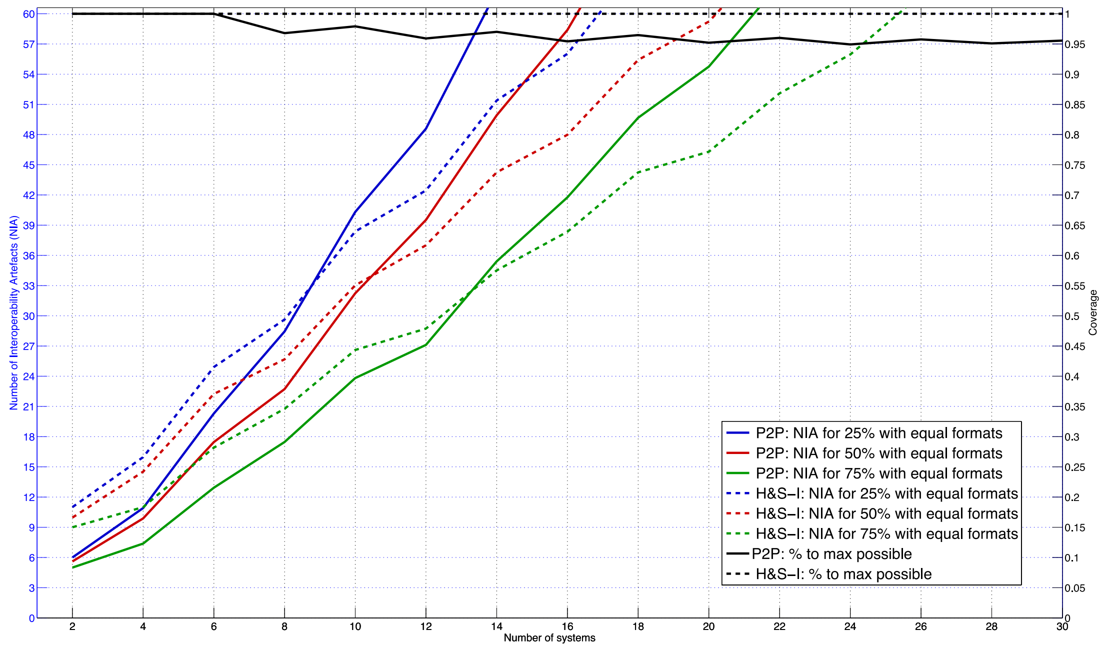


Image 87: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 75% of exporters and 75% of importers

Appendix E:
H&S vs. P2P Interoperability Comparison Graphics

In this appendix are presented the 12 graphics, generated to enable the comparison of between the Hub-and-Spoke Interoperability and the Point-to-Point Interoperability approaches. These graphics result from the combination of five parameters: number of systems, percentage of modifiable systems, percentage of systems with same formats, percentage of systems with export capabilities and percentage of systems with import capabilities; being the results grouped by equal values of percentage of modifiable systems and by the pair of percentage of export and import capabilities. The number of systems varies from 5 to 115 with a step of five. All the other four parameters assume the values 25%, 50% and 75%, being performed all the possible combinations, assuring that the sum of the percentage of Exporters and Importers is always equal or greater than 100%. The variation of these parameters and the corresponding graphics are identified in the table below.

		Modifiable Systems (%)		
		25	50	75
Exporters/Importers (%)	25/75	Image 88	Image 92	Image 96
	50/50	Image 89	Image 93	Image 97
	50/75	Image 90	Image 94	Image 98
	75/75	Image 91	Image 95	Image 99

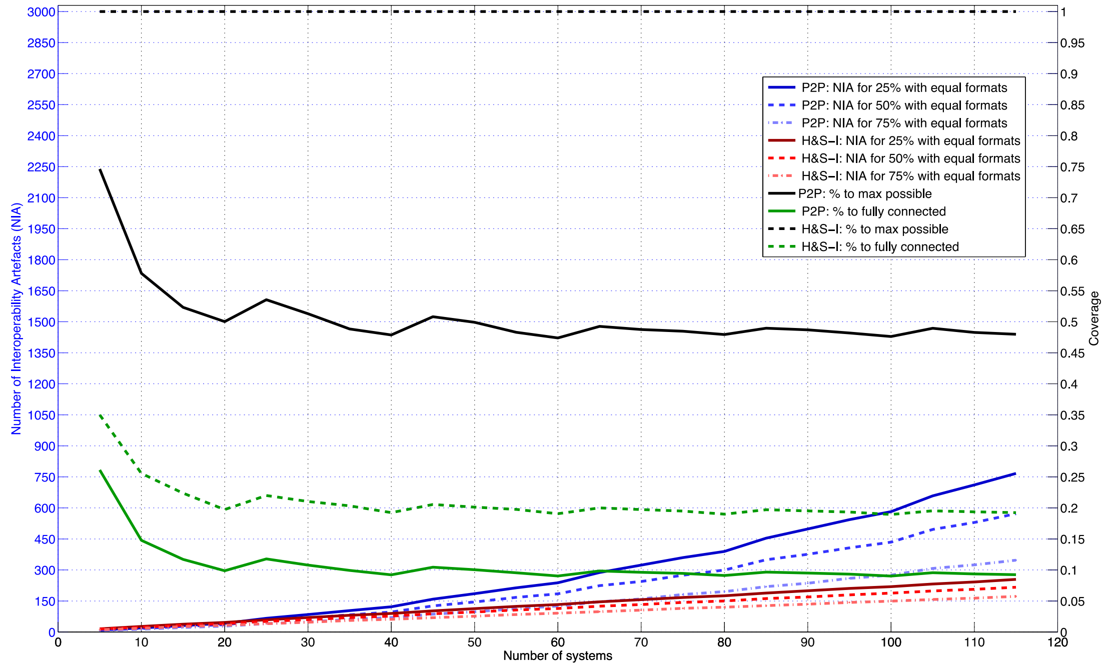


Image 88: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 25% of exporters and 75% of importers

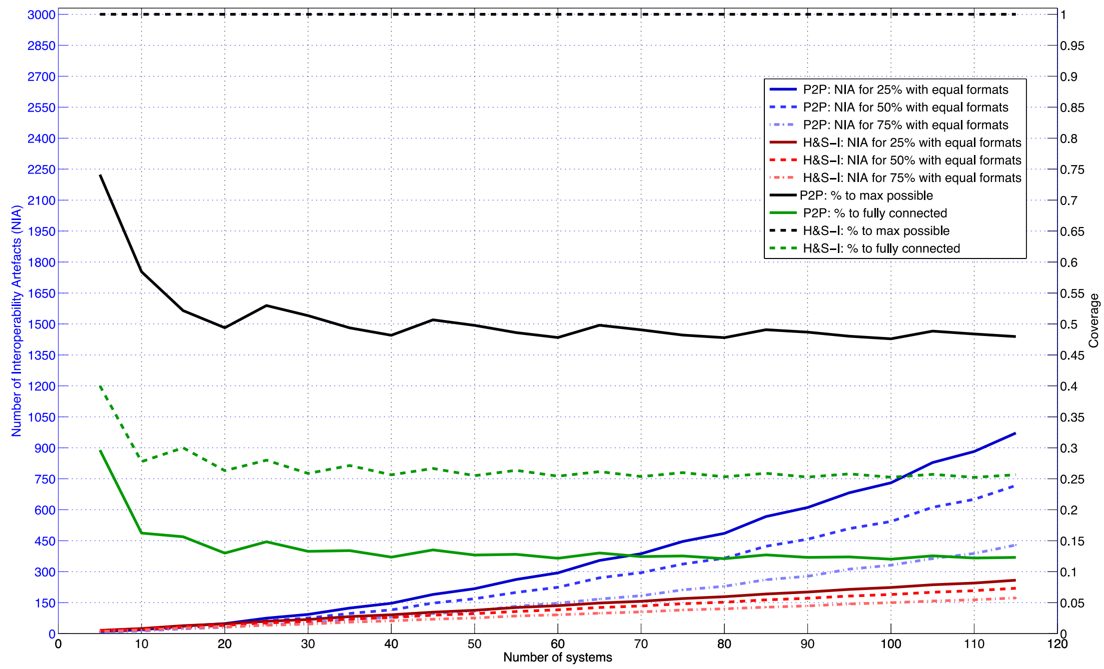


Image 89: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 50% of exporters and 50% of importers

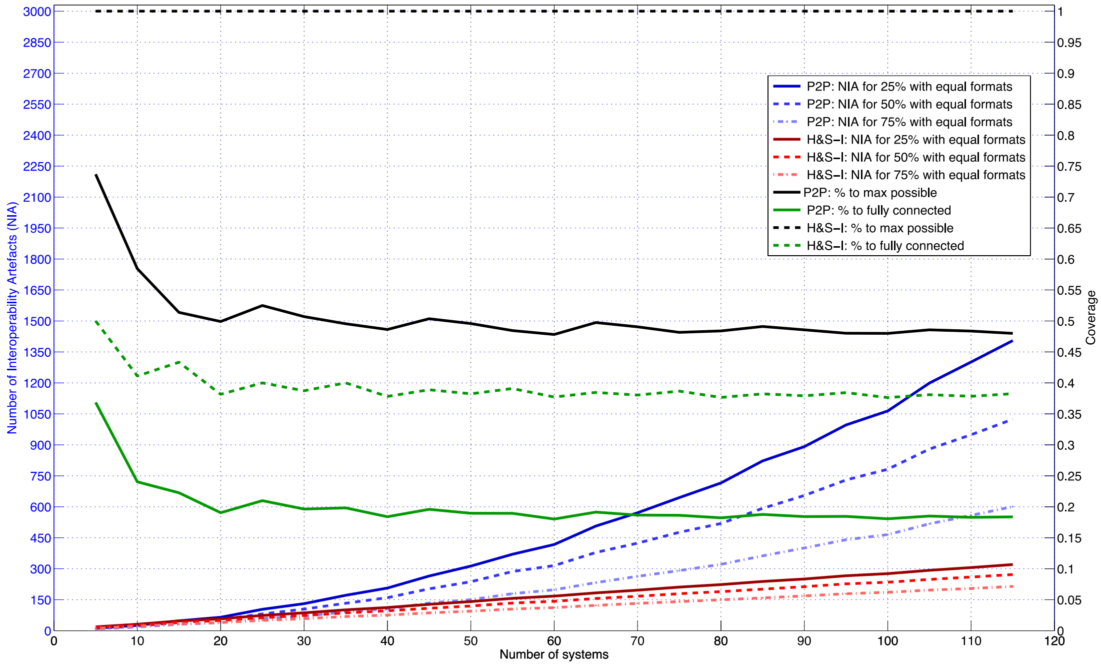


Image 90: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 50% of exporters and 75% of importers

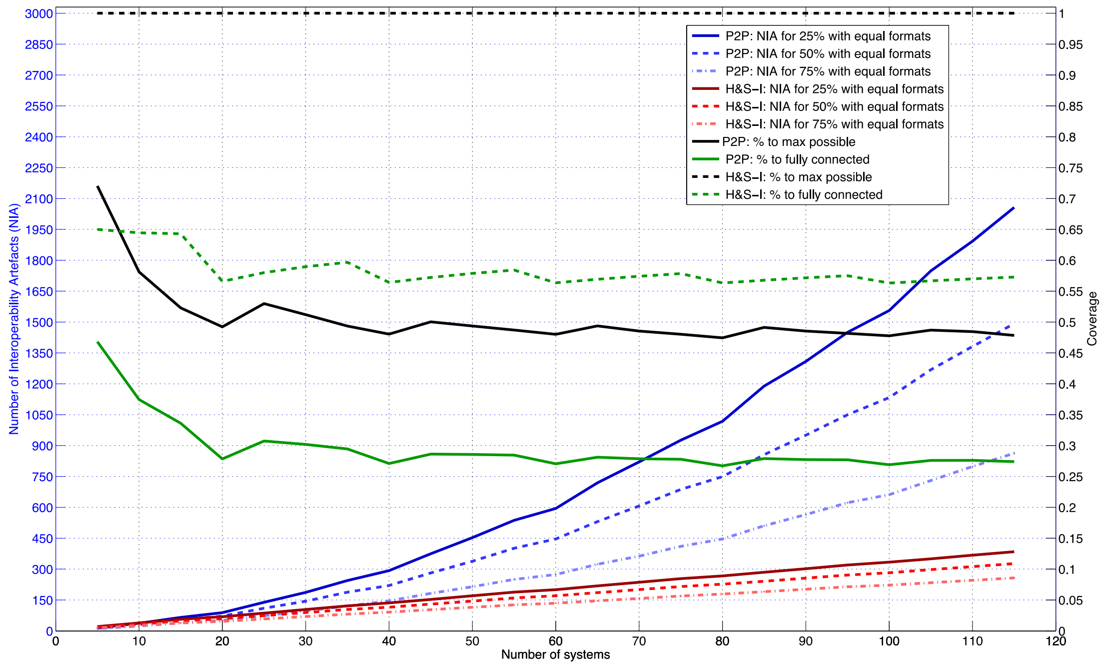


Image 91: Coverage and NIA of P2P vs. H&S Interoperability for 25% of modifiable systems, 75% of exporters and 75% of importers

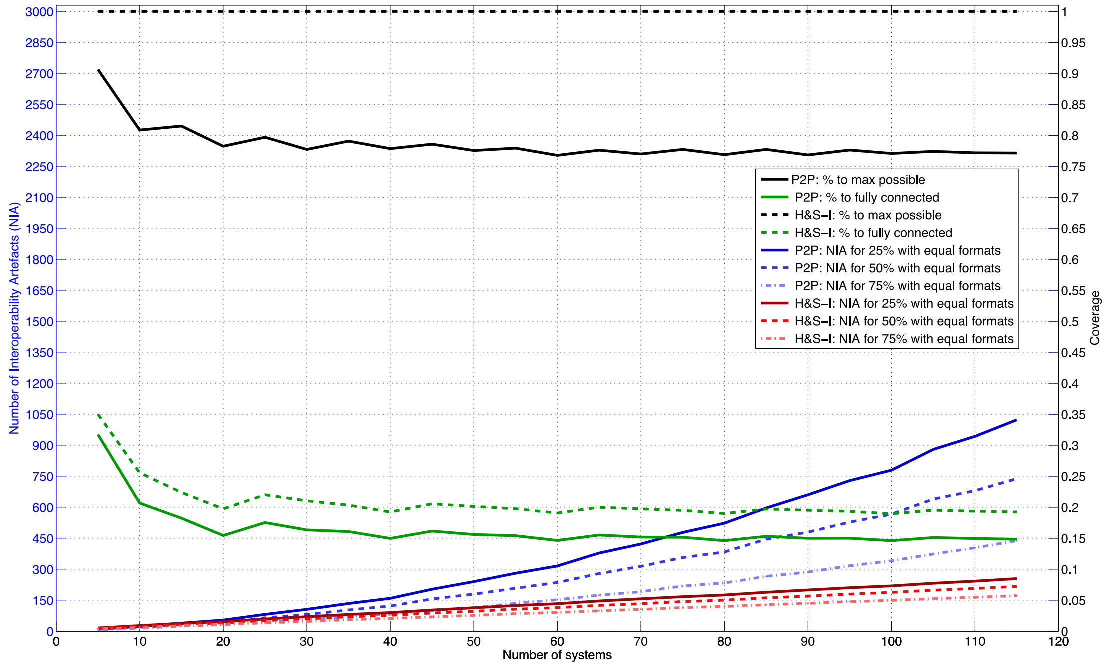


Image 92: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 25% of exporters and 75% of importers

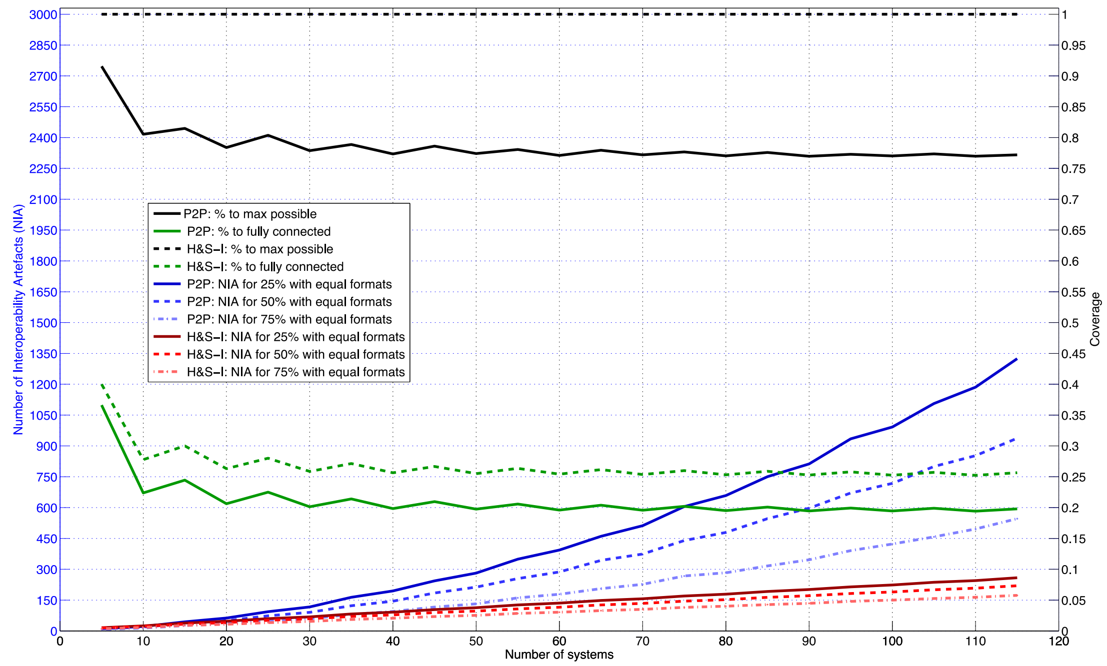


Image 93: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 50% of exporters and 50% of importers

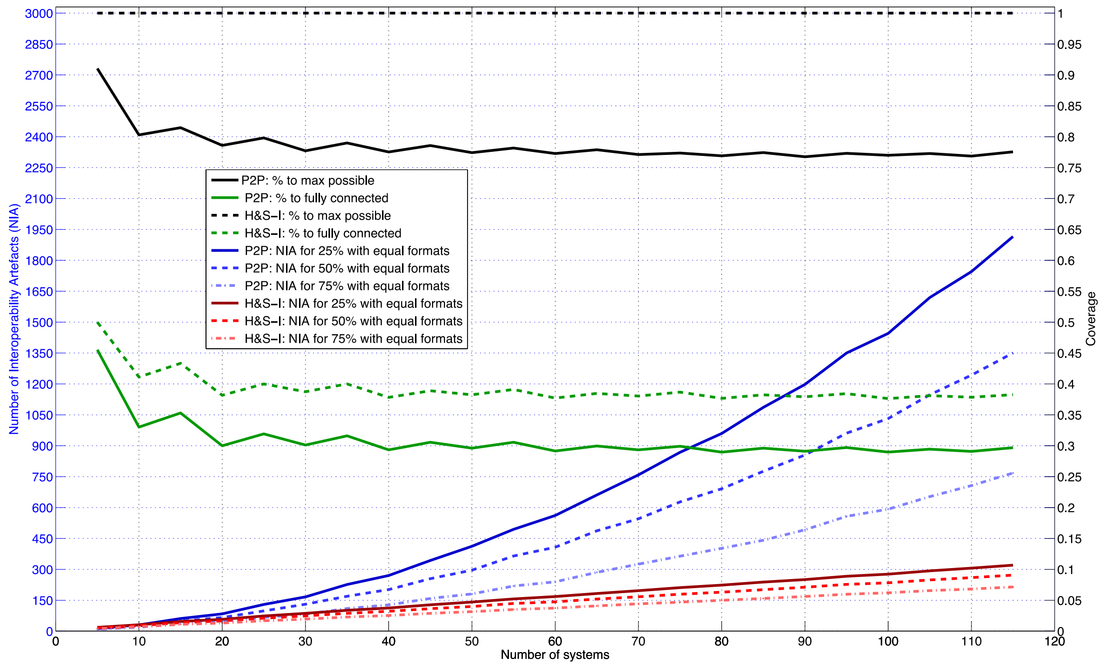


Image 94: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 50% of exporters and 75% of importers

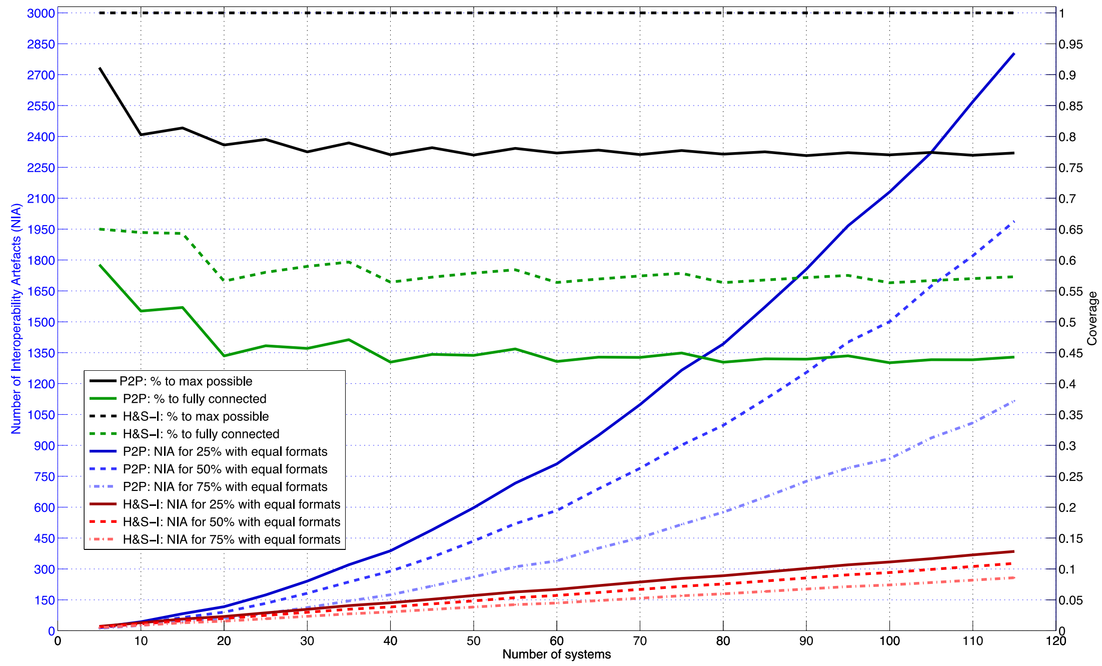


Image 95: Coverage and NIA of P2P vs. H&S Interoperability for 50% of modifiable systems, 75% of exporters and 75% of importers

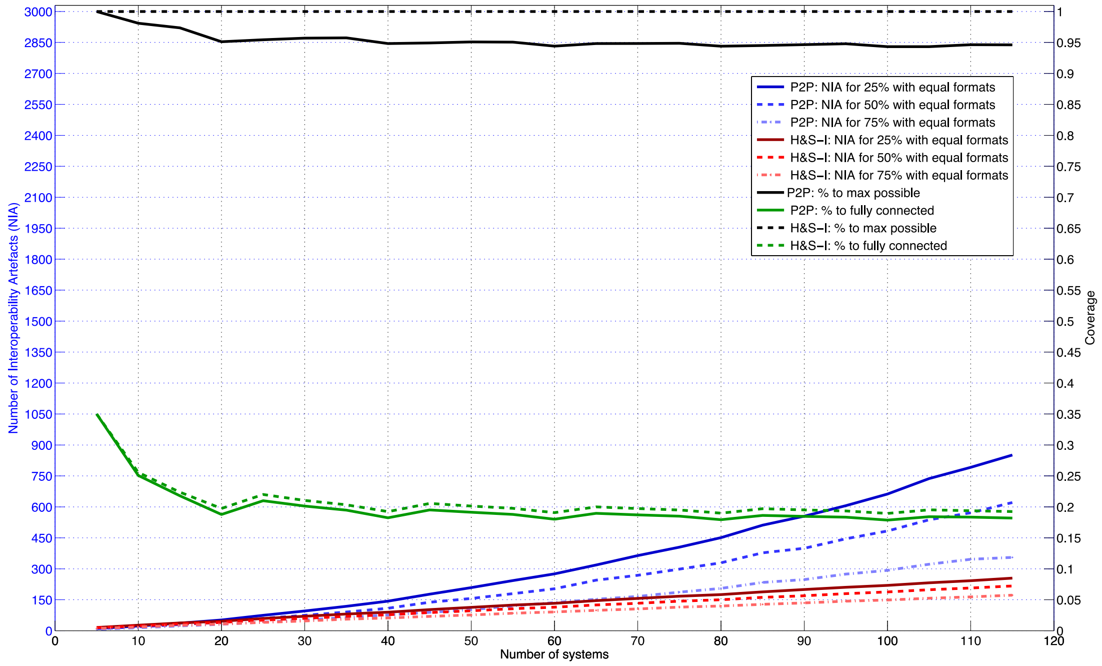


Image 96: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 25% of exporters and 75% of importers

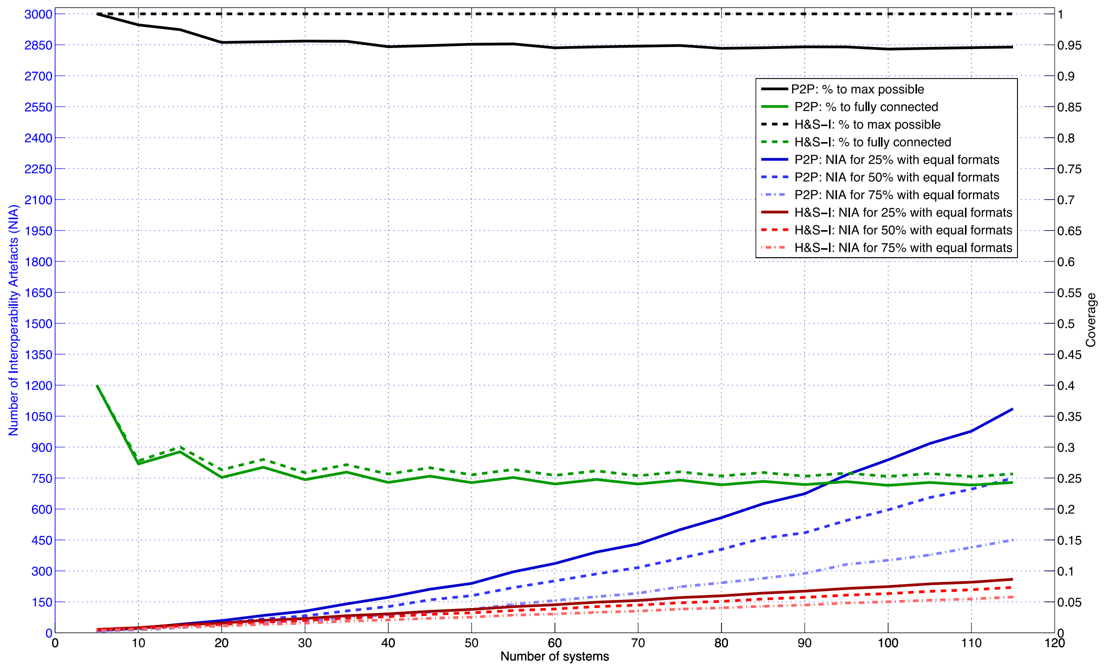


Image 97: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 50% of exporters and 50% of importers

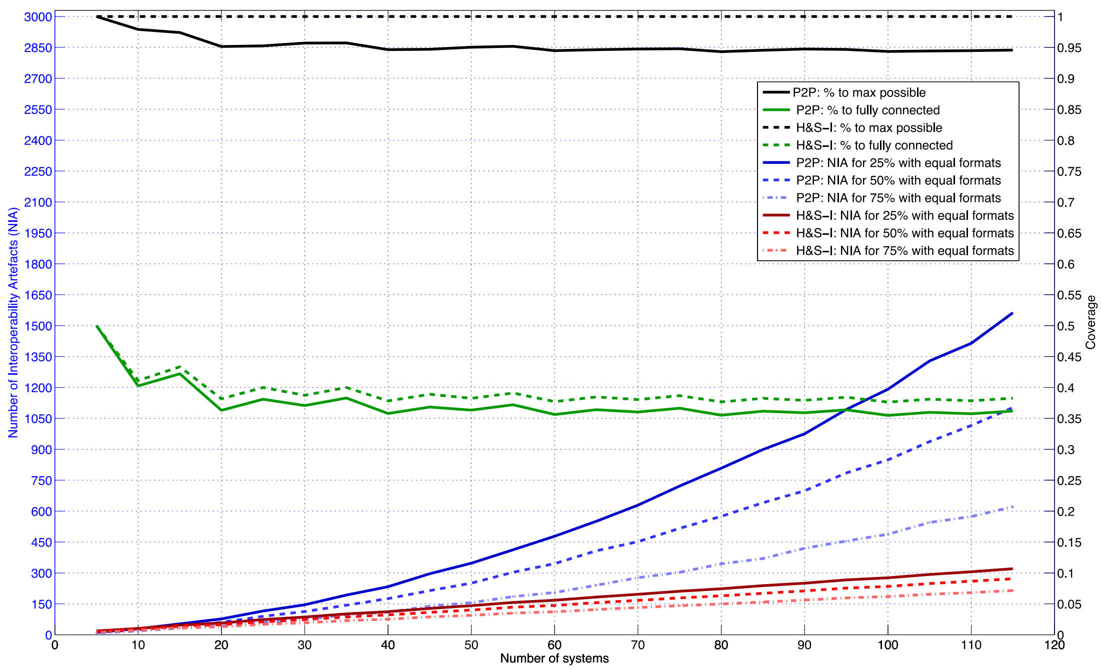


Image 98: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 50% of exporters and 75% of importers

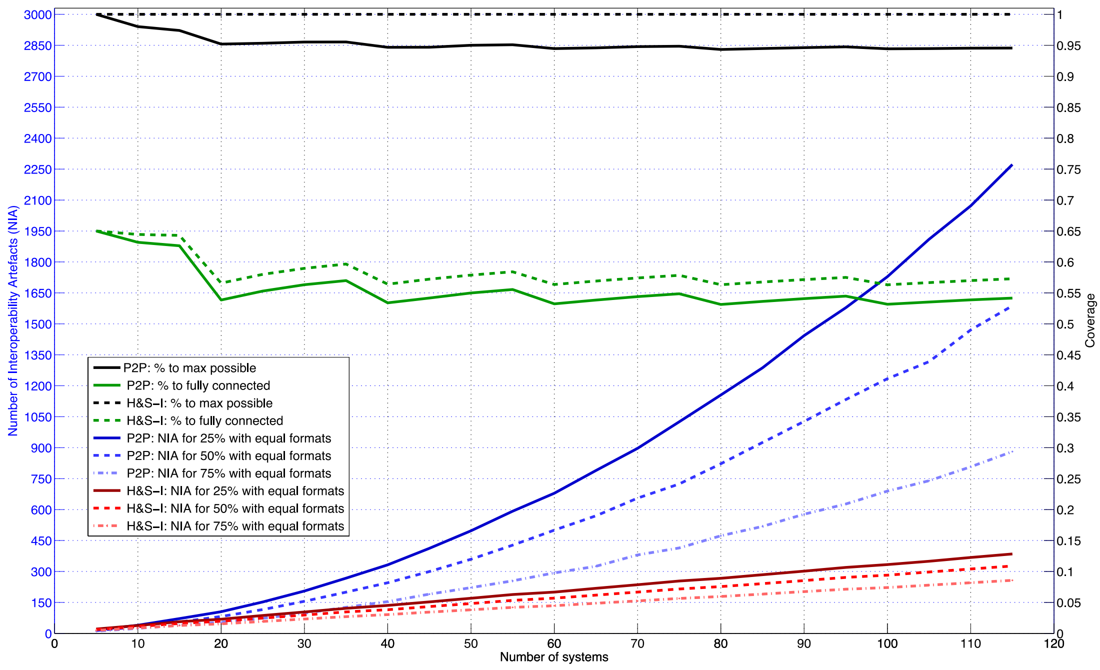


Image 99: Coverage and NIA of P2P vs. H&S Interoperability for 75% of modifiable systems, 75% of exporters and 75% of importers