Luís Alexandre Ferreira da Silva

Master

A Pattern-Based Approach to Scaffold the IT Infrastructure Design Process

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy in
Computer Science

Adviser: Doctor Fernando Manuel Pereira da Costa Brito e Abreu, Researcher at ISTAR-IUL,
Associate Professor, DCTI, ISCTE-IUL

Co-adviser: Doctor Miguel Carlos Pacheco Afonso Goulão,
Researcher at NOVA-LINCS,
Assistant Professor, DI, FCT/UNL

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

December, 2017

[This page has been intentionally left blank]

A Pattern-Based Approach to Scaffold the IT Infrastructure Design Process

[This page has been intentionally left blank]

To Luzia, Tomás and Beatriz

[This page has been intentionally left blank]

Acknowledgements

Firstly, I would like to thank and express my deep and sincere gratitude to my advisor Professor Fernando Brito e Abreu for the continuous support and advices for so many years. He has been a source of encouragement and friendship, for which I am grateful. I would like also to thank to my co-advisor Professor Miguel Goulão for his advices and support in the preparation of this dissertation. Based on their guidance, wisdom and patience I could not have imagined having better advisors for my Ph.D.

Besides my advisors, I would like to thank the rest of my thesis advisory committee: Professor João Bernardo de Sena Esteves Falcão e Cunha, Professor Ana Maria Dinis Moreira, and Professor Pedro Manuel Moreira Vaz Antunes de Sousa, for their insightful advices and comments that allowed me to see my research from different perspectives.

I would like to express my gratitude to the coordinators of the PhD Program in Computer Science, Professor Pedro Manuel Corrêa Calvente Barahona and Professor Nuno Manuel Robalo Correia which was also my Tutor. I express my gratitude also to the teachers and colleagues of the *Departamento de Informática of the Faculdade de Ciências e Tecnologia*, *Universidade Nova de Lisboa* (FCT/UNL), mainly to Anacleto Correia and Sérgio Bryton for their patience to listen and providing valuable comments in the initial phases of my research.

I would like to thank Professor Olga Romão for her help and pragmatism during the review of the survey and Patrick Langlois for letting us publish the online survey in the LinkedIn Infrastructure Architect community. I would like to thank also to all the survey respondents and participants in the controlled experiment.

Last, but not the least, I would like to thank my parents for all their love, support and encouragement and for always believing in me. A special thank to my brother that during the secondary education, changed my path in a very positive way by transferring me to a school that allowed me to do the first steps in Computer Science. And most of all for my loving, supportive, encouraging, and patient wife during the final stages of this Ph.D. Thank you.

[This page has been intentionally left blank]

Abstract

Context. The design of Information Technology (IT) infrastructures is a challenging task since it implies proficiency in several areas that are rarely mastered by a single person, thus raising communication problems among those in charge of conceiving, deploying, operating and maintaining/managing them. Most IT infrastructure designs are based on proprietary models, known as *blueprints* or *product-oriented architectures*, defined by vendors to facilitate the configuration of a particular solution, based upon their services and products portfolio. Existing *blueprints* can be facilitators in the design of solutions for a particular vendor or technology. However, since organizations may have infrastructure components from multiple vendors, the use of *blueprints* aligned with commercial product(s) may cause integration problems among these components and can lead to *vendor lock-in*. Additionally, these *blueprints* have a short lifecycle, due to their association with product version(s) or a specific technology, which hampers their usage as a tool for the reuse of IT infrastructure knowledge.

Objectives. The objectives of this dissertation are *(i)* to mitigate the inability to reuse knowledge in terms of best practices in the design of IT infrastructures and, *(ii)* to simplify the usage of this knowledge, making the IT infrastructure designs simpler, quicker and better documented, while facilitating the integration of components from different vendors and minimizing the communication problems between teams.

Method. We conducted an online survey and performed a systematic literature review to support the state of the art and to provide evidence that this research was relevant and had not been conducted before. A model-driven approach was also used for the formalization and empirical validation of well-formedness rules to enhance the overall process of designing IT infrastructures. To simplify and support the design process, a modeling tool, including its abstract and concrete syntaxes was also extended to include the main contributions of this dissertation.

Results. We obtained 123 responses to the online survey. Their majority were from people with more than 15 years experience with IT infrastructures. The respondents confirmed our claims regarding the lack of formality and documentation problems on knowledge transfer and only 19% considered that their current practices to represent IT Infrastructures are efficient. A language for modeling IT Infrastructures including an abstract and concrete syntax is proposed to address the problem of informality in their design. A catalog of IT Infrastructure patterns is also proposed to allow expressing best practices in their design. The modeling tool was also evaluated and according to 84% of the respondents, this approach decreases the effort associated with IT infrastructure design and 89% considered that the use of a repository with infrastructure patterns, will help to improve the overall quality of IT infrastructures representations. A controlled experiment was also performed to assess the effectiveness of both the proposed language and the pattern-based IT infrastructure design process supported by the tool.

Conclusion. With this work, we contribute to improve the current state of the art in the design of IT infrastructures replacing the ad-hoc methods with more formal ones to address the problems of ambiguity, traceability and documentation, among others, that characterize most of IT infrastructure representations.

Categories and Subject Descriptors:C.0 [**Computer Systems Organization**]: System architecture; D.2.10 [**Software Engineering**]: Design-Methodologies; D.2.11 [**Software Engineering**]: Software Architectures-Patterns.

Keywords: Information Technology, IT Infrastructure, UML Profile, Design Patterns.

Resumo

Contexto. O desenho de infraestruturas de Tecnologias de Informação (TI) consiste num desafio complexo porque implica proficiência em muitas áreas que raramente são dominadas por uma só pessoa, o que acarreta problemas de comunicação entre aqueles que são responsáveis pela concepção, instalação, operação e manutenção/gestão daquelas infraestruturas. As infraestruturas são criadas maioritariamente com base em modelos proprietários, frequentemente denominados de *blueprints*, definidos por fabricantes que possuem como base a sua oferta de produtos e serviços. Os *blueprints* existentes podem ser facilitadores no desenho de soluções de uma determinada tecnologia ou fabricante. Contudo, uma vez que as organizações possuem componentes de infraestrutura de múltiplos fabricantes, a utilização de *blueprints* alinhados com produto(s) pode causar desafios na integração entre estes componentes e a que fiquem dependentes de um só fabricante. Adicionalmente, os *blueprints* possuem um ciclo de vida curto, porque estão associados a uma determinada versão de produto(s) ou tecnologia, o que dificulta a sua utilização como uma ferramenta para reutilização de conhecimento de infraestruturas de TI.

Objetivos. Os objetivos desta dissertação são *(i)* mitigar a incapacidade de reutilizar o conhecimento em termos de boas práticas sobre o desenho de infraestruturas de TI, *(ii)* simplificar a utilização desse conhecimento, tornando o processo de desenho de infraestruturas mais simples, mais rápido, melhor documentado, facilitando a integração de componentes de diferentes fabricantes e minimizando os problemas de comunicação entre as equipas.

Métodos. Conduzimos um questionário em linha e realizámos uma revisão sistemática como forma de suportar o estado da arte e fornecer evidências que esta investigação era relevante e não tinha sido conduzida anteriormente. Foi seguida uma abordagem orientada por modelos para a formalização e validação empírica de regras de boa formação com o objetivo de melhorar o processo de desenho de infraestruturas de TI. Para simplificar e suportar o processo de desenho uma ferramenta de modelação com uma linguagem abstrata e concreta foi também estendida de forma a incluir as principais contribuições desta dissertação.

Resultados. Foram obtidas 123 respostas ao questionário, sendo que a sua maioria foi oriunda de pessoas com mais de 15 anos de experiência com infraestruturas de TI. Os entrevistados confirmaram a falta de formalidade e de documentação, problemas na transferência de conhecimento e apenas 19% consideraram que as práticas que usam para representar infraestruturas de TI são eficientes. É proposta uma linguagem para a modelação de infraestruturas de TI incluindo as suas sintaxes abstrata e concreta, para resolver o problema da informalidade no seu desenho. É também proposto um catálogo de padrões de infraestrutura para permitir expressar as melhores práticas no seu desenho. Para simplificar o processo, uma ferramenta de modelação foi estendida para incluir estas propostas. A ferramenta de modelação foi também avaliada e de acordo com 84% dos entrevistados, esta abordagem diminui o esforço associado com o desenho de infraestruturas de TI e 89% considerou mesmo que o uso de um repositório com padrões de infraestrutura, ajuda a melhorar a qualidade geral das representações de infraestruturas de TI. Foi também realizada uma experiência controlada para avaliar a eficácia tanto da linguagem proposta como do processo de desenho de infraestruturas de TI suportado pela ferramenta.

Conclusão. Com este trabalho contribuímos para melhorar o estado da arte na representação de infraestruturas de TI, substituindo os processos de desenho ad-hoc por processos mais formais para endereçar os problemas de ambiguidade, rastreabilidade, documentação, entre outros, que caracterizam a maioria das representações de infraestruturas de TI.

Categorias e Descritores de Assunto: C.0 [**Computer Systems Organization**]: System architecture;D.2.10 [**Software Engineering**]: Design-Methodologies; D.2.11 [**Software Engineering**]: Software Architectures-Patterns.

Palavras-chave: Tecnologias de Informação, Infraestrutura de TI, Perfil de UML, Padrões de Desenho.

Contents

[This page has been intentionally left blank]

List of Figures

[This page has been intentionally left blank]

List of Tables

[This page has been intentionally left blank]

Listings

Acronyms

ABC . . . . Activity-based costing

ACM . . . . Association for Computing Machinery

BGP . . . . Border Gateway Protocol

BPMN . . . . Business Process Modeling and Notation

BYOD . . . . Bring Your Own Device

CASE . . . . Computer Aided Software Engineering

CASP . . . . Critical Appraisal Skills Programme

CCM . . . . Cyclomatic Complexity Metric

CI . . . . Configuration Items

CIM . . . . Common Information Model

CIO . . . . Chief Information Officer

CMDB . . . . Configuration Management DataBase

CNA . . . . Converged Network Adapter

CNC . . . . Coefficient of Network Complexity

CRM . . . . Customer Relationship Management

CSV . . . . Comma Separated Values

CWM . . . . Common Warehouse Metamodel

DAS . . . . Direct Attached Storage

DBLP . . . . Digital Bibliography and Library Project

DMTF . . . . Distributed Management Task Force

DMZ . . . . DeMilitarized Zone

DSL . . . . Domain Specific Language

EA . . . . Enterprise Architect

EOS . . . . Extensible Operating System

ERIC . . . . Educational Resources Information Center

ERP . . . . Enterprise Resource Planning

FC . . . . Fiber Channel

FCoE . . . . Fiber Channel over Ethernet

FMC . . . . Fundamental Modeling Concepts

GoF . . . . Gang of Four

GOODLY . . . . Generic Object Oriented Design Language? Yes!

HA . . . . High Availability

HBA . . . . Host Bus Adapter

HKM . . . . Henry and Kafura Metric

ICMP . . . . Internet Control Message Protocol

IEEE . . . . Institute of Electrical and Electronics Engineers

IoT . . . . Internet of Things

iSCSI . . . . internet Small Computer System Interface

ISDN . . . . Integrated Services Digital Network

ISP . . . . Internet Service Provider

ITI . . . . Information Technology Infrastructure

ITIL . . . . Information Technology Infrastructure Library

itiLib . . . . itiPM Library for ITIs

itiPM . . . . ITI Profile Metamodel

LAN . . . . Local Area Network

LBFO . . . . Load Balancing and FailOver

LDAP . . . . Lightweight Directory Access Protocol

M2DM . . . . Metamodel Driven Measurement

MDG . . . . Model Driven Generation

MOF . . . . Meta-Object Facility

MPLS . . . . MultiProtocol Label Switching

NAS . . . . Network Access Storage

OCL . . . . Object Constraint Language

OGC . . . . Office for Government Commerce

OMG . . . . Object Management Group

OO . . . . Object-Oriented

OWL . . . . Ontology Web Language

PC . . . . Personal Computer

PLoP . . . . Pattern Languages of Programs

POSA . . . . Pattern-Oriented Software Architecture

RM-ANOVA . . . . Repeated-Measures Analysis Of Variance

RTF . . . . Rich Text Format

SAN . . . . Storage Area Network

SLR . . . . Systematic Literature Review

SNMP . . . . Simple Network Management Protocol

SPAM . . . . Sending and Posting Advertisement in Mass

TCO . . . . Total Cost of Ownership

TCP/IP . . . . Transmission Control Protocol / Internet Protocol

TMF . . . . TeleManagement Forum

TOGAF . . . . The Open Group Architecture Framework

UML . . . . Unified Modeling Language

UoD . . . . Universe of Discourse

USE . . . . UML-based Specification Environment

VPN . . . . Virtual Private Network

WAN . . . . Wide Area Network

WFR . . . . Well-Formedness Rule

WSDM . . . . Web Services Distributed Management

XDE . . . . eXtended Development Environment

XMI . . . . XML Metadata Interchange

XML . . . . eXtensible Markup Language

Chapter 1

IT Infrastructure

Contents

1.1 Introduction

The term Information Technology Infrastructure (ITI)[1] is not new [MB89; Tur91], but surprisingly there is not a generally accepted definition for it [Gal08a; Gal08b; Gar13; Nyr06; OGC07]. According to version 3 of the Information Technology Infrastructure

[1] Some other common terms used to refer to ITI are Information Infrastructure, Technology Infrastructure, Technical Infrastructure, Infrastructure Architecture, Enterprise Infrastructure Architecture, Technology Infrastructure Architecture, Service Oriented Infrastructure, Cloud Infrastructure, Computing Infrastructure.

Library (ITIL) [Can11; Hun11; Llo11; Ran11; Ste11] from Office for Government Commerce (OGC), infrastructure includes *all of the hardware, software, networks, facilities, etc., that are required to develop, test, deliver, monitor, control or support IT Services* [OGC07], but not the associated people, processes and documentation. Aligned with ITIL and to frame the scope of this dissertation, we consider that the term ITI covers the following four domains:

- **Computing and Hardware.** This domain covers the computing and hardware platform such as mainframes, servers, racks, workstations and other physical components;
- **Network.** This domain comprises a set of aspects related with devices connectivity, such as transmission media, network protocols, topologies and includes network components such as routers, switches, bridges and hubs;
- **Storage.** This domain includes storage components such as storage arrays, storage network devices and storage protocols, such as fiber channel or internet Small Computer System Interface (iSCSI);
- **Software.** This domain encompasses the infrastructure software, such as hypervisors, operating systems, antivirus or backup software, among others.

As with ITIL, we did not consider the human factor. We also left, outside the scope of this research work, some other aspects related to the delivery of IT services, such as software architecture and design.

1.1.1 The Concept of ITI

ITIs within organizations have long been understood in terms of resources such as servers, switches, routers, firewalls and operating systems, among other components working together to provide the foundation for business applications. [MB89] were the firsts to define ITI elements. Before the 1990s, an ITI was defined as a set of technology components such hardware, operating systems, networks and data [Kee91]. Since the mid-1990s, the definition of IT infrastructure was extended to consider management factors and was generally described as shared ITI resources and capabilities that comprehend four infrastructure elements:

- **IT Infrastructure Components.** These components are the foundations of ITIs and include hardware, software, networks and communication, data and other components to support business applications.

- **Shared IT Infrastructure Services.** These are the shared ITI services such as network services, communication services, data services, systems management services that are consumed by the business applications. From a business perspective, these services are viewed as building blocks for the applications business models.

- **Human IT Infrastructure.** Includes organizational knowledge, skills and experience on ITI in a way that can be understood by business people.

- **Business Applications.** These business applications consume ITI resources and services to directly support business processes in software applications.

The business processes are only possible because they are supported by ITI and applications [Ear89] [XK04]. Figure 1.1 illustrates the infrastructure elements of an ITI.

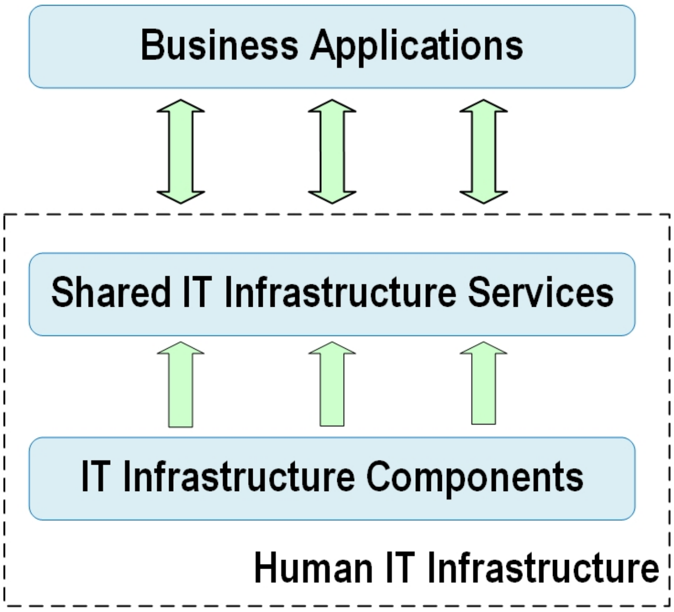


Figure 1.1: Infrastructure elements adapted from [MB89]

1.1.2 The Evolution of ITIs

ITIs as they exist today are an outgrowth of over 60 years of evolution in computing platforms. The evolution of ITIs can be classified in five different stages or eras, each representing specific configurations and infrastructure elements as follows:

1. **General purpose Mainframe and Mini computer (Since 1959).** The mainframe was invented by IBM and was the first centralized computer with networks of terminals concentrated in the computing department. Before that, there was an electronic accounting machine that began to replace humans in the accounting department. The mainframe evolved and continues to be used today to process huge amounts of data and to support important business processes. The mainframe continues to be a strong, viable component in many ITIs and IBM remains the largest supplier of mainframe computing.

2. **Personal Computer (Since 1981).** The Personal Computer (PC) from IBM in 1981 was considered the first PC because it was the first to become widely adopted. These computers used DOS operating system and later Microsoft Windows with Intel and became a standard desktop with the proliferation of PCs in the 1980s and early

1990s. These PCs supported a set of programs that were valuable to home and corporate users such as spreadsheets, word processors, electronic presentations and some other data management programs. The advances developed for personal computer in the home have given rise to much of the advances in corporate computing since employees demanded increased use of computers in the workplace. Initially these PCs were standalone systems until PC operating system software in the 1990s made it possible to link them into networks. In this era, IT started to help organizations increasing the number of opportunities with suppliers and customers, based on new products and services [Bha+10].

3. **Client / Server (Since 1983).** With the proliferation of PCs, they became more powerful and cheaper and organizations began using them as a replacement of minicomputers and mainframes by connecting them in a client / server model with the processing work split between these two types of machines. The client represents the user workspace and the entry point to the network while the server (that could be a mainframe) typically process and store shared data, manage clients, among other network activities. Today servers are powerful versions of personal computers, based on inexpensive chips and using multiple processors in a single computer box. Small organizations typically defined two-tiered client/server architectures (where several clients are connected to one server) while large organizations started to use complex multi-tiered client/server architectures that have several clients connected to several different levels of servers. This model allows the distribution of computing work across a number of smaller, inexpensive machines that cost much less than minicomputers, or centralized mainframe systems, resulting in an explosion in computing power and application throughout the organization.

4. **Enterprise Computing (Since 1992).** The success of complex multi-tiered client/server architectures created a new set of problems for organizations since it became difficult to integrate all of their local area networks (LANs) into a single, coherent corporate computing environment that can span multiple departments and different geographic locations. Due to this problem, in the early 1990s, organizations focused on networking standards and software tools that could integrate disparate networks. The Internet developed a standard trusted communications environment based on Transmission Control Protocol / Internet Protocol (TCP/IP) and later organizations started to use it to connect their disparate networks. These changes allowed connections among ITIs with different types and brands of computer hardware and smaller networks into an enterprise-wide network so that information can flow freely across the organization and between the organization and other organizations. The corporate network consisted now in mainframes, servers, PCs, mobile phones, hand-held devices connected to public infrastructures such as the telephone system, the Internet, and public network services. These changes in networking allowed the integration of multiple software solutions and a truly

integrated computing and IT services platform for the management of global enterprises. In this era ITI was being defined as a set of shared, tangible, technological resources including platform technologies, networks and telecommunication technologies, data, and software applications [Qi+08].

5. **Cloud Computing (Since 2000).** This is a computing model that provides shared processing resources and data to computers and other devices on demand. Cloud computing has become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. The main enabling technology for cloud computing is virtualization which completely reshape ITIs and the IT industry. Virtualization consists in a software that can emulate multiple "virtual computers" in the same physical hardware, each of which can be easily used and managed to perform computing tasks such as physical computers. Today with Cloud Computing, the resources are managed in terms of compute, storage and networks and each requires different capabilities.

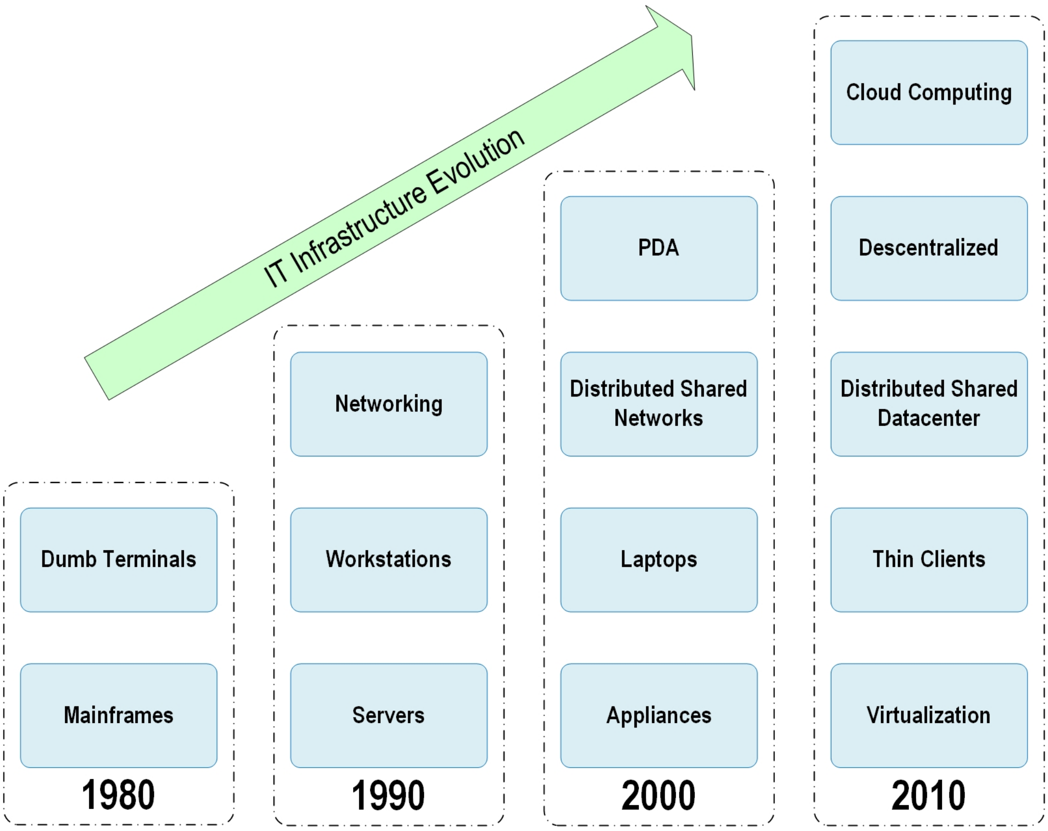


Figure 1.2: Evolution of IT

It is interesting to see that there are a significant number of ITIs that still contains technologies from all or some of these stages. As an example, there are organizations that have some business processes supported by traditional mainframes and already adopted cloud computing for other business processes. Figure 1.2 represents some of the stages of ITI evolution.

The so-called "digital transformation" is changing the way business gets done with customers demanding a digital and richer experience. A digital business is less about creating new products and solutions and more about empowering customers to search and buy differently. This implies a digital transformation and significant changes in existing ITIs. The digital transformation is an important opportunity for organizations open to change and to adopt new innovative technologies that can take ITIs to a new level of agility, efficiency, and ultimately shape growth and provide the catalyst for new business models, products, services and experiences.

1.1.3 Assessment of ITIs

As organizations grow, their ITIs grows along with them. But often that growth is uneven, driven as much by the conditions under which they operate, as by the model they aspire to. Most times new business requirements need considerable changes in infrastructures that must be implemented in the shortest possible time, to meet business deadlines. In some organizations, this pressure leads to wrong infrastructures, increases complexity, decreases the effectiveness and efficiency resulting in an infrastructure more difficult to manage, new components without integration with existing ones, waste of resources and delays, among other challenges.

This fusion between business and technology requires the expertise of both business and IT professionals that should support their decisions based on concrete information to better align business with infrastructures. Both should understand “what is the ITI currently?”, “what they want it to be?” and turn what they have, into what they want in a disciplined way. Better understanding of ITIs can provide valuable information to business and IT professionals and is crucial to support their decisions regarding the growth of ITIs. Better understanding can be accomplished with an ITI evaluating process created during our research activities around ITIs. Some examples of benefits of that assessment are:

- Define and maintain ITI operations and administrative policies;
- Check if best practices are being applied across the entire infrastructure;
- Understand better the impact of changes of ITIs in other systems;
- Simplify infrastructure management;
- Evaluate emerging technologies and business potential or impact through complexity analysis and evaluation;
- Analyze and predict ITI growth over the years;
- Help defining an ITI strategy, planning, architecture and optimization to meet business goals and objectives.

1.1.4 ITI Demand for Flexibility

More than ever, in the world's current economic scenario, where organizations need to innovate and change constantly, the characteristics of ITIs (in terms of scalability, robustness, ease of change and documentation, among other aspects) may represent the difference between organizations success or failure. The need for more flexibility and agility, and trends such as cloud computing, changed the way organizations look into ITIs [Kha11].

Having the right ITI, at the right time, to support new business requirements is a challenging task, because new initiatives emerge unpredictably [Nyr06]. New requirements for a particular solution may lead to considerable changes in existing ITIs that must be planned and implemented quickly. This pressure, coupled with the lack of robust design processes, driven by non-experts, leads frequently to availability, integration or capacity problems, waste of resources and ultimately in a more complex ITI to manage, which decreases its effectiveness and efficiency [PG07].

The flexibility of an ITI is required to survive in the present competitive environment and can be described as the ability of an ITI to efficiently adapt to the organization business goals [Qi+08]. Before it became widely used, the goal of ITI for some organizations was to help them to generate business advantages. Over the years organizations recognize the importance of effective and flexible ITI [Eva01].

The effectiveness of an ITI can be assessed using criteria such as reliability, operation with low downtime and capacity to efficiently adapt to the changing business requirements.

1.1.5 Digital Transformation

The main goal of ITIs over the last 25 years has been to support the business by keeping corporate data and applications running. The focus on ITIs is changing now from supporting to differentiation with new levels of efficiency, flexibility and automation.

In the digital world where we are living, there is no organization or industry immune to digital disruption. The customers are now interacting with products and solutions in a complete different way.

Organizations of all sizes are focused on ensuring that they stay relevant in their unique competitive markets and they are looking at ways to optimize their operations to drive efficiency to enable new products and business models.

A digital transformation implies rethinking how to add value to customers. For most organizations, digital transformation is a critical step for their future survival and they rely heavily on ITIs and technology innovation to be able to reach and serve their customers.

Notice that besides technology, to be successful in a digital transformation it is important to consider also people and processes.

Digital transformation is happening rapidly, and more and more organizations are embracing the four mega-trends" Cloud, Big Data, Mobile and Social technologies. Each of these trends implies incorporation of new technologies, capabilities and several changes to existing ITIs to support for instance connected devices and other “things” within the Internet of Things (IoT), the growing amount of data, the emergence of advanced analytics, machine learning, artificial intelligence to name a few.

1.1.6 Representing ITIs

Representing or designing ITIs, particularly for large organizations, is a challenging task, mainly because it requires knowledge on existing organization processes, the views of different stakeholders, and the coordination of technical expertise in multiple domains. Most ITIs are created, designed or adapted by consultants, administrators, developers and other individuals with the only purpose of responding to the requirements of a particular business application. Furthermore, large ITIs usually contains hardware, networking and infrastructure software components from different suppliers, that require integration to grant interoperability and provide a variety of services, both within and outside the organization. The resulting infrastructures often look like a patchwork composition.

The design of ITIs has been supported by extensive *blueprints* accompanied by detailed design and implementation guidelines and scripts, among other materials, to assist in the activities of deploying a specific component or solution [Boa98; Sne12; Tro+03]. The use of *blueprints* was grasped as a business opportunity by several IT organizations to standardize typical ITI building blocks, based on their commercial products or services. Some of those companies developed methodological approaches to ITI pattern-based design, by proposing *design blueprints* embodying vendor-specific components [Tro+03] *whitepapers*, reference architectures, blogs and other informal documents. As an example, there are seven design patterns in the whitepaper [Dat09] produced by the organization Arista Networks[2] following is part of the design pattern description: *"Arista´s revolutionary Extensible Operating System (EOS) software system is at the heart..."*, *"Arista´s low-latency switches permit..."*, *"Arista´s top-of-the rack switches provides..."*.

At first glance, it seems that the use of these *blueprints* only brings benefits. However, besides the commercial wording, there are many negative aspects in their adoption. For example, most of them are extensive and hard to read and are also vendor-dependent, which favors the often-undesirable vendor lock-in phenomenon [Tru10] and hampers the ability to build multi-vendor solutions. Since ITIs are normally built with components from multi-vendors that require integration to grant interoperability and provide a variety of services, both within and outside the organization, the *blueprints* often become useless in such heterogeneous ITIs. Complicating matters further, ITIs have similar

[2] http://www.aristanetworks.com/en/company visited in Feb 28, 2017

characteristics to those of legacy software systems: there are no updated models, the introduction of changes often has serious side effects and knowledge reuse about its design and maintenance is scarce since it resides essentially only in the minds of some individuals. ITI staff turnover thus becomes critical because best practices in ITI design are not shared [Mon13].

The aforementioned situation is largely due to the lack of standardized modeling languages capable of representing the many elements used in the design of infrastructures, with the required rigor and detail [Sil+12b]. This leads practitioners to use ad-hoc approaches to modeling, whose lack of precision hampers model validation and the reuse of ITI design best practices.

1.2 Research Drivers

The starting point of our research process was to formulate the main problems we are tackling (section 1.2.1). This drove us to the research questions (section 1.2.2). To answer these questions, we had to formulate some hypotheses. The explanatory theory of the problems posed by the research questions is embodied in the research hypotheses. For validating our research contributions, we had to set up some experiments and surveys that support our conclusions about the benefits and effectiveness of our proposals, as described in section 1.2.3.

1.2.1 Research Problems

The aim of our research work was to tackle the following problems in the realm of ITIs:

RP1. **Informal notation used to represent ITIs.** The current state of practice to represent ITIs is characterized by informal notations. ITIs are becoming more complex with an increasingly high number of critical applications and services and with more internal and external dependencies. This informality leads to several problems, as recognized by some organizations which are looking for alternative approaches.

RP2. **Inability to express best practices in ITIs design.** The lack of formality in ITIs, the use of solutions from a single vendor and the increasing complexity of ITIs are some of the reasons that make the reuse of knowledge in the design of ITIs a challenge. For instance, due to this inability to express best practices, some organizations include in the training process of newcomers the *shadow approach*, which consists in learning in the field from other more experienced colleagues. In the context of software design a similar problem was mitigated using patterns. Chapter 3 provides more details about modeling ITIs and chapter 4 more details about patterns.

RP3. **Absence of methodological support for a pattern-based approach to ITI design.** As in other domains, in the design of ITIs, there are also problems, which sometimes have more than one solution, but not all are optimal. Patterns are an important part of the solution. However as will be detailed in chapter 4, an approach purely based on patterns[3], without a methodology, is not sufficient to achieve the best results and enable ITI people (e.g. architects, consultants, administrators) to use the proposed approach.

The results of the systematic literature review in section 2.2, allow us to corroborate that there are no proposals from the scientific community that can address the problems above. There are also documented evidences that the problems faced by the author in the field and tackled within this dissertation are real industry problems. As an example, Air New Zealand tried to model their ITI to support a datacenter migration but found no appropriate notations for this task [Zha11]. Furthermore, they claim that the Unified Modeling Language (UML) is not suitable, because it was designed for software, and that the representations often performed with *Microsoft Visio*, Excel Spreadsheets and other tools have several limitations. Since their ITI was large (approximately 900 servers, 200 racks, 20 network domains and 70 teams), they decided to setup a team during several months to model the ITI and, in the end, the project was successful.

With some articles already published, we started being contacted by experts in the area, confirming the problems and asking for more information. Some examples received, came from an IT Director that is working as an independent IT Consultant[4] an Enterprise Architect[5] and an Infrastructure Architect[6]:

"This article contains the key to resolving a problem I have been struggling with for years: the precise description of an infrastructure."

Frans Wiggers - August 2015

"I wanted to ask if there has been any further development of these concepts, and if the profile your constructed in Sparx EA is published/available for consumption."

Nate Campbell - September 2015

"I'm working as Infrastructure Architect for Capgemini, and I'm interested in how to formally represent an IT Infrastructure. Until now, I've mostly worked with Excel and Visio, but I quickly realize those tools don't scale properly for the job, and lack formalisms when we are dealing with large enterprise-scale Infrastructures."

Alexandre Dumont - March 2016

[3]For simplicity sake, herein the word "infrastructure" will be shorthand for information technology infrastructure and the word "patterns" will stand for information technology infrastructure patterns.

[4]**Frans Wiggers** https://www.linkedin.com/pub/frans-wiggers/0/516/506

[5]**Nate Campbell** https://www.linkedin.com/pub/nate-campbell/59/b31/a74

[6]**Alexandre Dumont** https://www.linkedin.com/in/alexandredumont

1.2.2 Research Questions

To address the research problems of section 1.2.1, the following research questions were formulated:

- RQ1: How to model ITIs to capture more information?
- RQ2: How to express and reuse ITI best practices?
- RQ3: Which methodological approach can we use to create a new ITI based upon patterns?

The RQ1 is related with RP1, RQ2 with RP2 and RQ3 is related with RP3. Each of the above research questions is also related with the research hypotheses in section 5.4.

1.2.3 Main Contributions

This section presents a summary of the main results achieved and some of the benefits with the development of this dissertation:

1. **To address the problem of informality in ITI designs, we proposed a language including its abstract and concrete syntax.** To address the list of previously mentioned problems, related with informally designed ITIs, using *ad-hoc* methods, we created an abstract representation (ITI metamodel) and a concrete representation (ITI notation) of the same language. The ITI metamodel contains more than one hundred ITI concepts, each with a set of related attributes, to enable their unambiguous representation. We published these results at [Sil+12a; Sil+12b] and we provide a subset of the profile in section 3.4 and a complete ITI metamodel in appendix C.

2. **To express best practices in the design of ITIs, we proposed a set of ITI patterns.** Design agility is achieved in most engineering fields by using appropriate abstractions. Although often "*the devil is in the details*", raising the level of abstraction allows architects to find, share and apply standardized solutions to recurrent phenomena, by only retaining the information which is relevant for a particular purpose. Wrapping up those standardized design solutions, resulted in what has been coined as design patterns. We applied this concept to ITIs, by creating ITI patterns to support the design process. We published ITI patterns at [SA10a; SA10b; SA11] and included in section 4.3 some ITI patterns published in conferences such as EuroPLoP [SA10a] for exemplification purposes. We provided also a patterns catalog in section 4.3.3.

3. **To simplify the use of ITI patterns, we deployed the ITI profile and the ITI patterns in a modeling tool.** Despite the importance of ITI patterns, they may not be sufficient to enable people to learn and use them when designing ITIs. The previously mentioned contributions would not be easily deployed if not provided by a modeling tool, when building or changing infrastructures. From a pure patterns

perspective, there are some modeling tools, such as *Rational Rose eXtended Development Environment (XDE)* and *Together*, that are only focused in a specific class of patterns and, according to [Bos+05], their descriptions only consist of a few sentences and sketches, which is not enough to understand the meaning of patterns. As mentioned before, to incorporate the profile and a set of patterns, we decided to extend the Enterprise Architect (EA) tool.

Some of the benefits arising from these results and confirmed by a survey and controlled experiment include:

- **Decrease ITI design effort.** Since many ITI building blocks are already defined as ITI patterns, the ITI design process is simpler and contributes to obtain a more robust final solution, since those patterns are based upon proven solutions;
- **Knowledge reuse.** The use of a structured approach based on patterns allows the capture of knowledge used in the design of ITIs in a way that allows reuse. This knowledge is important to the business and to the IT professionals, since it decreases the risk of introducing bugs in the design process;
- **Simplify communication among stakeholders.** Due to the fact that most solutions are well-known, in most cases, doubts, questions and difficulties can fade away simply by providing the name of an ITI design pattern. The use of patterns facilitates communication, sharing of ideas, identification of recurrent problems, and promotes guided approaches to solve those problems [Tro+03], thereby improving design quality and efficiency [Yan10];
- **Facilitate integration.** ITIs normally include components from different vendors. Because the ITI patterns are vendor-agnostic, the integration is easier. This makes the process of building complex and heterogeneous solutions (those comprising multiple provider technologies) simpler;
- **Improve documentation.** Since ITI patterns provide a higher level of abstraction and are well documented, the effort to document an ITI based on those patterns will be smaller.

1.3 Dissertation Outline

Figure 1.3 shows how the six chapters of this dissertation are organized and interconnected along with the methods used to solve problems and the contributions obtained based on the methods used.

The claims described in section 1.2.3 of this dissertation were supported by a literature review and a systematic literature review, both available in chapter 2. In the systematic ITI literature review we used a methodological approach that allowed us to summarize the best available evidence on how IT infrastructure are represented.

The online survey allowed us to reach ITI experts from industry and academia around the world and confirmed the existence of the problems that motivated our research questions. The six chapters of this dissertation are grouped in five parts and can be briefly described as follows:

Part 1: Initiation

- **Chapter 1 - Introduction.** This chapter contains a short summary of problems addressed by this dissertation and briefly describes the contributions.

Part 2: Problems Confirmation

- **Chapter 2 - State of the art.** This chapter describes related work relevant to the understanding of the challenges around the representation of ITIs. It further confirms the poor state of the art found in the literature, by means of the results of an online survey.

Part 3: Implementation

- **Chapter 3 - Modeling IT Infrastructures**. This chapter describes the metamodels evaluated in terms of their extension capabilities to address the requirements of IT Infrastructure design. It contains also the overall structure of a proposed metamodel extension (profile) along with well-formedness rules enforcement, best practices and supporting tool.
- **Chapter 4 - IT Infrastructure Patterns**. This chapter reviews the concept of design patterns and proposes a catalog of IT Infrastructures patterns.

Part 4: Validation

- **Chapter 5 - Validation**. This chapter presents the results from the experimental studies and provides empirical evidence about the effectiveness of the approach used to validate the results obtained.

Part 5: Conclusions

- **Chapter 6 - Conclusions and Future Work** – In the final chapter we summarize the contributions resulting from the research work and we discuss future work.

1.4 Summary

The first chapter was intended to give a general overview of the research work in this dissertation and starts with a quick introduction of what IT Infrastructures are (section 1), and justifies the research problems (section 1.2.1).

Research questions and main contributions expected from the dissertation are described, respectively, in sections 1.2.2 and 1.2.3. Finally, it provides an outline with a brief explanation of what is included in each chapter (section 1.3).

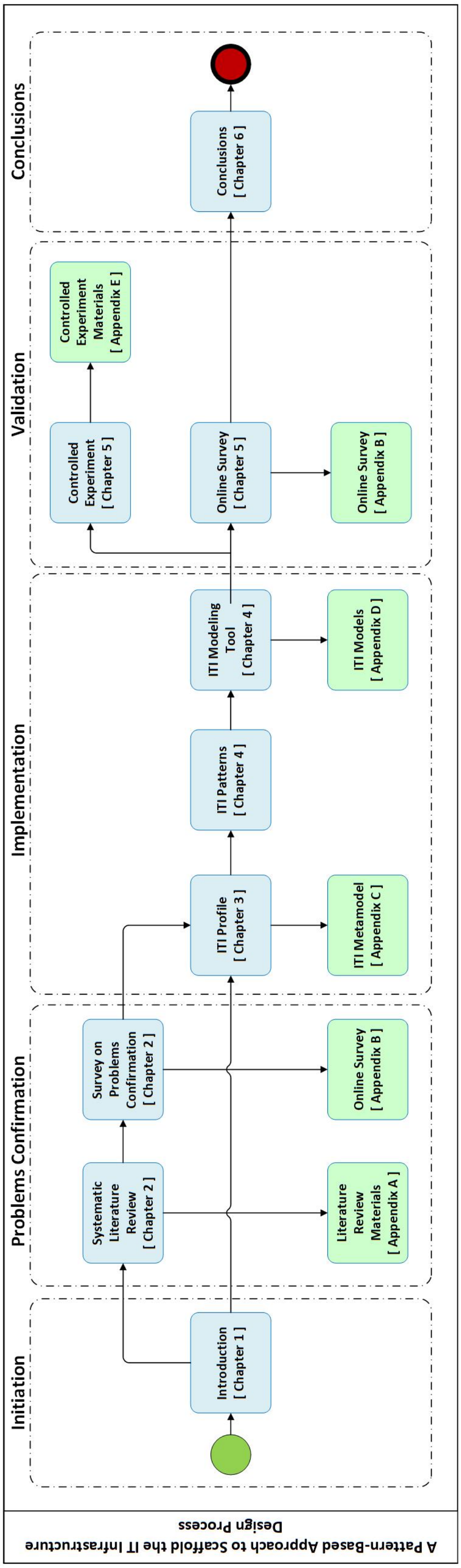


Figure 1.3: Dissertation outline

Chapter 2

State of the Art on ITI Design

Contents

2.1 Introduction

The objective of this chapter is to systematically review published research on design of ITIs methodologies/frameworks in organizations. We expect that this process presents a fair evaluation of the current state of the art in the area of IT infrastructures, by using a trustworthy, rigorous, and auditable methodology.

2.2 Systematic Literature Review

Among the several approaches that can be adopted to conduct literature reviews, we choose Systematic Literature Review (SLR) since we want an in-depth summary of the literature, based upon research questions and a set of criteria [Ken07]. The SLR conducted on the realm of ITIs, was first validated by peers and supported by several design guidelines to mitigate the risk of not having clear and narrow research questions [SN07].

We followed Kitchenham´s best practices [KC07] to find as many primary studies as possible, based upon the research questions presented in section 2.2.1.

Our approach to conduct the SLR was based in three main phases, which are depicted in more detail in figure 2.1. The main phases are: *(1)* **Plan SLR** that consists in creating an SLR protocol to guide the whole process with seven main interactive steps and an evaluation process at the end, to ensure the SLR quality *(2)* **Conduct SLR**, where the search with the instantiation of the SLR protocol is conducted and *(3)* **Document SLR** which is the final phase with the summary of findings.

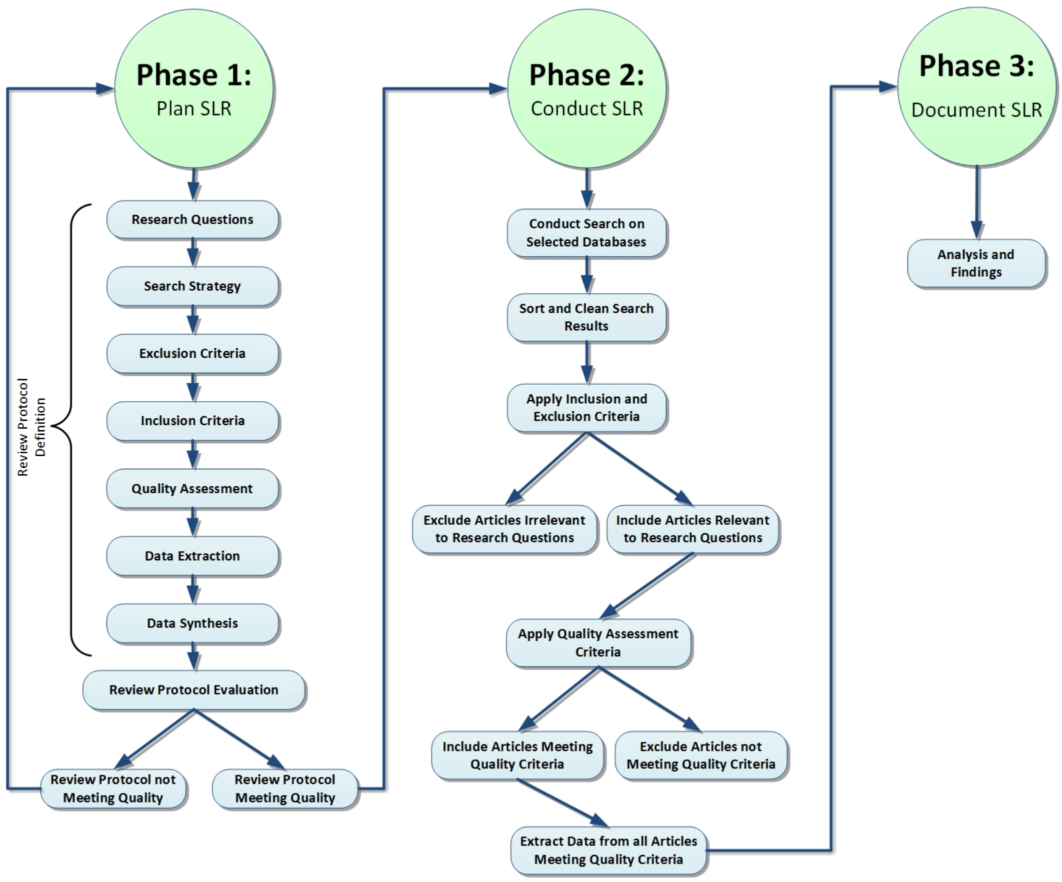


Figure 2.1: Phases of the systematic literature review

Next the section 2.2.3 describes the search strategy; section 2.2.4 presents the selection criteria; section 2.2.5 contains the quality assessment; section 2.2.6 contains the data extraction techniques and section 2.2.7 describes the data synthesis. We then document the results obtained (SLR protocol phase 3).

2.2.1 Research Questions

The goal of this SLR is to assess the current state of the art in the ITI design, by identifying studies that provide an approach based on the following four perspectives (P):

- **Pa**: ITI Design Approach
- **Pb**: ITI Complexity
- **Pc**: ITI Best Practices
- **Pd**: ITI Knowledge

The aforesaid perspectives originated the following research questions (RQ):

- **RQa**: How are ITIs currently designed?
- **RQb**: Is ITI complexity being evaluated?
- **RQc**: To what extent are best practices rules being applied to ITI design?
- **RQd**: Is ITI knowledge being shared, expressed or reused?

Notice that the above research questions were used to support and guide the SLR and were later reformulated in the research questions already mentioned in section 1.2.2.

2.2.2 Preliminary Search

Before conducting this SLR we conducted a preliminary search on the topic, to understand if a SLR regarding the design of ITIs had already been performed. In this preliminary analysis, we used an *Online Knowledge Library Portal* called *b-on*[1] because this portal has an integrated search that allows searching 25 databases in the engineering and technology area. The databases, their links and content types are available in table A.2.

2.2.2.1 Search Terms

To identify search terms to build this preliminary search string we used synonyms of "IT Infrastructure" and synonyms of "Systematic Review". Regarding "IT infrastructure" we based the synonyms mainly in our experience in the field as follows: {IT Infrastructure, Technology Infrastructure, Infrastructure Architecture, Information Technology Infrastructure, Technical Architecture, Enterprise Infrastructure, Technology Infrastructure Architecture, Enterprise Infrastructure Architecture}.

[1] The b-on is available at `http://www.b-on.pt` and makes unlimited and permanent access available, within the Portuguese research and higher education institutions, to full texts from over 16 750 scientific international publications from 16 publishers, through subscriptions negotiated on a national basis with these publishers.

To validate the synonyms of ITI, we counted the number of hits using *b-on* and two free search engines (*Microsoft Academic Search*[2] and *Google Scholar*[3]). The results are displayed in Table 2.1.

Table 2.1: ITI search terms

Term	Google Scholar	M.Academic	b-on
IT Infrastructure	11,600	359	227,982
Technology Infrastructure	5,530	136	86,273
Infrastructure Architecture	1,110	27	42,836
Information Technology Infrastructure	2,100	65	73,051
Technical Architecture	4,760	68	89,629
Enterprise Infrastructure	900	18	27,141
Technology Infrastructure Architecture	26	0	35,944
Enterprise Infrastructure Architecture	6	0	12,930

The same approach was used for SLR. For the SLR we used the terms available in [Bio+07], which are: {Systematic Review, Research Review, Research Synthesis, Integrative Review, Research Integration, Systematic Research Synthesis, Integrative Research Review}. The results are represented in Table 2.2.

Table 2.2: SLR search terms

Term	Google Scholar	Microsoft Academic	b-on
Systematic Review	7,100	209	401,563
Research Review	4,890	40	672,066
Research Synthesis	714	8	277,383
Integrative Review	442	13	48,932
Research Integration	710	8	263,134
Systematic Research Synthesis	10	0	57,729
Integrative Research Review	26	1	34,418

2.2.2.2 Preliminary Search String and results

All the search terms returned results, so the resulting search string combining the ITI synonyms with SLR synonyms is then the following:

> (("IT Infrastructure" **OR** "Technology Infrastructure" **OR** "Infrastructure Architecture" **OR** "Information Technology Infrastructure" **OR** "Technical Architecture" **OR** "Enterprise Infrastructure")
>
> **AND**

[2] `http://academic.research.microsoft.com`
[3] `http://scholar.google.com`

("Systematic Review" **OR** "Research Review" OR "Research Synthesis" **OR** "Integrative Review" **OR** "Research Integration" **OR** "Systematic Research Synthesis" **OR** "Integrative Research Review"))

The combined search, was used in the same search engines (*Microsoft Academic Search, Google Scholar and b-on*) to identify if an SLR was already performed in the ITI topic. We did not found any such review on the aforementioned topic, so we decided to move forward with the SLR with the goal of identifying as many primary studies as possible, according to the research questions.

2.2.3 Search Strategy

The main benefit of defining a search strategy is to increase the rigor of the search process, while making the work transparent and replicable. These criteria helped us to find as many primary publications relating to the research questions as possible, using an unbiased search strategy.

The defined search strategy includes the use of automated and manual search techniques to decrease the chance of missing pertinent information. The automated search was implemented in multiple online scientific databases for Engineering and Computer Science. The manual search method includes physically examining journals (hand searching), reference list checks [Woh14], and researching key authors' work.

2.2.3.1 Digital Libraries

The initial plan, was to use only the advanced search provided by the *b-on portal*. However, one of the challenges we had, was related with the size of the search query which was limited to only two fields, each limited to a maximum of 100 characters and separated by an operator (AND/OR/WITHOUT). This limitation forced us to use the engine of each digital library. The following digital libraries that provide Computer Science and Engineering related contents were used:

- Association for Computing Machinery (ACM) Digital Library
- Institute of Electrical and Electronics Engineers (IEEE) Xplore
- Science Direct / Scopus (Elsevier)
- SpringerLink (Springer / Kluwer)
- Wiley Online Library

The reasons to choose these databases were based on the preliminary search and analysis of results. Other databases such as Digital Bibliography and Library Project (DBLP)[4], was not included due to limitations on the search engines (e.g. capacity to

[4]http://dblp.uni-trier.de/db/

search only in titles) and because the publications retrieved were duplicated with other databases.

To minimize the risk of not including relevant publications and to retrieve the number of citations of each publication we used the *"Google Scholar"* search engine.

2.2.3.2 Search Period

We decided to not restrict the search period of our SLR. The decision was based on the results obtained during our preliminary searches (section 2.2.2), where we identified the most common terms and in the relatively low number of results returned using the *Google Scholar* search engine. Figure 2.2 shows the evolution of the number of ITI papers published over the years. We used the *Google Scholar* advanced search to restrict the search to the title of the article, search for the exact phrase "IT Infrastructure" (the most common term) and restrict to the publications in a specific year. We excluded from our search results patents.

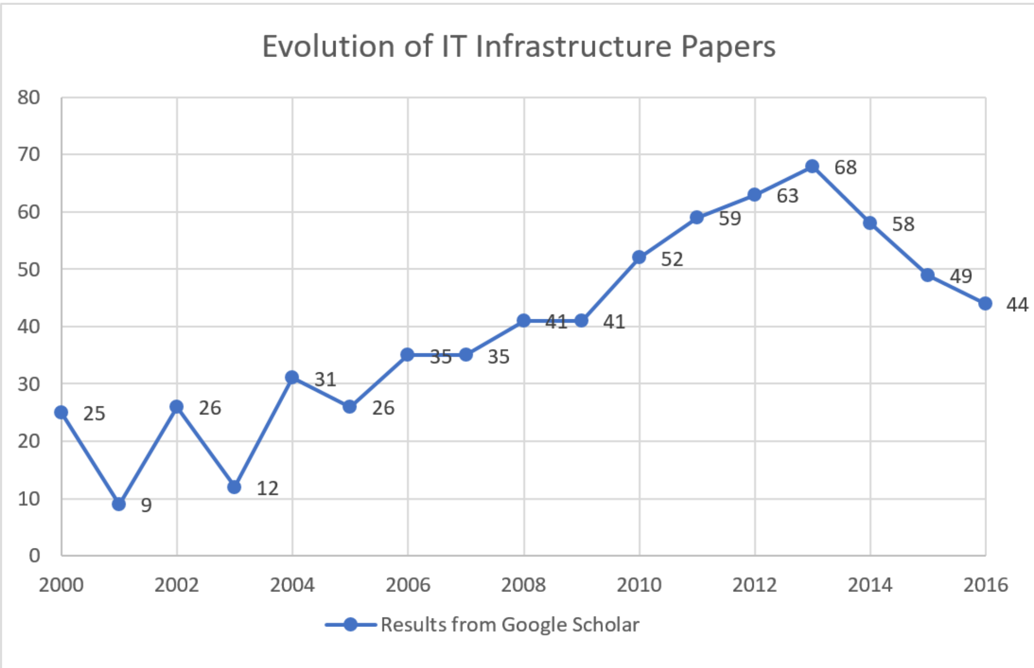


Figure 2.2: Published ITI papers over the years

2.2.4 Selection Criteria

This section presents the search string (with the unfolding of expressions, concatenated) and inclusion and exclusion criteria to retrieve a preliminary set of publications to be evaluated in more detail.

2.2.4.1 General Search Terms

To build a query we needed to know what terms are relevant for each research question and include them in our search string, that will be presented in the next section. To discover relevant terms, we used our experience in the field, combined with online resources such as the Educational Resources Information Center (ERIC) Thesaurus of Descriptors[5].

According to [MB64] our list of general search terms was amended to include only the relevant terms. We considered relevant to our research questions, publications with the term ITI in the publication name and at least one of the following keywords: {*Architecture, Documentation, Perspective, Express, Design, Reference, View, Analysis and Analyses, Assessment, Description, Evaluations, Vocabulary, Structure, Representation, Drawings, Picture, Implementation, Language, Technical Diagram, Software Infrastructure, Hardware Infrastructure, Networking Infrastructure, Inventory, Best Practice, Guidelines, Blueprint, Knowledge*}.

2.2.4.2 Search String

The search string creation was an important step in the SLR since it was the query used in the search engines and a single word may significantly affect the results.

To minimize the risk of not including relevant studies, we decided to retrieve first all publications related with ITIs and store them in a database and then use queries and filters to search within results for relevant terms related with the research questions. This decision was also important to overcome the issues related with the limited number of characters in the search field of most search engines.

As performed earlier in section 2.2.2, we used the same approach to create a query string with synonyms of ITIs combined by using the boolean "OR" operator. Because we want to find exact expressions, we used quotes (e.g. we wanted to include results with the word "IT Infrastructure" but not isolated occurrences of "IT" or "Infrastructure").

The first query string used to retrieve all ITI publications was as follows:

> "IT Infrastructure" **OR** "Technology Infrastructure" **OR** "Technical Infrastructure" **OR** "Infrastructure Architecture" **OR** "Service Oriented Infrastructure" **OR** "Cloud Infrastructure"

The above query string was used in each of the digital libraries available in section 2.2.3.1. As mentioned before, the results were then stored in a global database, each with the indication of the digital library name where the publication was retrieved. We then searched for terms related with our research questions such as model, design, patterns among others. Since each search engine has a specific syntax, we had to adapt the query string to each of the five databases selected.

The queries were as follows:

[5] http://www.eric.ed.gov

- **ACM**

  acmdlTitle:("IT Infrastructure" "Technology Infrastructure" "Technical Infrastructure" "Infrastructure Architecture" "Service Oriented Infrastructure" "Cloud Infrastructure")

- **IEEE**

  ((((((("Document Title":"IT Infrastructure") ***OR*** *"Document Title":"Technology Infrastructure")* ***OR*** *"Document Title":"Technical Infrastructure")* ***OR*** *"Document Title":"infrastructure architecture")* ***OR*** *"Document Title":"Cloud Infrastructure")*

- **ScienceDirect**

  TITLE("IT Infrastructure") ***OR*** *TITLE("Technology Infrastructure")* ***OR*** *TITLE("Technical Infrastructure")* ***OR*** *TITLE("infrastructure architecture")* ***OR*** *TITLE("Service Oriented Infrastructure")* ***OR*** *TITLE("Cloud Infrastructure")*

- **SpringerLink**

  Title:(("IT Infrastructure" ***OR*** *"Technology Infrastructure"* ***OR*** *"Technical Infrastructure"* ***OR*** *"infrastructure architecture"* ***OR*** *"Service Oriented Infrastructure"* ***OR*** *"Cloud Infrastructure"))*

- **Wiley**

  "IT Infrastructure" in Publication Titles ***OR*** *"Technology Infrastructure" in Publication Titles* ***OR*** *"Technical Infrastructure" in Publication Titles* ***OR*** *"infrastructure architecture" in Publication Titles* ***OR*** *"Service Oriented Infrastructure" in Publication Titles* ***OR*** *"Cloud Infrastructure" in Publication Titles*

2.2.4.3 Inclusion criteria

We only considered primary studies and the criteria for inclusion of publications was the presence of terms relevant with the research questions. Relevant terms for our research must be present in *titles* of the publications and were already defined in section 2.2.4.1.

2.2.4.4 Exclusion criteria

The following exclusion criteria were used to filter the retrieved publications:

- **Non-English and Invalid**. The documents that do not had at least the abstract in English were excluded. We based this decision not only in the fact that some publications based upon East Asian languages such as Chinese, Japanese, and Korean may be difficult to access and translate but also in the assumption that if a publication is relevant, someone would have translated to English, since this is the most widely used language in the scientific community.

- **Out of Scope**. Publications not related with Information Technology Infrastructures were excluded. The rules were only to include publications if doubts arise.

- **Relevance**. Publications that are not relevant to the previously identified research questions were excluded. If it was not clear whether publication was relevant, the rules was to include them for a detailed quality assessment.
- **Duplicated**. Publications duplicated among databases were discarded.
- **Patents and Citations**. Patents and Citations were discarded.
- **Our Own Publications**. Our publications were discarded.
- **Relevant Keywords**. Publications without at least one of the keywords in the title relevant to this research will be discarded.

2.2.5 Quality Assessment

The assessment of contents quality for primary publications is generally considered important for several reasons, such as to provide more details regarding inclusion and exclusion criteria, to compare the importance of individual publications in the result synthesis or to guide the interpretation of findings [HG11]. Since there was not an agreed definition of *quality* in this research, *quality* was measured in terms of relevance [Ham98], while minimizing bias[6] and maximizing internal and external validity [CRD09; HG11].

2.2.5.1 Quality Checklist

In order to evaluate *quality*, we separate the results of each digital library. Each publication contains a column with the name of the library where the publication was retrieved and the *Paper Title*, *Abstract*. There is also a field for *Classification* that can have the following values:

- **Out**. The publication is clearly out of scope and was not further analyzed.
- **Maybe**. There are some doubts regarding the scope. Publication was downloaded for detailed analysis before deciding upon inclusion in the scoped literature.
- **In**. The publication was clearly within scope and was included in the scoped literature.

The publication classified as "**In**" was then analyzed in detail to identify the primary studies.

2.2.5.2 Evaluation Checklist

The strategy to evaluate the quality of these papers classified as **In** or **Maybe**, is based on a checklist with eleven criteria. The criteria were based on good practices for conducting empirical research proposed by [KP02c] and in the Critical Appraisal Skills Programme (CASP)[7]. The latter was used in different types of publications by Dybá [DD08; Dyb+07;

[6] A tendency to produce results that depart systematically from the "true".
[7] `http://www.phru.nhs.uk/casp/casp.htm`

Dyb+08] and since we envisaged that our review will also include different types of publications, we decided to adopt the same criteria. The criteria developed to assess quality covered three main quality issues considered necessary when appraising primary papers:

- **Rigor**. Has a thorough and appropriate approach been applied?
- **Credibility**. Are the findings well-presented and interpretations plausible and/or credible?
- **Relevance**. Are the findings related with IT infrastructure relevant for the research community?

2.2.6 Data Extraction

To provide standard information regarding the papers under analysis, we created a data collection form available in appendix (table A.3). This data collection form helped to identify how each of the publications addressed our research questions.

2.2.7 Data Synthesis

To synthesize findings from data extracted we took in consideration the seven-step process for synthesizing research as described in [Bri+02] since it is the most widely cited method. The seven phases are:

1. **Getting started**. The first phase involves identifying an intellectual interest that qualitative research might inform. In other words, this phase is related with finding something that is worthy of the synthesis effort.

2. **Deciding what is relevant to the initial interest**. This phase implies knowing who the audience for the synthesis is, to decide what is relevant.

3. **Reading the publications**. This phase requires careful reading of the chosen papers in order to identify the main concepts.

4. **Determining how the publications are related**. This phase consists in the identification of the same concepts in different publications.

5. **Translating the publications into one another**. In this phase, each paper will be part of a grid to simplify comparison. The metaphors and/or concepts and their interactions are compared with the metaphors and/or concepts and their interactions in other publications.

6. **Synthesizing translations**. This phase represents making a whole into something more than the parts alone imply. The various translations can be compared with one another to determine if there are types of translation or if some metaphors/concepts are able to encompass other publications.

7. **Expressing the synthesis**. The final phase consists in communicating the results effectively, what need to consider the intended audiences to use concepts and a language they can understand.

2.2.8 Results

Our combined search across all the selected research databases returned **706** publications that contain a synonym of ITI in the publication title. The number of publications from *Google Scholar* was **2447**. Notice that there are duplicated publications in these numbers, and as was mentioned before, the queries were not restricted to a specific period of time or to a specific domain in science. The number of results returned from each research database and the details regarding publication types are available in table 2.3. Notice that, since each digital library uses different taxonomies to classify publication types, we decided to group them into three categories:

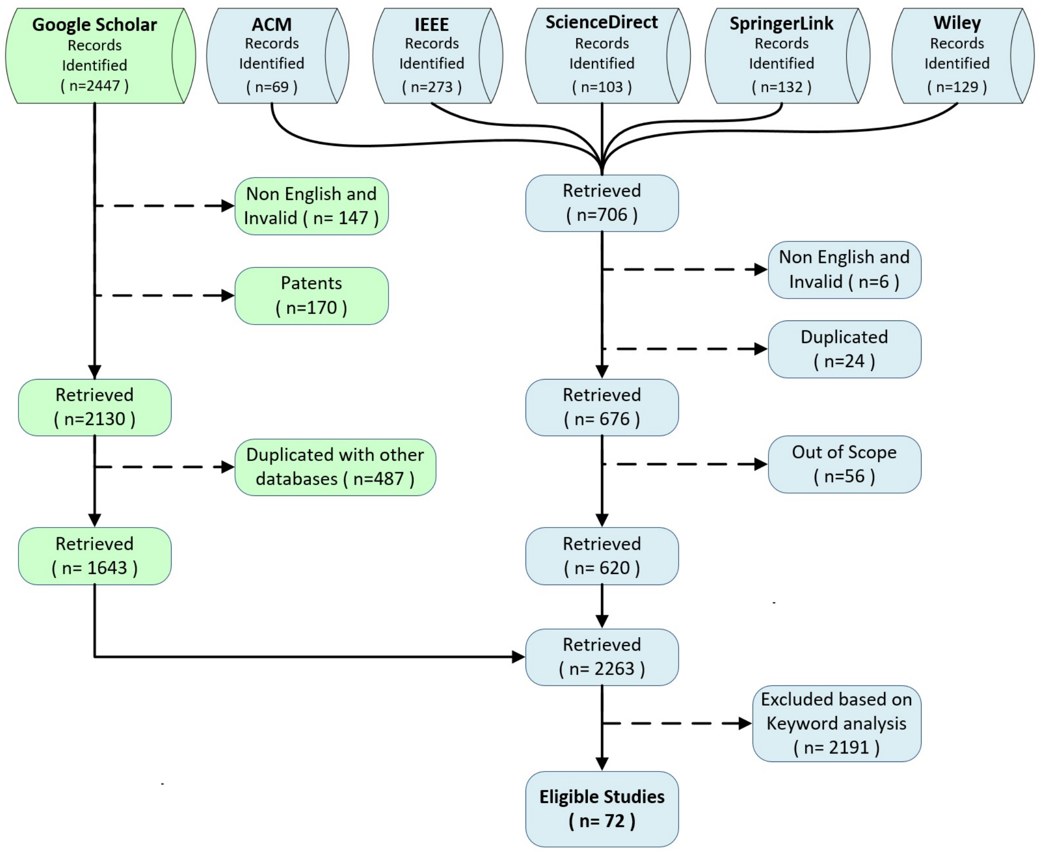


Figure 2.3: SLR results

- **Articles.** Includes journal articles, conference proceedings, magazines and generic publications.
- **Books.** Includes chapters, books chapter, ebooks and inbook publications.
- **Other.** Includes reference works, lab protocol, unpublished work and courses.

Table 2.3: Results returned from research databases

Publication	ACM	IEEE	ScienceDirect	SpringerLink	Wiley	Total
Articles	69	271	87	35	64	**526**
Books	0	2	15	97	65	**179**
Other	0	0	1	0	0	**1**
Total	69	273	103	132	129	**706**

The ACM allows the selection of *ACM Guide to Computing Literature*[8] and *ACM Full-Text Collection*[9]. Since our focus was computing, we decided to use the *ACM Guide to Computing Literature*.

The number of results retrieved from google scholar were much higher since the search was not restricted to specific databases. Table 2.4 present the databases with the highest number of records.

Table 2.4: Results returned from google scholar

Publisher	#	Publisher	#
IEEE	339	koreascience.or.kr	15
SpringerLink	161	adsabs.harvard.edu	14
ScienceDirect	82	agris.fao.org	14
search.proquest.com	73	arxiv.org	13
ACM	67	emeraldinsight.com	13
Citeseer	65	DTIC Document	12
books.google.com	63	pdfs.semanticscholar.org	11
researchgate.net	58	cds.cern.ch	10
academia.edu	47	gradworks.umi.com	10
Taylor & Francis	38	search.ebscohost.com	10
aisel.aisnet.org	36	diva-portal.org	8
en.cnki.com.cn	30	igi-global.com	8
ir.ptir.org	29	papers.ssrn.com	8
inderscienceonline.com	24	yadda.icm.edu.pl	8
Wiley	23	go.galegroup.com	7
ERIC	19	ideas.repec.org	7
infona.pl	19	ntrs.nasa.gov	7

[8]ACM Guide to Computing Literature is the most comprehensive bibliographic database focused exclusively on the field of computing. This database includes all of the content from The ACM Full-Text Collection along with citations, and links where possible, to all other publishers in computing

[9]The ACM Full-Text Collection will return results from all articles ever sponsored or published by ACM along with a set of vetted, hosted content from selected publishers. All of this content is made available directly from The ACM Digital Library.

2.2.8.1 Sort and Clean Results

With all the results in a database, we used several techniques to identify duplicated publications or publications in non-English languages. The process to identify duplicates, detected some duplicated articles in multiple databases stored with different names, different languages (e.g. "IT Infrastructure: interet pour la...") and with other minor changes in the title such as the existence of commas, semicolon or quotes.

To simplify the search within results, we normalized some names in publication titles such as changing *Information Technology Infrastructure* and *IT-Infrastructure* to *IT Infrastructure* or *IT Infrastructure Library* to *ITIL*.

We used mainly Microsoft Excel features such as sorting, filtering and formulas to detect duplicates and dictionaries and grammar checking to identify publications in other languages or clearly out of scope, as defined in the exclusion criteria. Regarding research databases these techniques allowed us to remove 86 publications and the final number of publications was **620**.

Regarding *Google Scholar* results, a closer look reveals the existence of a high number of records related with patents. According to the exclusion criteria, both were removed, and the number of publications decreased to 2130. According to our strategy, we decided to combine the results of the *Google Scholar* and research databases. In cases of duplicate records originating from *Google Scholar* and the research databases, we kept the records from the latter, as they are, in general, more detailed and accurate (indeed, *Google Scholar* also indexes these databases, so we prefer to use the data from its source).

According to our expectations most of the results from research databases were also in *Google Scholar* and after applying the same techniques to identify duplicated, invalid or publications in other languages we were able to remove 487 publications. The unique number of publications from *Google Scholar* was 1643 that, combined with the results from the research databases, gave us a total of 2263 ITI publications retrieved.

2.2.8.2 Search Challenges

Some of the main challenges we faced with this process were:

- There were **227** publication that were returned with no publication year, which is usually caused either because there is no year on the publication or because the engine does not recognize the year as such.
- One of the limitations with the ACM was the inexistent options to export all the results with abstracts. We had to export each article separately.
- The SpringerLink search engine did not have enough characters to include our complete search query. To overcome this limitation, we had to perform individual queries for each of the "OR" terms.
- With the Wiley Online Library, we had to construct two queries to search in "Publication Titles" and in "Article Titles".

2.2.8.3 ITI Keyword Analysis

According to our strategy and defined in our inclusion criteria, we will now search within these results to find publications related with the terms identified in section 2.2.4.1 that match the research questions in our database with 2263 ITI publications. Publications without at least one of our important keywords in the title will be excluded.

The results are presented in table 2.5. To be able to identify the most common words in the list of selected publications, we used text analyzers. Figure 2.4 shows the result and warn us to include in our search within results the shorter term (ex. Model instead of Modeling, Pattern instead of Patterns, among others) to be able to get all publications around a specific topic.

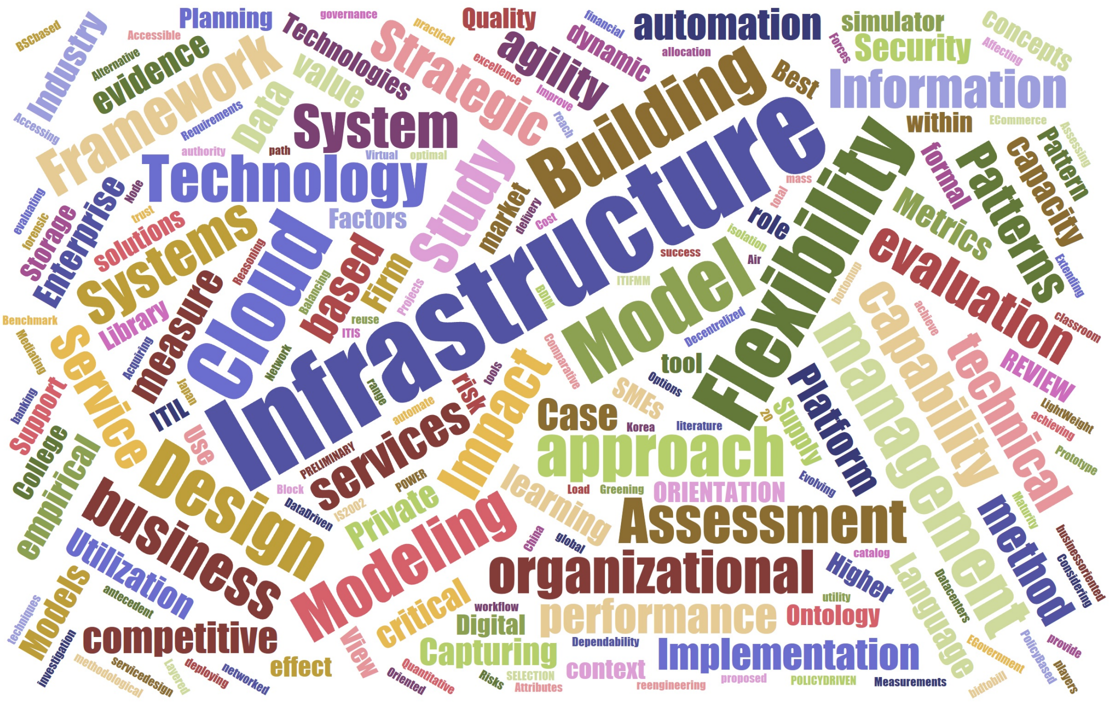


Figure 2.4: Most common words retrieved from selected publications

Based on the important keywords defined earlier in section 2.2.4.1, we identified the target publications, and we classified them as **In**, **Out** or **Maybe** according to the quality checklist defined in section 2.2.5. We were able to retrieve 72 eligible publications (available in table A.4). Full text reading allowed us to remove 50 publications not related with any of our research questions, therefore remaining only 22. The publications assessed against our research questions are identified in table 2.6.

Table 2.5: Publications returned by keyword

#	Keyword	Publications	#	Keyword	Publications
1	Model	154	12	Language	21
2	Architecture	122	13	Structure	14
3	Implement	77	14	Reference	13
4	Design	72	15	Pattern	8
5	Analysis	70	16	Ontology	8
6	Evaluation	40	17	Comparison	5
7	Flexibility	39	18	Taxonomy	5
8	knowledge	36	19	View	4
9	Perspective	32	20	Best Practices	4
10	Assessment	25	21	Blueprint	4
11	Empirical	24	22	Represent	4

Table 2.6: Target publications

Paper	Name
[MB89]	Building IT Infrastructure for the 1990s
[HV93]	Leveraging Information Technology for Transforming Organizations
[WB98]	Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology
[Bro+99]	Strategic context and patterns of IT infrastructure capability
[Wei+02]	Building IT Infrastructure for Strategic Agility
[Ado+06]	Understanding Patterns of Technology Evolution
[Bal+07]	CyberGuarder: A virtualization security assurance architecture for green cloud computing
[Sol+07]	Towards a Real-Time Reference Architecture for Autonomic Systems
[GR08]	Information Systems Implementation: The Big Picture
[Roh08]	An Integrated View on Business and IT-architecture
[Sau+09]	Business-driven design of infrastructures for IT services
[EB09]	An empirical examination of the relationship between IT infrastructure, customer focus, and business advantages
[Bha+10]	Building and leveraging information in dynamic environments
[Ghi+11]	Deploying an Agent Platform to Automate the IT Infrastructure Auditing Process
[Anc+11]	Infrastructure Pattern Discovery in Configuration Management DBs via Large Sparse Graph Mining
[Cal+11]	CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms

Continued on next page

Table 2.6 – continued from previous page

Paper	Name
[Sul+12]	A System of Architectural Patterns for Scalable, Consistent and Highly Available Multi-Tier Service-Oriented Infrastructures
[Krü12]	Conceptual Methods to Design Sustainable IT Infrastructures Standardization, Consolidation, and Virtualization
[Krü+12]	Feature modeling: An extended perspective to design IT infrastructures
[Pal+12]	Utilization analysis of servers in a data centre
[Hol+14]	Automatic data collection for enterprise architecture models
[Fre+15]	Weaving in Patterns into IT Infrastructure Models - Industry Case and Exemplary Approaches

2.2.9 Analysis and Findings

This section presents the findings after the analysis of selected studies. To provide the synthesis of the systematic review results, we decided to create a table where the rows represent the categories for evaluation identified in taxonomy and columns are the selected evaluations. The intersection between rows and columns are depicted as a symbol:

- Strong: ●
- Moderate: ⊙
- Weak: ○
- None/Basic: —

We classified each of the selected studies in four different dimensions that we considered important to this research work. The dimensions are:

- **ITI Design Approach.** A publication classified as strong, includes a formal language that is able to express any construct of a complex ITI design, their attributes and dependencies. None/Basic classifications should be used when the paper provides no mechanism to model or represent the ITI.
- **ITI Complexity.** A publication classified as strong, provides an approach that can be used to evaluate several complexity dimensions of an ITI, such as their size in terms of the number of nodes and connections among the former. None/Basic should be used when the publication does not evaluate the impact of complexity on ITI.
- **ITI Best Practices.** Strong means the publication includes the capability to express best practices using for instance well-formedness rules. None can be used when there is no information related with best practices.
- **ITI Knowledge.** The strong classification in knowledge reuse is for publications that provide approaches where the ITI knowledge can be shared, expressed or reused.

The None classification is applicable to publications where the impact of knowledge is not mentioned.

Table 2.7 show the assessment of each paper against our criteria. The last line includes our assessment of the contributions of this research work against the same criteria.

Table 2.7: Analysis of the selected studies

Paper	Design Approach	Complexity	Best Practices	Knowledge
[MB89]	—	—	—	○
[HV93]	—	—	—	○
[WB98]	—	—	—	○
[Bro+99]	—	—	—	○
[Wei+02]	—	—	—	○
[Ado+06]	—	○	—	—
[Bal+07]	○	—	—	—
[Sol+07]	—	—	○	—
[GR08]	—	—	—	○
[Roh08]	⊙	○	—	○
[Sau+09]	○	○	⊙	○
[EB09]	—	—	—	○
[Bha+10]	—	—	—	○
[Ghi+11]	—	○	—	⊙
[Anc+11]	○	—	○	⊙
[Cal+11]	○	—	—	—
[Sul+12]	—	○	—	—
[Krü12]	⊙	—	—	—
[Krü+12]	—	—	—	○
[Pal+12]	—	—	⊙	○
[Hol+14]	⊙	○	—	⊙
[Fre+15]	⊙	—	—	—
OUR	●	⊙	●	⊙

Strong: ● Moderate: ⊙ Weak: ○ None/Basic: —

2.2.9.1 ITI Design Approach

Different approaches are being used to design or represent ITIs. [Krü12] advocates the use of feature modeling for designing ITIs to give an additional perspective on systems architecture and implement functions and properties of IT infrastructure. The approach intends to describe the architecture of IT infrastructures from the functional view with the assistance of feature models. Feature modeling is commonly used in software development and is a compact representation in terms of "features". Another publication from [Krü+12] points out that it is advisable for designing ITIs to evaluate methods regarding their benefit for sustainability and concludes that standardization, consolidation and virtualization are important to reduce the number of hardware elements and therefore lower the total cost of ownership.

Another approach proposed in [Fre+15] to the design of ITIs (also used in software to describe software-intensive systems) was the Fundamental Modeling Concepts (FMC). [Fre+15] compared FMC modeling notation and ArchiMate/OIAm (Open Infrastructure Architecture method). Some authors such as [Hol+14] proposed an automatic network scanner to collect data and generate EA models using the ArchiMate metamodel. The ITI design approach of [Roh08] was based on architecture views. They used a component view to describe the elements of architecture and their relationships, a communication view that shows how the elements interact with one another and the distribution view that describes how the elements are distributed in terms of location or organizational assignment. One of the key elements is the use of a *blueprint* that takes the form of a table with two dimensions showing interdependencies among the ITI building blocks. Some authors in [Anc+11; Cal+11] used a model and an algorithm called *CMDB-Miner* for mining infrastructure patterns from Configuration Management DataBase (CMDB) and the representation is based on a graph. A simulation toolkit called "CloudSim" that enables simulation of basic data center and cloud infrastructure components such as virtual machines, hosts and resource provisioning policies is proposed in [Bal+07]. A model allowing IT infrastructure and operational processes, to be represented and related from a control and risk perspective is proposed. This provides a consistent way of capturing and relating the risk views for the various stakeholders within the organization.

In summary, the ITI design requires a deep understanding of several infrastructure components and models are useful tools to express this knowledge. However, based on the analyzed papers, we can conclude that ITI models are used surgically and there are no standard processes to design or represent ITIs. Some authors used generic metamodels constructed with purposes other than designing ITIs, that only allow to capture a limited subset of information. Each author seems to use its own approach to design ITIs and most of the representations are high level and informal. This is confirmed by [Fre+15] that concluded that none of the model notations evaluated allowed the creation of a conceptual or physical representation of the ITI. It was also mentioned that the concept of patterns is not supported in any of the modeling languages and important characteristics of ITI such as automation, redundancy, deployment is not possible to represent without extending modeling notations. [Sul+12] discusses several aspects of public cloud infrastructures and concludes that one of open research challenges consider how to model infrastructures, which is what this dissertation addresses.

2.2.9.2 ITI Complexity

Complexity, and the approaches to evaluate it, was a topic mentioned in some publications. The most common methods used to assess complexity is the use of interviews and surveys, such as the ones described in [WB98] and [Bro+99]. The latter outlined the growing importance of ITIs and developed the concept of ITI capability through the identification of ITI services and measurement of business connectivity, as a way to measure complexity. A similar approach is described in [Wei+02], where the concept of ITI capability cluster is used to evaluate the number and type of capabilities of an ITI.

The approach proposed in [Roh08] was based on the creation of three basic views that can reveal the level of complexity. The author proposed a consolidation of architecture description and claims that complexity can be reduced based on three basic views on architecture. [Ghi+11] used an approach to evaluate the ITI based on the use of the Simple Network Management Protocol (SNMP), Common Information Model (CIM) and Web Services Distributed Management (WSDM). The approach is based on a generic model that can support IT audits and a layered architecture based on the information collected by agents.

A state diagram-based approach, proposed in [Ado+06] that provides insights for visualizing and analyzing complex relationships between multiple technologies in a technology ecosystem. It is claimed that this approach provides managers an additional tool for supporting decision-making tasks involving new technologies and innovation and to think more systematically about the complex dynamics at play in the evolution of ITI.

In summary, complexity can take many forms and dimensions and identifying or evaluating complexity in an organization or an ITI, is a challenging task. For instance, evaluating complexity based on the size of an ITI, can range from a couple of servers to thousands of ITI components. The topic complexity was always addressed indirectly, based, for instance, in the number of ITI capabilities, flexibility among others, but a rigorous process to evaluate ITI complexity was not found. One of the main motivations to evaluate complexity, was to give an idea of the level of complexity to decision makers, to help them in the definition of strategies and future of ITI.

2.2.9.3 ITI Best Practices

There are several publications on the topic of architecture best practices. In [Sau+09] best practices for designing data center infrastructure are proposed. It is mentioned that the main departure from existing methodologies is that it evaluates and compares alternative designs using business metrics, rather than purely technical metrics. Specifically, the proposed methodology evaluates the business impact (financial loss) imposed by an imperfect infrastructure. [Krü12] also tried to adapt a widely used method from systems engineering to the design of ITIs. The authors of [Sol+07] presented a reference architecture created for autonomic computing using best practices from software engineering, and from real-time systems. The architecture allows for a more flexible implementation of an autonomic computing system that can function in a heterogeneous environment. A framework with best practices, was also proposed by [WB98] to evaluate ITI and enable business managers to make the important financial decisions regarding the design of ITIs. This was based in research with more than 100 multinationals using surveys and interviews. [EB09] reports that ITI flexibility was positively related to information generation and dissemination and that information generation was related with organizational responsiveness. The research proved that ITI flexibility can help to respond to market opportunities creating a competitive advantage. They empirically tested the model using data collected from senior executives of 105 manufacturing and service firms. In [Pal+12], statistical techniques are used to automatically identify and evaluate servers that satisfy conditions with respect to their utilization characteristics.

In summary, we evaluated several publications to identify where they provide mechanisms to avoid inconsistencies or to express best practices in the context of ITI design. Best practices can take many forms, such as well-formedness rules, that can be used to enforce consistency in processes, such as in the design of ITIs, however none focuses explicitly on the ITI design.

2.2.9.4 ITI Knowledge

ITI depends on the knowledge of several different professionals to be successful. Different approaches may be used to share, express or reuse ITI knowledge. McKay and Brockway in 1989 [MB89] were the first to describe ITI as the enabling foundation of shared IT capabilities and to describe the importance of ITI to the entire business. They focused on the need for a common language for sharing knowledge among the ITI stakeholders. The driver for a common language is however related with investments and budgets and not with the design of ITIs. [Bro+99] described the features of ITI capability, highlighting the role of human expertise in the development and management of infrastructure services. The approach of [Wei+02] was based on a framework and in the identification of ITI capabilities to increase the ITI knowledge and ensure investments are aligned with organization's strategic goals and business initiatives. The goal of the framework is to support decisions about the future of ITIs. Detailed interviews and extensive questionnaires in 89 enterprises were used to validate the framework results. The study [HV93] presents a model for strategic alignment of four domains: *(i)* business strategy, *(ii)* information technology strategy, *(iii)* organization infrastructure, and *(iv)* information technology infrastructure and processes to provide better knowledge and alignment across these domains. [Roh08] focused on fundamental design techniques for enterprise architecture, which includes infrastructure architecture. Three views are proposed: Component, Communication and Distribution to describe the infrastructure building blocks at a higher abstraction level.

In summary, most publications mention the importance of having a common language and view of the ITI, but little or no information is provided regarding how the knowledge on ITI design can be expressed in a way that can easily be shared and reused.

2.3 Online Survey

As stated before, an online survey was applied during this dissertation. The online survey had two main goals, which were to check the state of the art in the design of ITI and to evaluate whether our proposals address the previously identified research problems. This section intends to present a summary of the survey results related with the state of the art in the design of ITI. The complete survey is available in appendix B.1 and their design methodology and the results obtained in each question are available in section 5.5.

As will be detailed in section 5.5.4, we were able to obtain 117 valid responses from people with several years of experience working with ITIs. The level of experience was also assessed by an optional question and the results are presented in the figure 2.5. As can be seen only 3 had less than 5 years' experience.

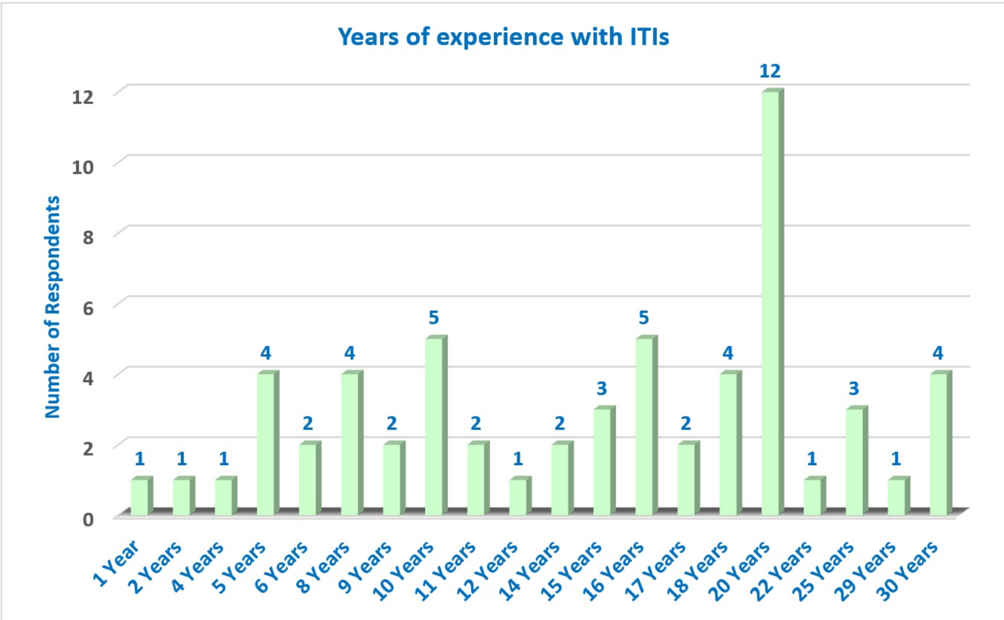


Figure 2.5: Years working with ITIs

Since the survey was available online, we were able to get responses from professionals all over the world. In terms of the state of the art in the design of ITI, we grouped results into the same four perspectives that were defined for the literature review:

- **ITI Design Approach**
- **ITI Complexity**
- **ITI Best Practices**
- **ITI Knowledge**

2.3.1 ITI Design Approach

The survey results, confirmed that existing approaches to design ITIs, do not follow a standard process and that there is not a standard methodological support to design ITIs.

As can be seen in table 2.8, it was without surprises that we confirmed with the survey, the use of 34 different tools to design ITIs.

Table 2.8: Software tools used to represent ITIs

#	Tool	#	Tool
1	*UML editors*	18	*BPM*
2	*UML based tools*	19	*Cisco modeling*
3	*Microsoft Visio*	20	*Visual Paradigm*
4	*Microsoft Excel*	21	*Balsamiq Mockups*
5	*Microsoft PowerPoint*	22	*Macromedia Fireworks*
6	*Microsoft Word*	23	*UML Model Designers*
7	*Draw.io*	24	*Rational Rose Tools*
8	*MAPS*	25	*Cloudstack*
9	*yEd*	26	*Wiki's*
10	*Omnigraffle*	27	*Custom-made solution*
11	*Inkscape*	28	*CMDB*
12	*Dia*	29	*Microsoft Paint*
13	*StarUML*	30	*Onenote*
14	*Sparx Enterprise Architect*	31	*Openoffice Draw Thinkcomposer*
15	*Visual Studio*	32	*Microsoft Visio with stencils*
16	*Gliffy*	33	*Vector-based OSS*
17	*Microsoft Visio with UML*	34	*Freeware graphics packages*

Almost all the tools being used are generic and are not related with the ITI domain and have no built-in capabilities to support a more formal ITI design process (e.g. using an ITI notation). There was a question in the survey related with the use of notations and according to our expectations (figure 2.7), most of the respondents confirmed that no specific notation is used to represent ITIs. Some respondents use the notation of UML deployment diagrams, one claims to use *ArchiMate* and five are using *The Open Group Architecture Framework (TOGAF)*. Section 3.2 compares these three approaches in more detail and the reasons why none of the approaches are suitable to design ITIs.

It was also without surprises, that we confirmed (figure 2.6) that the most common formats or tools used to represent ITIs, are diagramming software such as *Microsoft Visio* and presentation software such as *Microsoft PowerPoint*, which are tools where an ITI representation can easily be created with a good appearance, which may be important to present an ITI at a higher level of detail, however none of the tools have or support natively any kind of notation that can used to represent ITIs.

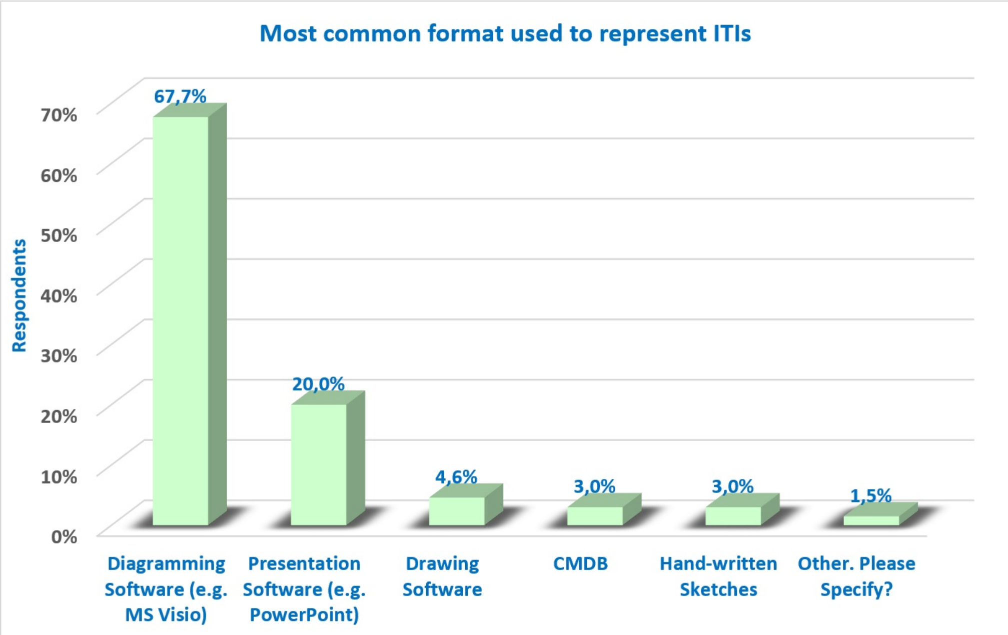


Figure 2.6: Common formats used to represent ITIs

Related comment (question 26)

I'm working as Infrastructure Architect for Capgemini, and I'm interested in how to formally represent an IT Infrastructure. Until now, I've mostly worked with Excel and Visio, but I quickly realize those tools don't scale properly for the job, and lack formalisms when we are dealing with large enterprise-scale Infrastructures. Lately I developed an interest for EA Tools, and I have now a license to use EA Sparx tool. TBH I'm still very new to UML modeling, Sparx EA and I find it overwhelming... I'm trying to make sense of it all right now (maybe it would have been easier for me had I come from development world). I believe what I am looking for would be called a UML profile to properly model today's IT Infrastructure. Today I stumbled upon your paper "Improving IT Infrastructures Representation: A UML Profile" (2012), where you mention an "ITI UML Profile" that seems to match what I'm searching.

These results provide us another confirmation of the research problem 1 (section 1.2.1), that the current state of the art to represent ITIs is characterized by informal notations such as sketches, pictures and other *ad-hoc* methods.

2.3.2 ITI Complexity

As was mentioned before, in section 2.3.2, there are several dimensions of complexity on ITIs, being the ITI size or heterogeneity of ITI components some of them. The ability to design an ITI and integrate hardware, software, networking, storage and support

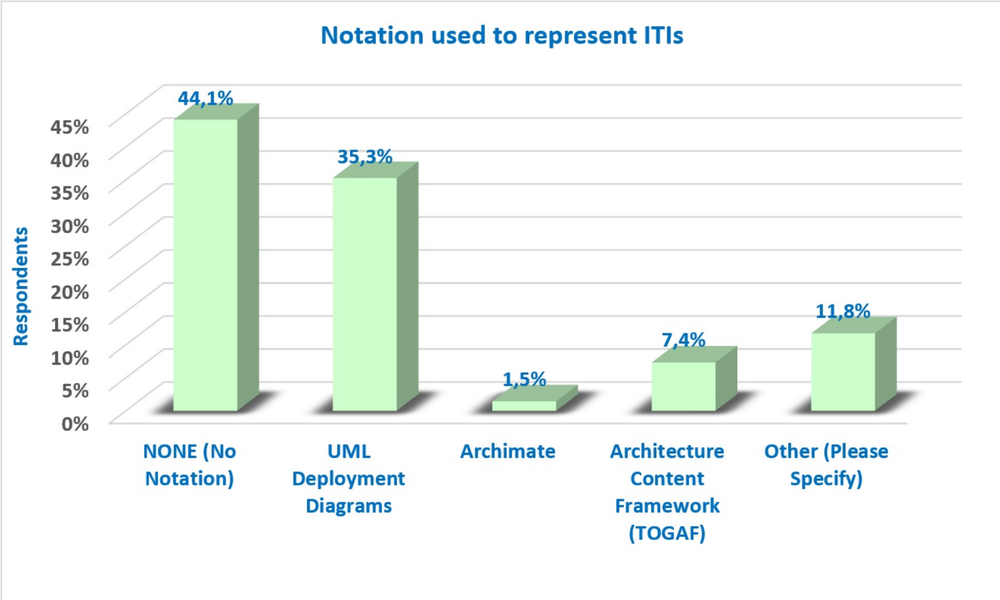


Figure 2.7: Notation used to represent ITIs

to provide the best possible combination in terms of cost, resiliency and features is a challenging task.

In the survey, 75% of respondents (figure 2.8), confirmed that designing ITIs, especially when involve components from multiple vendors is a complex activity which sometimes causes compatibility and integration issues.

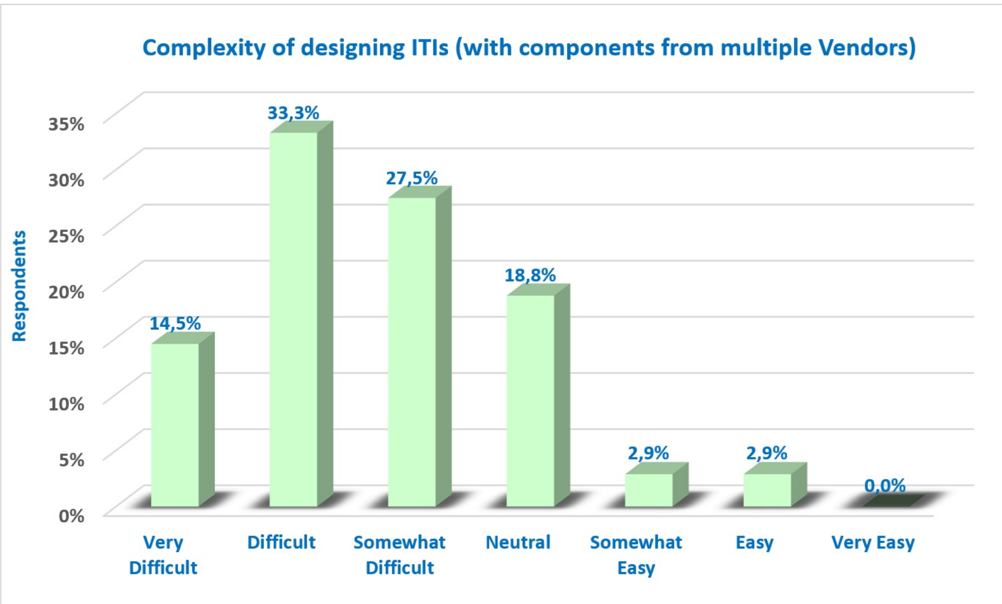


Figure 2.8: Complexity when having multiple vendors

Existing approaches, based on informal notation may provide a good-looking diagram, but miss important information. The capability to design ITIs without ambiguities and with all the relevant information (independently of their size and provenance), can help providing a single view of the ITI across the organization. Ignoring the importance of representing ITIs without ambiguity is a mistake since ITIs are becoming more and more complex and sound choices about which technologies to use and how to integrate them is crucial to have ITIs as dynamic as the business.

As can be seen in figure 2.9, most respondents confirmed that when dealing with complex ITI design issues, their very first action is to *ask colleagues* or *search for product blueprints*.

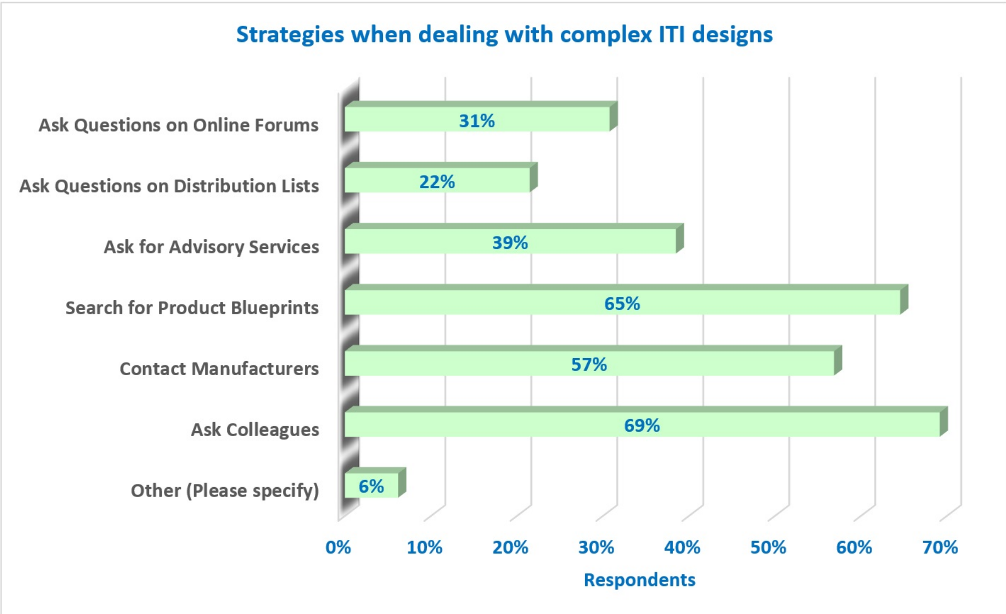


Figure 2.9: Complex ITI design

This is a confirmation of research problem 2 (section 1.2.1), that most of the knowledge is based on experience and resides in experts' heads and the evidence of the inability to express information about ITI design, such as best practices.

2.3.3 ITI Best Practices

The multitude of non-standardized methods, including the ad-hoc approaches in the design of ITIs, makes the process of expressing and reusing best practices in the ITI design very challenging. Current ITI design practices based on sketches, pictures and *ad-hoc* methods with limited or no formality, hampers the effectiveness to express ITI best practices and create standard solutions (including all the required configurations) that can be reused.

As presented in figure 2.10 more than 81% of survey respondents, are not satisfied with the existing approaches and 53,4% confirm that the current ITI representations are ineffective or completely ineffective. This highlight the importance of having an approach

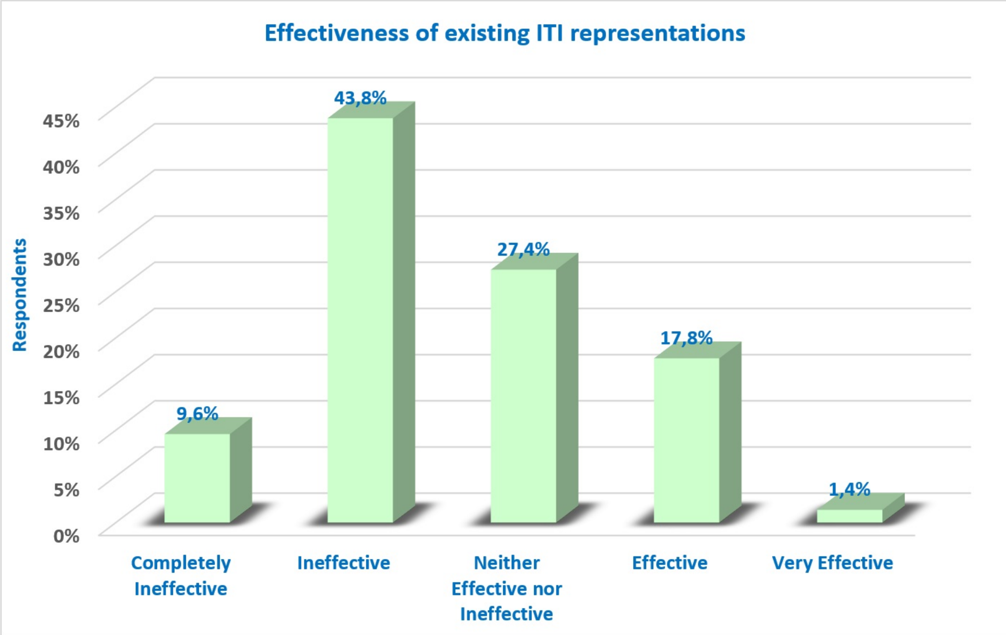


Figure 2.10: Complex ITI design

that allows ITI design experts to express best practices in the ITIs design, what confirms the research problem 2 (section 1.2.1).

It was also confirmed in the survey, that ITI designs are very informal and that with the current practices, a significant number of configurations are not included in the ITI design. It was also mentioned that the maximum number of attributes with the current design practices is 5 for each ITI component.

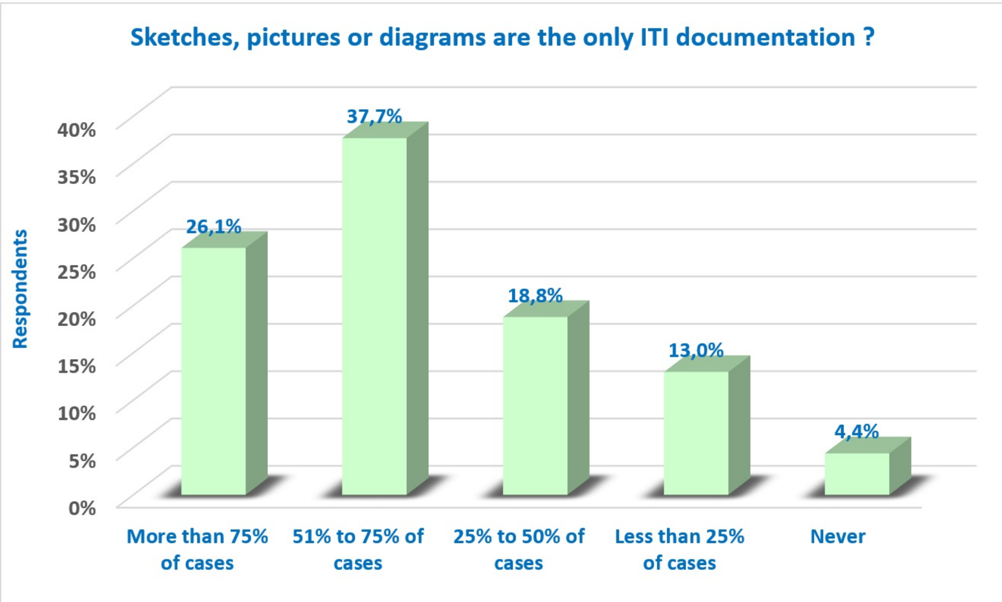


Figure 2.11: ITI documentation

What makes the use of best practices more complex, is the fact that, according to respondents (figure 2.11) most times, sketches, pictures and *ad-hoc* diagrams are the only ITI documentation available.

2.3.4 ITI Knowledge

The knowledge that can be extracted from the current design approaches is limited since most designs are informal and contain a very low number of attributes or configurations expressed. This is confirmed by the respondents in the answer to the question on how often these designs, contain information to support implementation activities, where the majority answered *Never* or in *less than 25% of cases* as can be seen in figure 2.12.

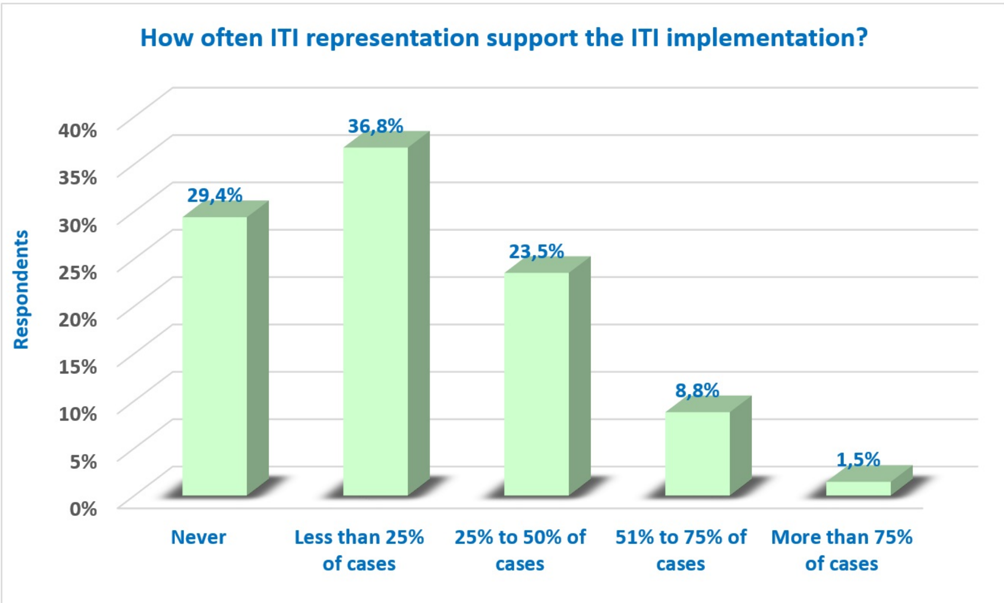


Figure 2.12: ITI implementation

Obtaining or storing ITI knowledge such as insights, experiences or lessons learned using the existing ad-hoc approaches is very limited, which lead most respondents to classify the effectiveness of existing practices as somewhat useless or neutral. This is aligned with our field experience that most of the knowledge is held by individuals, based on their accumulated experience, but not available for reuse. Respondents mentioned also that on-job training is the method used most often to share knowledge as presented in figure 2.13.

It was also confirmed that a significant percentage of ITIs are designed by non-experts such as managers, business decision makers, developers among others with some respondents commenting that at some stage everyone will be responsible for ITI designs and that ITI designs continue to be relevant.

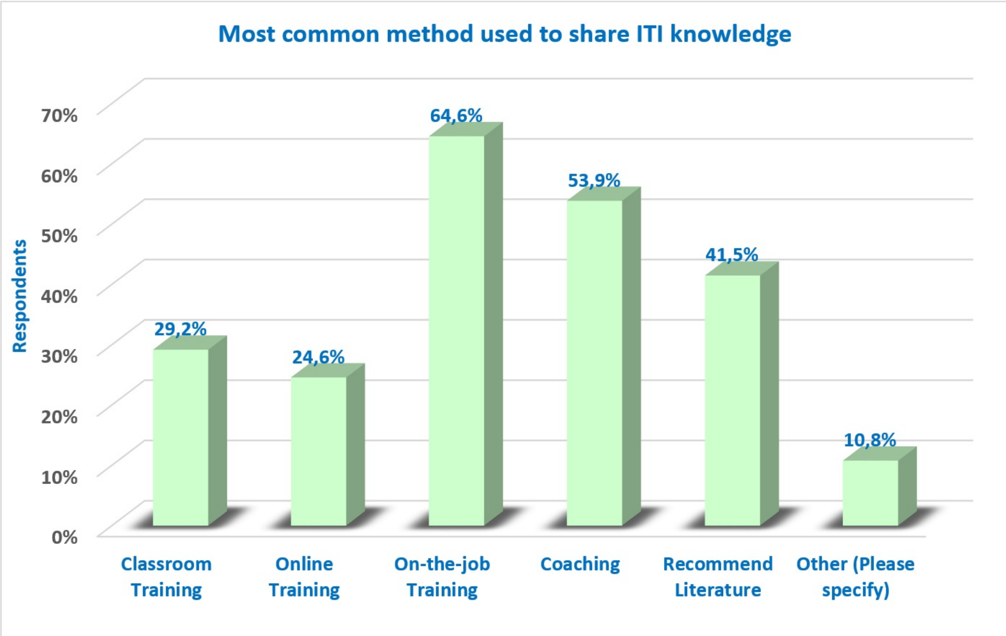


Figure 2.13: Common methods used to share ITI knowledge

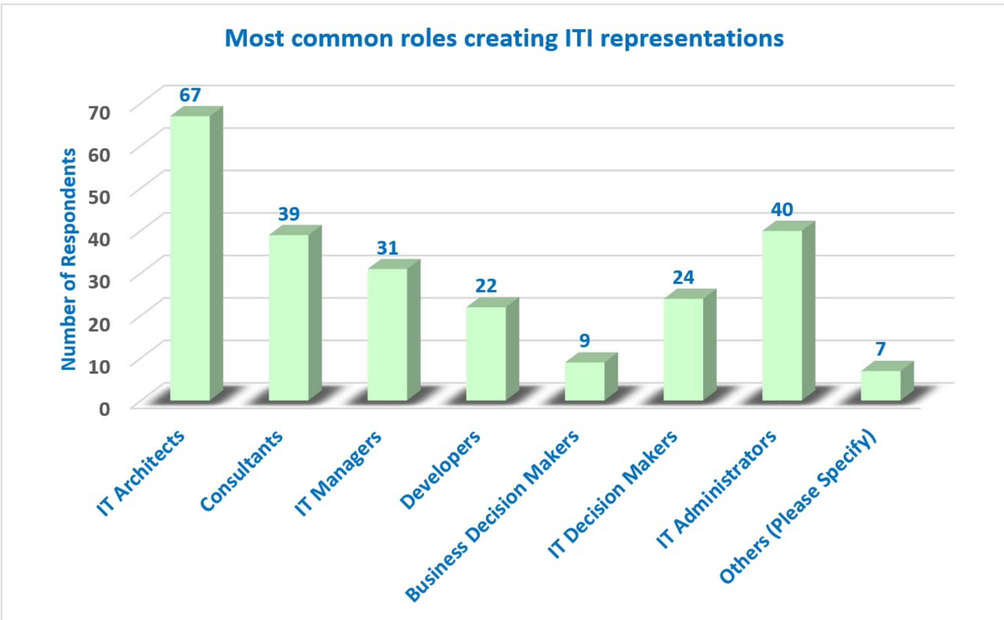


Figure 2.14: Common roles creating or adapting ITI representations

Related comments (question 26)

Your work is very good! Congrats. The problem of sharing IT Knowledge has to consider the capability skills of the team. Representation is very useful, but IT skills goes further than documenting and representing.

The majority of IT professionals, need to access these representations several times a week, which confirmed the importance of these ITI representations as it is presented in figure 2.15.

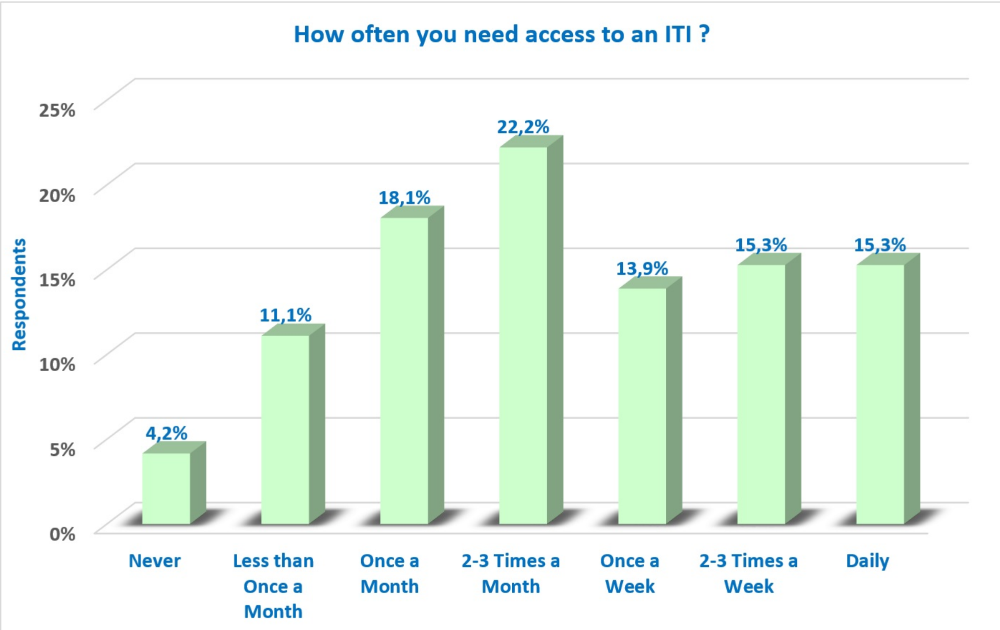


Figure 2.15: How often ITI representations are used

2.4 Summary

This chapter presents the state of the art in the design of ITIs and confirms the research problems tackled by this dissertation using an SLR and an online survey.

The work performed allowed us to confirm the increasing importance of designing ITIs for organizations and the relevance and importance of this research work, based on the number of papers returned from an SLR and in the significant percentage of users that need to access ITI designs in a daily basis.

In summary, the online survey revealed clearly that the current state of the art with the design of ITIs (which is mainly based on pictures, sketches and other diagrams with low quality) is poor. Most users consider useless the current ITI design practices and existing methods to reuse ITI best practices are not effective. Existing practices are also not good to store, capture or sharing technical knowledge, insights, experiences or lessons learned about the design of ITIs which explains the reason for having so many different tools used to design ITIs in a universe of 117 respondents.

Chapter 3

Modeling ITIs

Contents

3.1 Introduction

To represent ITIs, there has to be a way to represent all components that are part of an ITI and their relations. For that purpose, we need a common language that captures all the relevant knowledge of the ITI in a readable, searchable and reusable way. Since several specifications of model driven approaches have been proposed, we evaluate the candidate ones to fulfill the goal of modeling ITIs.

3.2 Candidate Metamodels for ITIs

Picking on previous considerations, a metamodel is a modeling ontology, and as such, it should describe the constructs used in the modeling process, their properties and relations. In other words, a metamodel describes the grammar and semantics of a modeling language. There are several metamodels constructed with different purposes and it will not be accurate to assert that one approach is better than another in all circumstances. The following sections present an overview of modeling languages that provide constructs for modeling ITIs with precision. We review their metamodel, since the latter expresses their grammar, i.e., their constructs and composition rules to assess the extent to which they support our objectives.

3.2.1 UML2 Metamodel

UML [OMG08] is a general-purpose modeling language embodying a collection of best engineering practices that have proven successful in the modeling of large and complex systems of a wide range of domains. Under the stewardship of the Object Management Group (OMG), UML has emerged as the software industry's dominant modeling language. ITIs are modeled in UML with Deployment Diagrams. The latter allow representing the hardware for a system, the software that is installed on that hardware, and the middleware used to connect machines.

We described in [Sil+12b] how we found evidence that Deployment Diagrams are among the less used UML2 diagrams. A possible interpretation for this phenomenon is that UML2 offers limited modeling constructs (e.g., nodes, components and associations), that do not cope "as is" with the required diversity for modeling ITIs [Pet13]. UML2 has a comprehensive coverage of the whole lifecycle in software development with a large specification, spanning more than 900 pages [OMG08], and is in, some aspects, too abstract. In short, "plain vanilla" UML provides no specialized stereotypes for the many concepts and association types used in any ITI, making it a weak candidate for modeling ITIs. Figure 3.1 shows the UML metamodel extract corresponding to the Deployment Diagrams.

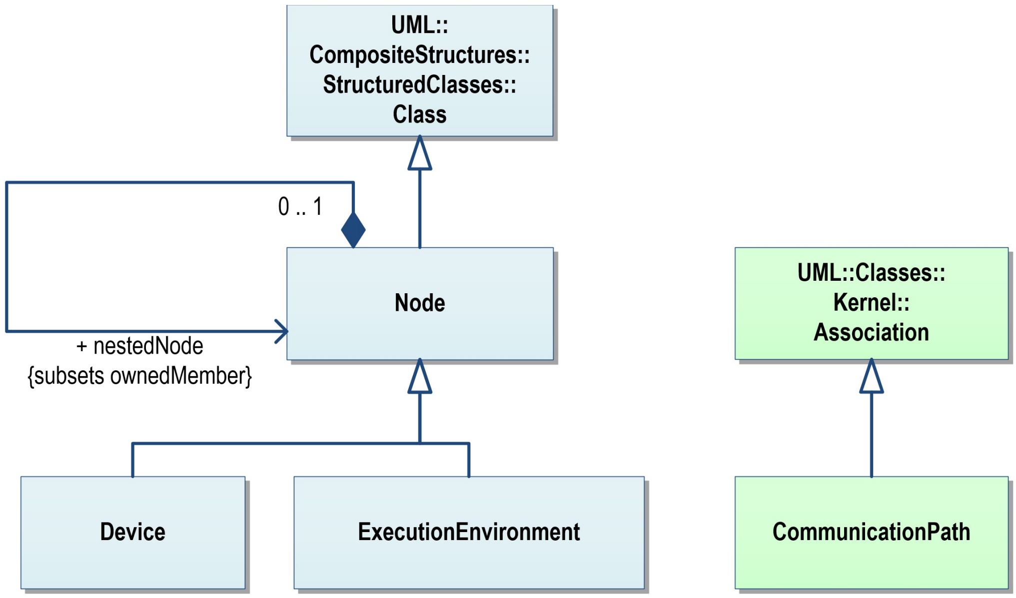


Figure 3.1: UML metamodel extract corresponding to the deployment diagram

3.2.2 TOGAF Content Metamodel

TOGAF is a framework for enterprise architecture developed by the Open Group Architecture Forum in the United States, which provides a comprehensive approach for designing, planning, implementation, and governance of an enterprise information architecture. TOGAF models include four levels: Business, Application, Data, and Technology. It relies heavily on modularization, standardization and proven technologies and products. Its latest release (version 9.1), at the time of writing, also includes a metamodel, called Content Metamodel [Ger+10], that defines all types of building blocks that exist within an architecture and how they are related to each other, to allow architectural concepts to be captured, stored, filtered and queried in a structured and consistent manner. Notice that ITI in TOGAF terminology is called Technology Architecture.

As can be seen in figure 3.2 from an ITI perspective, there are few concepts in the Content Metamodel: The *Platform Service* that represents the support for delivering applications, the *Logical Technology Component* used to represent a class of technology products, and *Physical Technology Component* to represent specific technology products. As with UML, these general concepts from TOGAF´s Content Metamodel allow, in theory, to model ITIs. However, the lack of specialized stereotypes and relationships appears to be a serious hindrance for its effective adoption. Furthermore, some authors argue that TOGAF´s Content Metamodel lacks a formal ontology to mitigate its ambiguities and inconsistencies, no cardinality is available and no possibility to define well-formedness rules. [Ger+10].

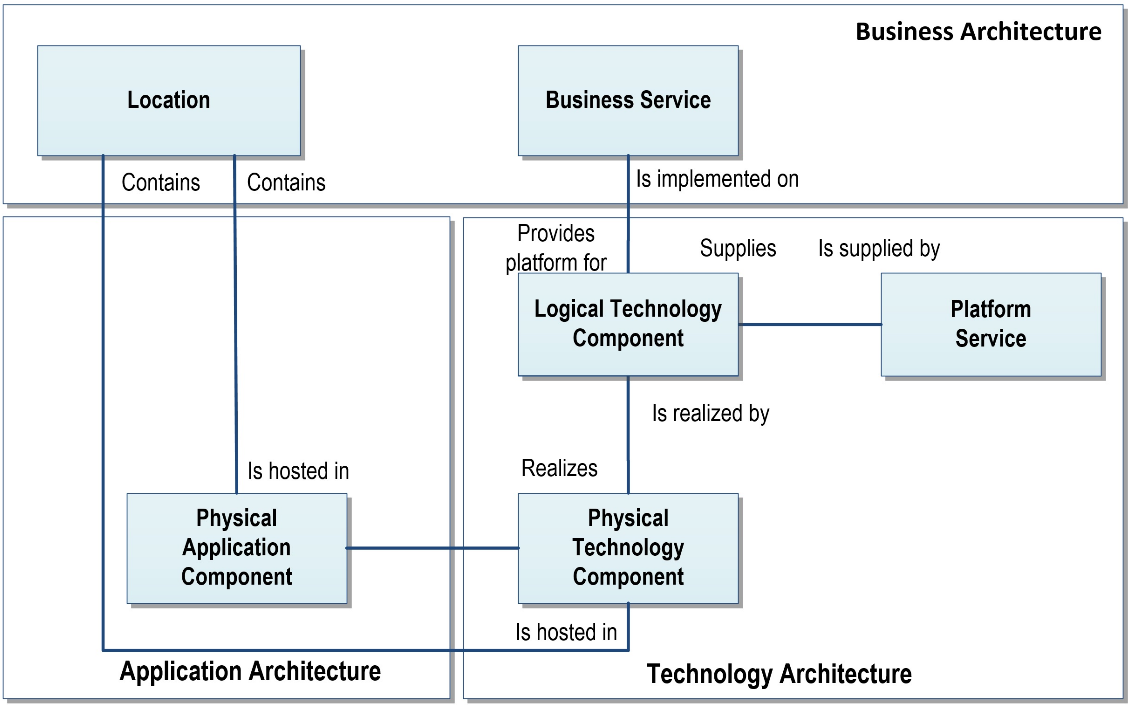


Figure 3.2: TOGAF 9.1 content metamodel extract corresponding to ITIs

3.2.3 ArchiMate Metamodel

ArchiMate [HP12a] is an open architecture modeling standard with focus on the visualization of viewpoints. The metamodel encompasses several enterprise architecture domains (Business, Application, Information, Technology) and is an extension to the UML2.0 standard [OMG08]. As can be seen in figure 3.3, *Node* is also the main structural concept and is specialized in *Device* (e.g., servers) and *System Software* (e.g., operating system called *Execution Environment* in UML). The color green is used to distinguish passive structure, yellow for behavior and blue for active structure.

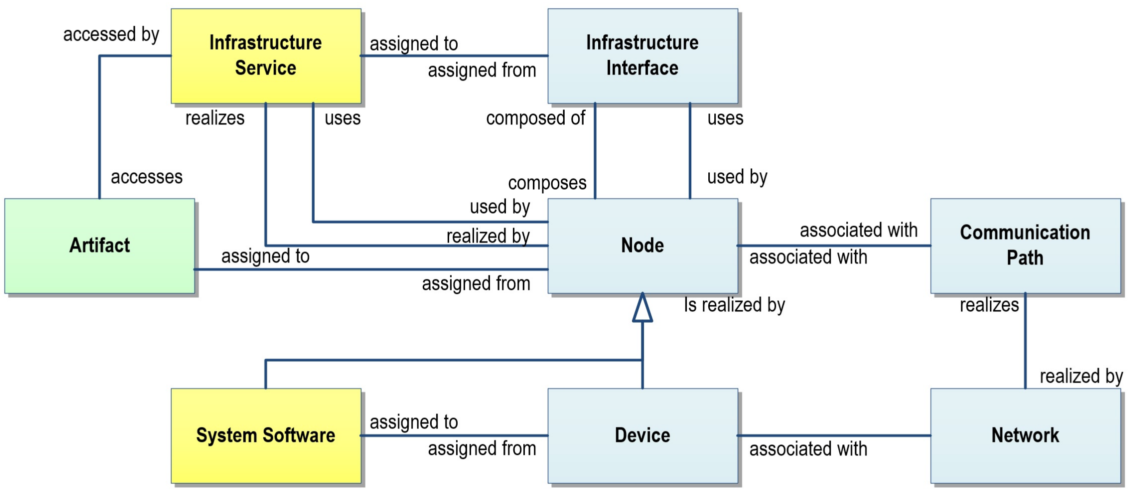


Figure 3.3: ArchiMate metamodel extract corresponding to ITIs

The *Infrastructure Interface* is the "logical" location where the Infrastructural Services offered by a *Node* can be accessed by other *Nodes*. The *Communication Path* and *Network* are used to connect interrelated components in the technology layer. The *Artifact* (also taken from UML 2.0) represents a physical piece of information and can be deployed to a *Node*. ArchiMate's technology architecture metamodel is more detailed than the corresponding TOGAF metamodel, namely by allowing to model the hardware platforms and communication infrastructure. However, it is still too generic and with more focus on describing the relationships between layers than providing clear guidelines and rules on how to model the various components of the technology architecture. It is argued in the ArchiMate specification [HP12a] that modeling infrastructure components, such as routers or database servers, would add a level of detail that is not useful at the enterprise abstraction level.

3.2.4 Overall Assessment

The purpose of this section is to evaluate the candidate metamodels (summarized in table 3.1) based on some characteristics or criteria that we found relevant for the evaluation of ITIs. To highlight the characteristics of the various modeling approaches, we chose the following list of evaluation criteria:

- **Expressiveness.** The modeling language should provide relevant Universe of Discourse (UoD) concepts such as client and server computers, network devices, physical and logical connections, middleware and application components;
- **Relevance.** The modeling language should be supported by a well-known organization committed to its evolution;
- **Models availability.** We should have access to real-world ITI models expressed in the modeling language, or be able to reverse engineer them;
- **Metamodel instantiability.** From existing or reverse-engineered models, we must be able to instantiate the corresponding metamodel;
- **Understandability.** The required effort to learn the notation and to recognize the modeling concepts and their interrelationships should be as small as possible;
- **Extensibility.** The metamodel should be extensible to new concepts or existing concepts should be amenable to further detail. Extensions should not cause revisions of existing definitions.

Each of these evaluation criteria will be applied to the selected modeling languages to allow a comparison to be made.

3.2.4.1 UML Overall Assessment

- **Expressiveness.** UML allows the representation of ITIs by using Deployment Diagrams. This kind of diagram has a limited number of abstractions (physical nodes, software components, associations and dependencies). To increase the representation richness, we must use stereotypes and tagged values. This option limits models' portability, namely if we intend to use some ITI's models capture tool.

- **Relevance.** The OMG has been committed in the standardization and evolution of UML. The latest published version is 2.5 which is formally a minor revision to the UML 2.4.1 specification. All OMG specifications are subject to continuous review and improvement.

- **Models availability.** Although an eXtensible Markup Language (XML) based format for UML models' portability has been proposed [OMG07] and is supported by several tool vendors, the examples of XML Metadata Interchange (XMI) files found in the web are few and relate mostly to toy examples and to other diagrams (mostly class and use case ones) rather than deployment diagrams. Most UML tools now support at least some features of UML 2.x. However, we could not find tools to capture UML deployment diagrams out of existing ITIs.

- **Metamodel instantiability.** The UML 2.x metamodel is fragmented across many packages, which hampers its instantiability.

- **Understandability.** The concepts of UML, the language and notation are aspects easy to recognize. However due to poor UML expressiveness concerning ITIs, models may be complex and difficult to read and understand.

- **Extensibility.** The UML metamodel has three extensibility mechanisms, which are tagged values, stereotypes and constraints. Tagged values allow arbitrary information to be attached to model elements. This extensibility mechanism allows users to define new element properties for any model element. Stereotypes allows subclassification of model elements. Stereotypes can be used to introduce additional distinctions between model elements that are not explicitly supported by the UML metamodel. Constraints allow new semantic restrictions to be applied to elements.

3.2.4.2 TOGAF Overall Assessment

The application of the previous characteristics to TOGAF is as follows:

- **Expressiveness.** TOGAF Architecture Content Metamodel addresses four main architecture domains (Business, Data, Application and Technology). The Technology Architecture includes high level concepts such as the "Logical Technology Component" and the "Physical Technology Component" that are required to support the deployment of business, data, and application services. To be able to represent a detailed ITI, the content metamodel has to be extended.

- **Relevance.** TOGAF is the *de facto* global standard for Enterprise Architecture developed and maintained by the Open Group Architecture Forum[1], comprised of more than 200 enterprises. The latest TOGAF version published was 9.1 [Har11].
- **Models availability.** The available technology architecture models are very limited and generic. The TOGAF documentation mentions that the TeleManagement Forum (TMF)[2] developed technology models relevant to the telecommunications industry, but we were unable to access them.
- **Metamodel instantiability.** TOGAF does not include a notation and does not provide rules or classification for relationships. The TOGAF standard does not specify the language or languages that can be used for modeling. This is mainly due to the fact that TOGAF is more an enterprise architecture framework than a modeling language.
- **Understandability.** The content metamodel provides a single view with entities for all the architecture domains. TOGAF artifacts are agnostic and can be created using any modeling language. In terms of understandability, TOGAF is too high level.
- **Extensibility.** TOGAF presents a simple conceptual definition of a modeling language, with a core set of elements and six metamodel extensions namely *governance* with concepts such as goal, objective, measure and contract, *services* to enable description and management of IS services, *process modeling* to allow detailed modeling of process flows, *data* to enable management of data and introduces the concept of logical and physical data components and *infrastructure consolidation*, adding the location entity and logical and physical application components and *motivation* to enable measurement of business performance.

3.2.4.3 ArchiMate Overall Assessment

The application of the previous characteristics to ArchiMate is as follows:

- **Expressiveness.** The ArchiMate was created mainly to address the Enterprise Architect's modeling requirements, which is one of the main differences from other languages such as UML or Business Process Modeling and Notation (BPMN). It is a modeling language with notations and with the capacity of providing a higher level of abstraction, which is very helpful in getting a high-level view of an organization.
- **Relevance.** Currently in version 3.0, the ArchiMate is an Open Group standard developed and maintained by The Open Group ArchiMate Forum. In the last version,

[1] http://www.opengroup.org/getinvolved/forums/architecture
[2] https://www.tmforum.org/

the consistency and structure of the language have been improved, definitions have been aligned with other standards, and its usability has been enhanced.

- **Models availability.** There are several examples of ArchiMate models for describing ITIs, most of them provided as templates in the more than 20 software tools that allow the creation of ArchiMate models.

- **Metamodel instantiability.** ArchiMate provides a high-level generic metamodel, as well as detailed per domain or layer metamodels. The metamodel instantiability is supported by various tool vendors and consulting firms.

- **Understandability.** ArchiMate has a visual language with a set of entities and relationships with their corresponding iconography for describing, analyzing, and communicating many concerns of Enterprise Architectures, as they change over time. It defines a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure. This insight helps stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.

- **Extensibility.** The ArchiMate language contains only the elements and relationships that are necessary for general architecture modeling. It is possible to enrich ArchiMate elements and relationships by means of a "profiling" specialization mechanism.

3.2.5 Metamodels Comparison

Table 3.1 provides a comparative analysis of the previously described metamodels. The intersections between rows and columns are depicted as a symbol: Strong: ●; Moderate: ⊙; Weak: ○; None/Basic: —

Table 3.1: Comparative analysis of metamodels

Metamodel	**Expressiveness**	**Relevance**	**Models availability**	**Instantiability**	**Understandability**	**Extensibility**
UML	○	●	○	○	○	●
TOGAF	○	●	—	○	○	⊙
Archimate	⊙	●	⊙	⊙	○	⊙

Due, on one hand, to the lack of expressiveness in the ITI domain evidenced by all the three alternatives and, on the other hand to the extensibility features of UML, we decided to create an UML Profile for ITIs.

3.3 Metamodel Driven Measurement

To perform the representation of ITIs and be able to evaluate them, we must be able to express without ambiguities some descriptive variables on a more than ordinal scale on our UoD. In order to achieve that, we must agree on what we are talking about, that is, we need a well understood representation of our UoD. This is where ontologies come to the rescue. Ontology is the science of what is, of the kinds and structures of the objects, properties and relations in every area of reality [Smi03]. However, in Computer Science, we use the word "ontology" in a more loose way, to designate an abstract representation of the relevant concepts and their relationships in a given domain. If that domain is the one of modeling, then, instead of talking about a "modeling ontology", we simply call it a "metamodel". There are several ways of formally defining an ontology. Generic languages, as well as domain-specific ones, such as Ontology Web Language (OWL) [MH04], have been proposed for this task. In the scope of metamodeling, the most widely used ontology language is UML, by using meta-class diagrams enriched with Object Constraint Language (OCL) constraints [CP99], as happens with all Meta-Object Facility (MOF) based metamodels published by the OMG[3]. Since we have identified the need for a metamodel, we turn our attention to the metrics definition and collection problems. Metrics should be formally defined to avoid subjectivity in their interpretation and in the implementation of collection instruments. In this dissertation, we will use the Metamodel Driven Measurement (M2DM) technique in which metrics are defined as OCL expressions upon the adopted metamodel. In addition to the formality granted by the use of this constraint language, OCL expressions can be automatically evaluated upon the instantiated metamodel using an ITI evaluator component. The M2DM technique was originally proposed upon the Generic Object Oriented Design Language? Yes! (GOODLY) metamodel [Abr+99]. The M2DM technique was used in several distinct contexts, such as the ones of defining Object-Oriented (OO) design metrics based upon the UML 1.x metamodel [BA02], expressing object-relational database schema metrics based upon the Common Warehouse Metamodel (CWM) [Bar+06], evaluating components reusability upon the UML 2.0 metamodel [GA07] assessing components composition using the CORBA Components Metamodel (CCM) [GA05] or assessing the quality of BPMN models [CB15].

[3]http://www.omg.org/spec/

3.4 ITI Profile Metamodel

The adoption of UML2 as a standard and the advances in recent releases made this modeling language applicable to multiple domains and contexts. However, in the context of ITI design, UML2 lacks some important concepts in terms of ITIs such as networking, storage and computer systems. Fortunately, UML makes provisions for its own extension, by allowing "customization" to a specific area or domain, with a so-called UML Profile. The latter is a coherent collection of UML modeling elements (stereotypes, tagged values, and constraints) that allows refining the standard semantics in strictly additive manner (i.e. without contradicting it). For instance, a profile may use a stereotype to refine the concept of *Node*. Several UML profiles have been proposed in the literature and some of them have been endorsed by the OMG itself. Examples include a profile for aspect-oriented software development [Ald+03], a profile for requirements management of software and embedded systems [Arp+11], a profile for business process modeling [LK05], and a profile for modeling real-time embedded systems [OMG09].

The decision for extending UML, instead of developing a Domain Specific Language (DSL) from scratch, was based on the following rationale:

- There is a large community, both in industry and academia, that understands and actually uses the UML language;
- Extending the UML language allows reusing existing UML modeling elements, with well-defined syntax and semantics;
- There are tools that support the development of UML profiles.

The ITI profile uses the UML 2 Superstructure specification [OMG08] which is the basis for any UML profile and also uses the OCL 2.0 specification for all constraints specified in OCL. Figure 3.4 shows these dependencies.

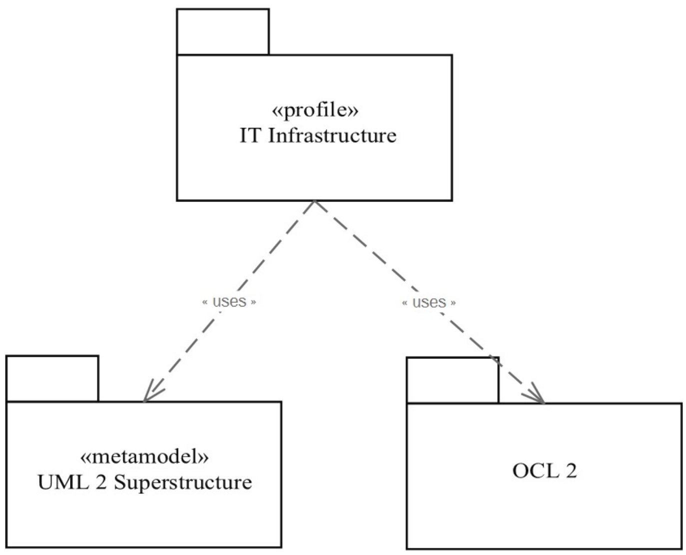


Figure 3.4: ITI profile dependencies

The created ITI metamodel was intended for infrastructure engineers to design and verify the correctness of their models. To design the ITI metamodel we defined several rules and constraints to avoid the most common metamodel problems outlined by [Fue+03], such as empty elements, non-accessible elements and duplicated names and derived associations. We also followed metamodels design principles such as *(i) modularity* where we tried to minimize the number of dependencies between modules, *(ii) extensibility* through the creation of an ITI profile with multiple stereotypes, tagged values and constraints to make it more related with the ITI domain specificities and *(iii) reusability*, since the constructs from the ITI metamodel are instances of predefined constructs. More details regarding our ITI metamodel are available in [Sil+12b] and in the following sections.

3.4.1 ITI Profile Foundations

The profile is organized in a set of packages named "ITI profile foundations", as presented in figure 3.5. The following principles were used to create this profile:

- It should support vendor independent modeling of both infrastructure software and hardware networking and storage components and the relationships between them.
- It must provide enough modeling constructs to cover the ITI design process.
- It must allow multiple actors to design ITI infrastructures.

To document ITIs, the first step was to define a metamodel with a set of rules and constructs, that were mainly based on the professional experience of the author in this field and also based on other general metamodels such as ArchiMate's Technology Architecture.

Each package includes a set of meta classes, stereotypes and constraints that extend the UML metamodel to capture and express as much information as desired in the domain of ITIs. Each stereotype was enriched with additional attributes (called "tagged values" in UML), to allow the capture of information for each model element. For example, we may need to capture the disk size information on a server and this can be done with a tagged value.

3.4.1.1 Stereotype attributes

The attributes chosen for each stereotype were mainly based on our field experience and on the CMDB repository of two organizations. These databases contained IT assets commonly referred to as Configuration Items (CI) with descriptive relationships between such assets. For security reasons, we got no approval to disclose the content of the databases or the name of the organizations, but the CMDB software used on the first organization was *EasyVista Service Manager*[4] and *Microsoft System Center*[5] in the second organization. In

[4] http://www.easyvista.com/product/servicemanager/configuration-management/

[5] http://www.microsoft.com/SystemCenter

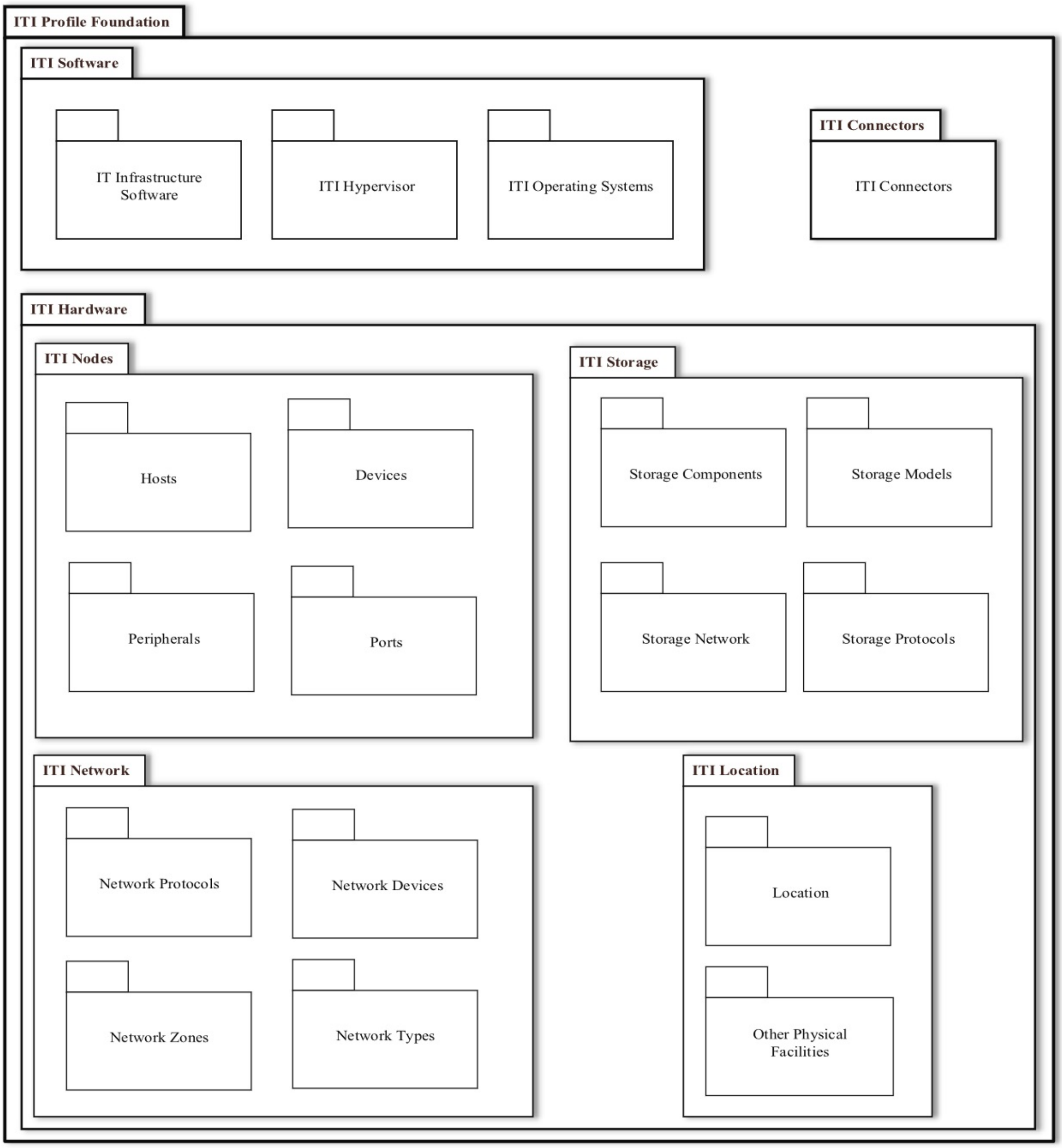


Figure 3.5: ITI profile foundations

both cases, the database engine was *Microsoft SQL Server*[6], which simplified the process of cleaning and extracting data to a format we could easily use. Since we only had access to the database files, to extract the data we had to install the *SQL Server* software in a server, restore databases and reverse engineer the schema of each database.

Regarding the *EasyVista Service Manager*, we extracted all the data, but we had not many insights in terms of attributes, since we realized that most IT assets of the organization were not in the CMDB, because the automated discovery inventory was not enabled and most content was loaded manually.

[6]http://www.microsoft.com/sql

The CMDB of the second organization, based on *Microsoft System Center*, was completely integrated with the rest of the ecosystem (e.g. Customer Relationship Management (CRM) systems, directory services databases, monitoring tools, etc.) and the automated discovery inventory features were enabled. Almost all IT assets were in the CMDB and the main challenge was to clean and extract the data we needed to make sure we do not miss any important stereotype and their attributes in our model. The figure 3.6 is a diagram representing this CMDB, which had more than 800 tables (represented by dots) and hundreds of relations (represented by lines). We retrieved more than 500,000 rows of data which took us more than 24 hours in a server with 8 CPU cores.

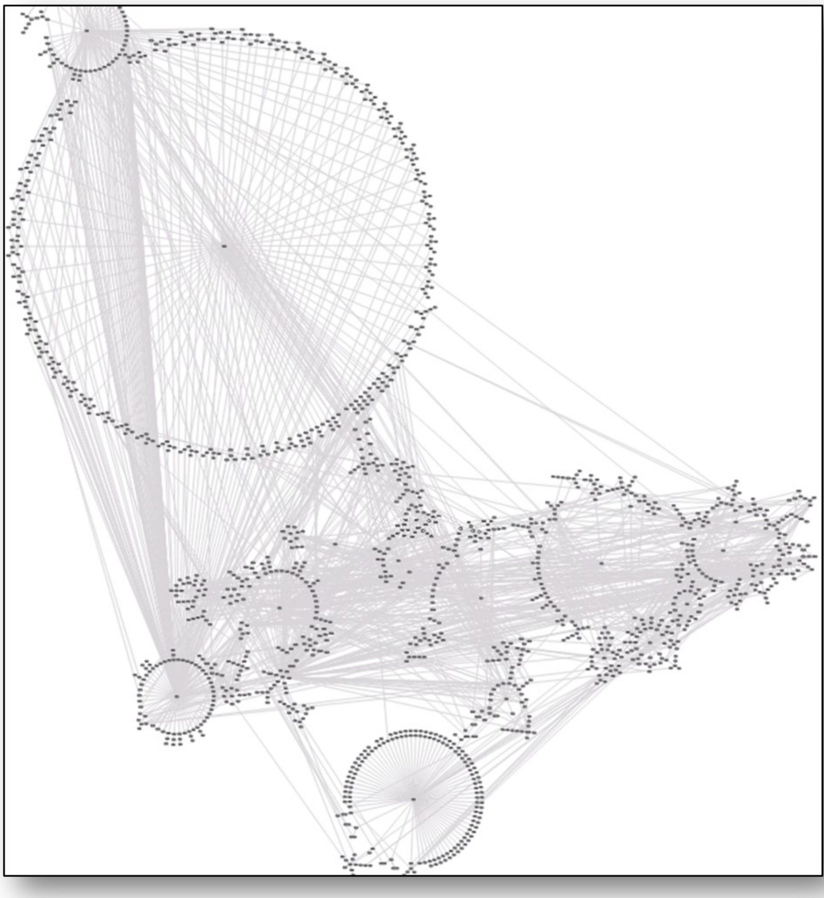


Figure 3.6: CMDB of the organization used to enrich the profile

The list of attributes to choose for each stereotype was also inspired on the CIM standard [DMT99] created by the Distributed Management Task Force (DMTF). The latter is a worldwide initiative spearheaded by industry-leading technology companies such as AMD, Broadcom Corporation, CA, Cisco, Citrix Systems, EMC, Fujitsu, HP, Huawei, IBM, Intel, Microsoft, NetApp, Oracle, RedHat, SunGard and VMware. CIM was created to provide a common approach to the management of systems, networks, applications and services and enable multiple vendors to exchange semantically rich management

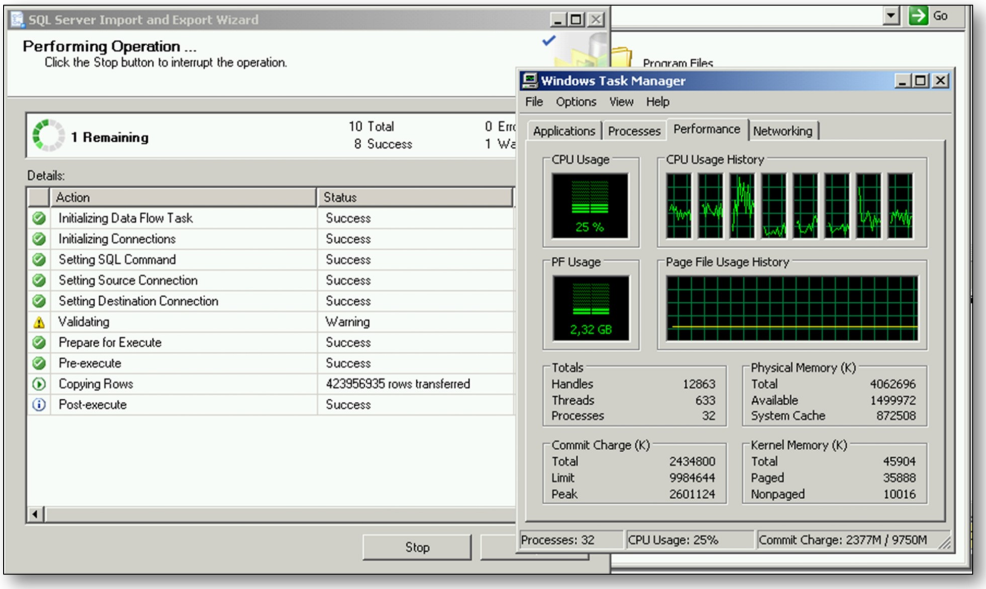


Figure 3.7: Extracting data from the CMDB

information between systems throughout the network.

3.4.2 Metamodel Notation and Structure

Our metamodel was dubbed **itiPM**[7], and is structured around five main packages:

- **ITI Nodes**. Implements the concepts of clients and servers.
- **ITI Network**. Defines networking constructs used to connect clients and servers.
- **ITI Storage**. Contains remote storage constructs, that are accessible mainly to servers.
- **ITI Software**. Defines constructs used to represent the ITI software running on clients and servers.
- **ITI Location**. Contains the constructs that allow the identification of physical location for each component.

Each package has meta-classes that are connected using a set of meta-relationships created for ITIs (figure 3.8). These relationships extend UML metaclasses such as Association, Composition, CommunicationPath and include:

- **itiConnectsTo**. Indicates that a component connects to another component.
- **itiContains**. Defines a relationship of composition between components.
- **itiExecutesOn**. Indicates that a component runs on another component.
- **itiCreates**. Used when one component creates another.
- **itiResidesIn**. Used mainly to identify the location where a component resides.

A complete description of the **itiPM** is presented in appendix C.

[7] all constructs in the **itiPM** are preceded by the word **iti** and are spelled in the singular. As an example *itiServer* should be read as *IT Infrastructure Server* or simply *Server*.

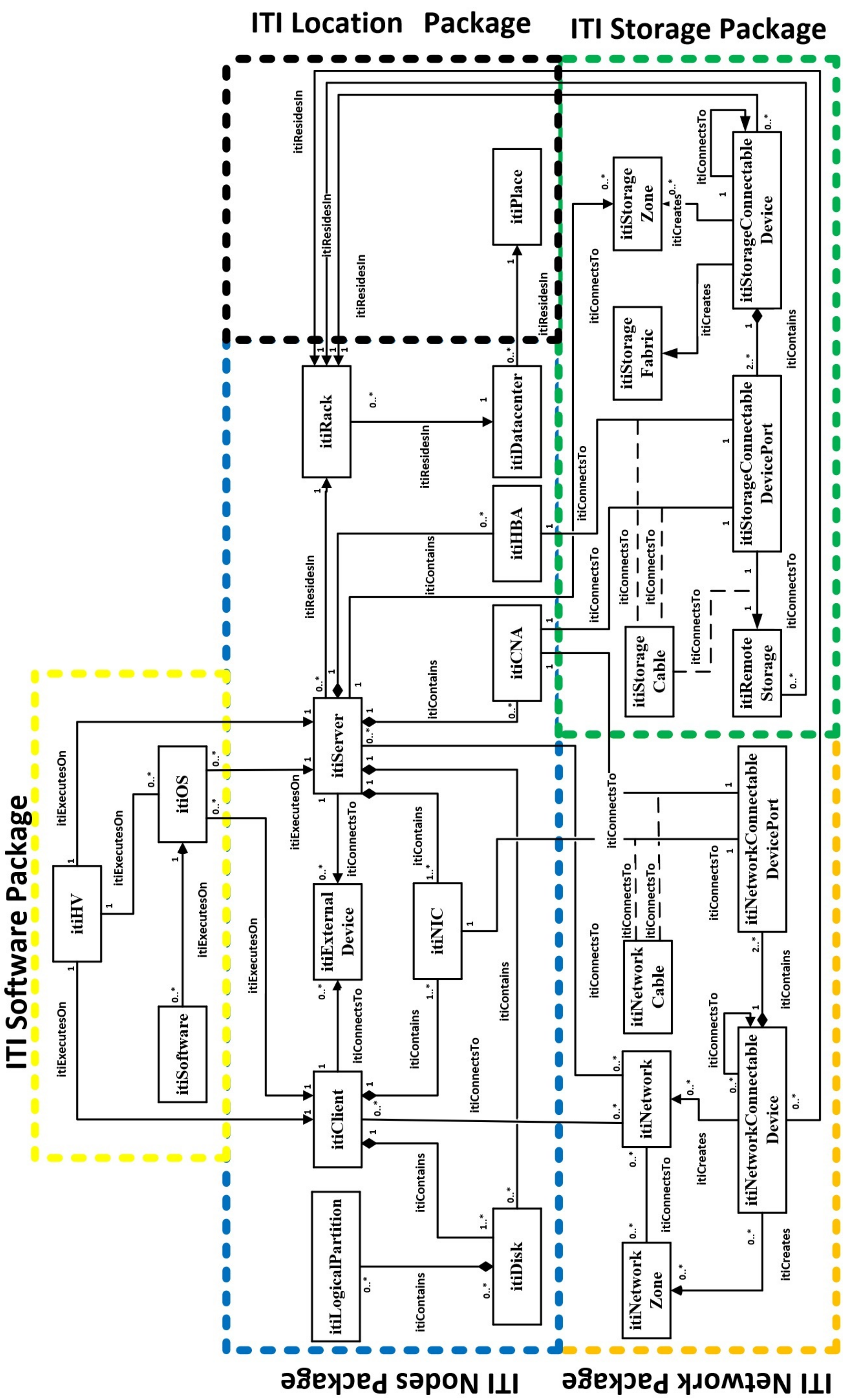


Figure 3.8: IT profile metamodel (itiPM)

3.4.3 ITI Nodes Package

The **ITI Nodes Package** includes the concept of *itiServer* (Server) and *itiClient* (Client) which inherit the properties of an *UML2 Node* (from *UML::Deployment*) and are part of **itiPM** as represented in Figure 3.9.

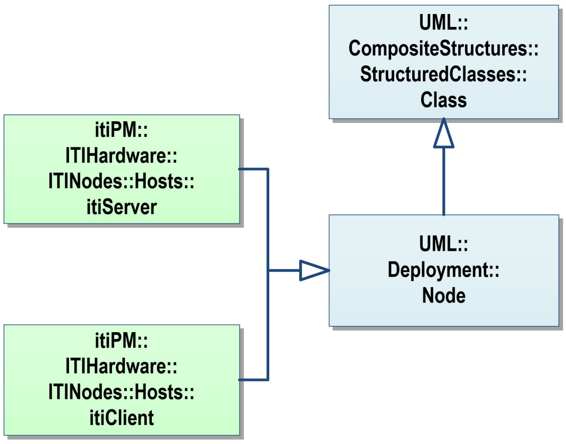


Figure 3.9: Inheritance of itiNode

The **ITI Nodes Package** of **itiPM** contains also other constructs to model other components, as can be seen in Figure 3.10 and will be detailed in the next sections.

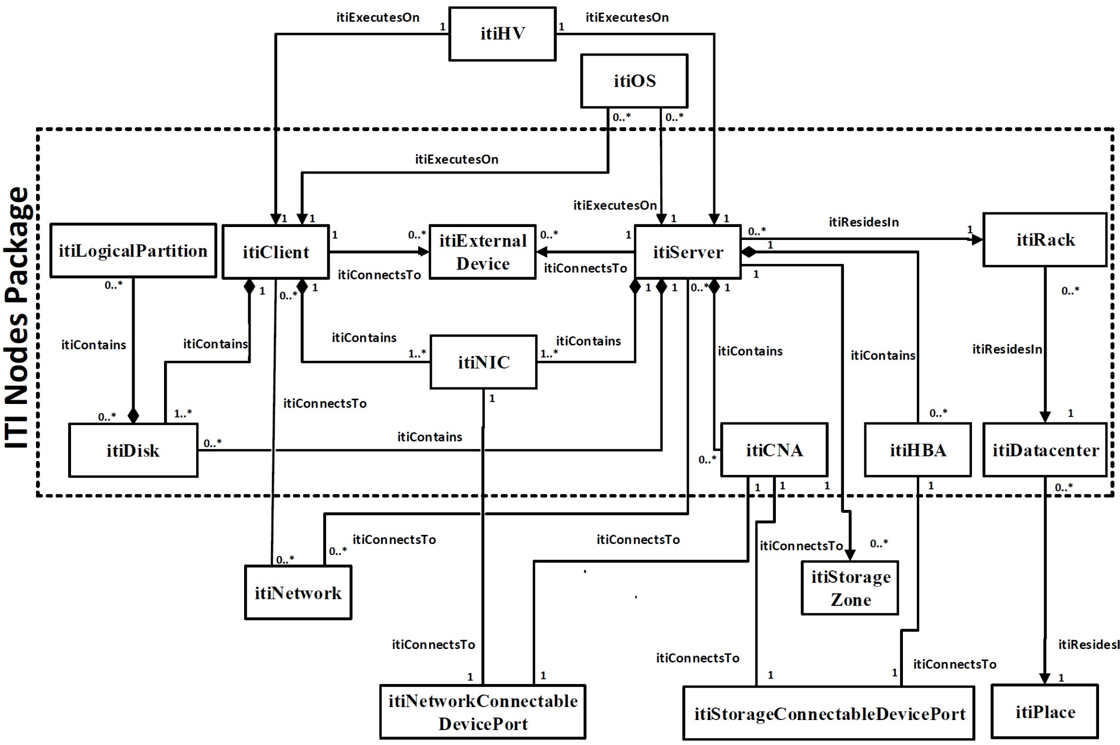


Figure 3.10: ITI nodes package (itiPM)

3.4.3.1 itiDatacenter

An *itiDatacenter* represents the facilities where all infrastructure elements may reside. This component may not exist in all organizations, but when it does, it normally includes one or more *itiRacks*.

3.4.3.2 itiRack

An *itiRack* consists in an enclosure where components (e.g. *itiServer*, switches, storage devices) can be mounted. Notice that these components may also reside outside an *itiRack*. For instance, an *itiServer* can be in a remote location where *itiRacks* or datacenters do not exist.

The unit of measure of an *itiRack* is the *Rack Unit* which describes the height of the equipment and is represented in the metamodel as a tagged value. The *itiNetworkConnectableDevice*, *itiStorageConnectableDevice* and *itiRemoteStorage* are physical components that will be detailed later.

3.4.3.3 itiServer

One of the main concepts of **itiPM** is *itiServer*, which is a specialization of the *Node* concept of UML 2.0. An *itiServer* is used to model physical server hardware components. An *itiServer* is a physical server (computational resource) where software can be deployed for execution. There are different types of *itiServer*. Each one has some particular characteristics, but shares the same configurations of the *itiServer*. Examples include clusters, blade-servers, supercomputers, and mainframes, among others.

3.4.3.4 itiClient

The *itiClient*, is another specialization of the *Node* concept of UML 2.0. An *itiClient* is used to model physical client hardware components. An *itiClient* is a physical client (computational resource) where software can be deployed for execution.

When designing ITIs, the concept of client is frequently used to represent who consumes the service. Because there are many different types of client devices, that share the same characteristics of *itiClient*, they were made available in the metamodel to simplify the design. Examples include mobile phones, tablets and personal computers, among others.

3.4.3.5 itiNIC

The Network Interface Controller (*itiNIC*) is the component that allows an *itiServer* and an *itiClient* to connect to networks. Since this component is part of all kind of devices is mandatory and it is assumed that *itiServer* and *itiClient* have at least one network adapter, each.

The relationship type among classes is a composition, because an *itiNIC* is part of an *itiServer* or *itiClient* and cannot exist in isolation.

3.4.3.6 itiHBA

The *itiServer* requires access to local or remote storage. In case of access to remote storage, a special adapter called Host Bus Adapter (HBA), is needed. In the metamodel the HBA is known as *itiHBA*.

3.4.3.7 itiCNA

The functionalities of *itiNIC* and *itiHBA* can be combined in a single adapter, called Converged Network Adapter (CNA). In the metamodel the CNA is known as *itiCNA*.

3.4.3.8 itiExternalDevice

An external device (*itiExternalDevice*) represents any device that can be connected to servers or clients or both, such as a portable hard drive or printer. The *itiExternalDevice* can be connected to *itiServer* or *itiClient*.
The *itiExternalDevice* is used to expand capabilities and they may be classified in four types:

- **Input.** Input devices are those that can be used to send data to *itiServer* or *itiClient* and include keyboards and mice, among other devices.
- **Output.** Output devices are those that can be used to receive data from the *itiServer* or *itiClient*. Examples are monitors and printers, among other devices.
- **Input and Output.** Devices that can be used to send and receive data such as touch screens.
- **Storage.** Devices that can store data and include, for instance, portable hard drives and flash drives.

3.4.3.9 itiLogicalPartition

An *itiLogicalPartition* is a functionality that allows the partition of storage resources in multiple logical units that will then be available to both *itiServer* and *itiClient*. In the **itiPM** it is assumed that all *itiServer* instances and *itiClient* instances have at least one *itiLogicalPartition*.

3.4.3.10 itiDisk

An *itiDisk* is where data is stored. It is part of an *itiServer* or *itiClient* and is a required component to allow the creation of an *itiLogicalPartition*. It is assumed that an *itiServer* may boot using local disks or it may boot directly from remote storage without requiring access to an *itiDisk*.

3.4.4 ITI Network Package

The *itiServer* and *itiClient* available in the **ITI Nodes Package** can be interconnected using *itiNIC* by *itiNetworkConnectableDevices* (Network Devices). The *itiNetworkConnectableDevices* include, among others, devices such as access points, firewalls, hubs, routers and switches, that communicate using specific protocols such as Frame Relay or Ethernet. Those devices are also used to create different *itiNetworkZone* (Network Zones) for security reasons such *Perimeter Networks*, *Intranets* or *Extranets* and they can create different types of *itiNetwork* (Networks) such as LAN, Wide Area Network (WAN) or Wi-Fi.

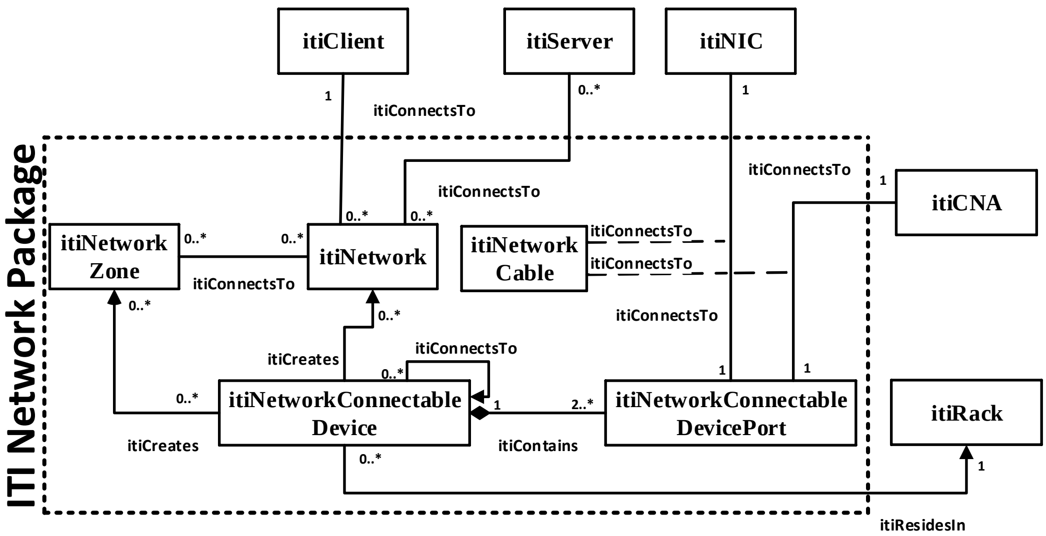


Figure 3.11: ITI network package (itiPM)

3.4.4.1 itiNetworkConnectableDevice

A network device (*itiNetworkConnectableDevice*) is one of the main components of a network (e.g. router, switch) responsible for creating multiple networks and zones within an ITI.

3.4.4.2 itiNetworkConnectableDevicePort

This metaclass represents the multiple ports available in network devices (*itiNetworkConnectableDevice*). Each port can receive a network cable (*itiNetworkCable*).

3.4.4.3 itiNetwork

An *itiNetwork* is a logical term and allows resources such as an *itiServer* or *itiClient* to exchange data through network devices (*itiNetworkConnectableDevice*) and physical cable or wireless media (*itiNetworkCable*).

3.4.4.4 itiNetworkZone

For security, segmentation, or other reasons, an *itiNetwork* can be divided in multiple network zones (*itiNetworkZone*). Examples of these zones are the *perimeter network* (frequently referred as *DeMilitarized Zone (DMZ)*) which is a physical or logical subnetwork that exposes ITI services to a larger untrusted network such as the *Internet*. Other popular zones are the *Intranet* or *Extranet*.

3.4.4.5 itiNetworkCable

The physical connection between networked computing devices is established using *itiNetworkCable* and is represented as an UML association class. Through an *itiNetworkCable* it is possible that network devices (e.g. server, printer or scanner) connect to other network devices. The devices can be separated by a few meters (e.g. via Ethernet) or arbitrarily large distances (e.g. Internet).

3.4.5 ITI Storage Package

The package *ITI Storage* contains devices that store data and may be configured in different storage models such as *Direct Attached Storage (DAS)*, *Storage Area Network (SAN)* or *Network Access Storage (NAS)*. These storage components are connected to servers trough storage networks that can use specific storage protocols such as *iSCSI*, *Fiber Channel (FC)* or *Fiber Channel over Ethernet (FCoE)*.

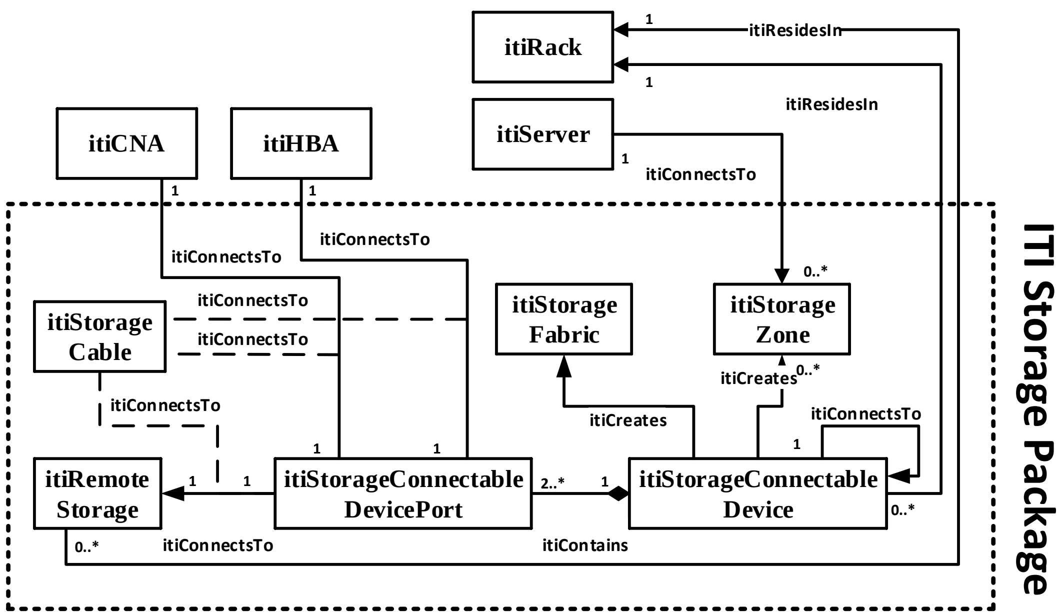


Figure 3.12: ITI storage package (itiPM)

3.4.5.1 itiStorageConnectableDevice

The storage network equipment *itiStorageConnectableDevice* is a physical component that is responsible for providing access to storage (*itiRemoteStorage*) through its ports (*itiStorageConnectableDevicePort*). Each *itiStorageConnectableDevice* contains one or more *itiStorageConnectableDevicePort*.

3.4.5.2 itiStorageFabric

A storage fabric (*itiStorageFabric*) consists in a configuration of an *itiStorageConnectableDevice* (e.g. switch) that defines which devices are connected to whom, to form a network.

3.4.5.3 itiStorageZone

The storage zoning (*itiStorageZone*) can be defined as a finer segmentation in a way that only the members of the same zone can communicate within that zone, with all the other attempts rejected. It is important to mention that a device (e.g. *itiServer*) can be in more than one zone.

3.4.5.4 itiRemoteStorage

The remote storage (*itiRemoteStorage*) is a place where data is stored and there are different technologies that can provide this service. The most well-known are *(i)* Direct Attached Storage (single disk drive or single tape drive) that is directly attached to single host (or multiple hosts in case of a cluster) through a media, *(ii)* the NAS which consists in storage that is remotely accessible through TCP/IP based network (LAN or WAN) and *(iii)* the storage area network which consists in a dedicated network that allow an *itiServer* to transparently access remote storage and can include disk arrays or tape libraries, among other devices.

3.4.5.5 itiStorageCable

itiStorageCable is the physical wiring or cable used to form a storage network that allows an *itiServer* to connect to storage and is represented as an association class. FC is the most commonly used network technology. One *itiStorageCable ConnectsTo* one *itiStorageConnectableDevicePort*.

3.4.6 ITI Software Package

This package contains the constructs used to represent IT infrastructure software. Its main metaclasses are the *itiHV* (hypervisor), *itiOS* (Operating System) and *itiSoftware* (Infrastructure Software).

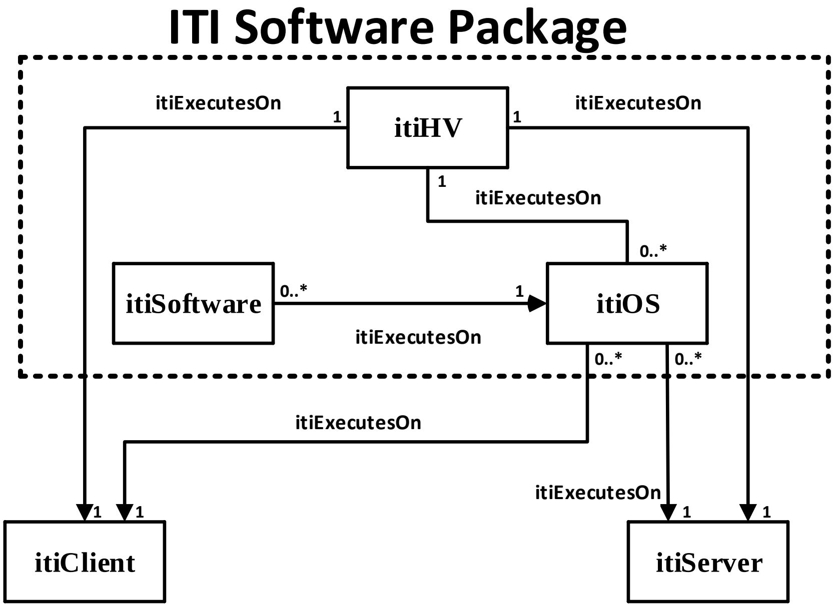


Figure 3.13: ITI software package (itiPM)

3.4.6.1 itiHV

An *itiHV* represents the software or firmware that allows the creation and use of virtual machines. The hypervisor software can be installed directly in bare-metal servers and provides a set of features to control hardware resources, such as processor power, memory, and I/O. The operating system may be deployed directly on hardware or it can be deployed on top of a hypervisor software such as XEN, HyperV or VMware. The hypervisor is also modeled as an execution environment (Node). It is frequent to call *Host* to the hypervisor and *guest* to the operating system running in a virtual machine.

It is technically possible to have multiple instances of hypervisors installed on the same *itiServer* or *itiClient*. However, because only one instance can run at a time and this is not a common or recommended solution, it is assumed that an *itiServer* or *itiClient* can run only one hypervisor instance.

3.4.6.2 itiOS

The *itiOS* is the software that manages hardware resources of an *itiClient* or *itiServer* and represents a fundamental component because applications usually require an operating system to work. The operating system is represented using the UML concept of Execution Environment (Node) and is nested inside an *itiServer* and *itiClient* from package *ITI Nodes*. Examples of ITI operating systems include Windows Phone, Windows Server, Red Hat Enterprise Linux, Solaris, HPUX, IBM AIX, MAC OS, Android, among others.

3.4.6.3 itiSoftware

The *itiSoftware* is represented as a UML Component and is the software that provides IT capabilities. These IT capabilities are, for instance, antivirus, application servers, backup servers, collaboration servers, database servers, directory, and email servers, among others.

3.4.7 ITI Location Package

The **ITI Location package** is used to represent the multiple physical locations where an infrastructure may reside. Examples of metaclasses that may reside in this package are the datacenter (*itiDatacenter*) which represents the place where organization's main infrastructure components, such as servers, networking and storage systems are located. Other locations include Headquarters, Branch Office, Regional Office among others.

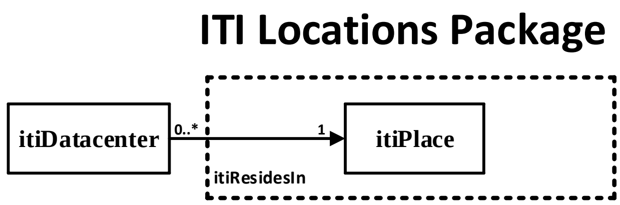


Figure 3.14: ITI location package (itiPM)

3.5 ITI Profile Methodological Support

Instead of creating another modeling tool from scratch, we decided to evaluate existing modeling tools that can be extended to incorporate our proposals. The evaluation was based on the list of features, last release, version of UML supported, usability, extendability, among other factors. The EA from the Australian company *Sparx Systems* was found to be the best available choice to incorporate most capabilities. Besides, it is considered one of the most widely used UML modeling tools currently used by several small and medium-sized enterprises [8].

Sparx EA is a modeling tool with support for most of the UML features defined in the UML 2.0 specification, allowing the modeler to:

- Create UML models and profiles.
- Structure elements in packages and diagrams.
- Defining connectors between elements
- Documenting those elements
- Generating code

[8] http://list.ly/list/2io-popular-uml-modeling-tools accessed on 2017/04/04

The main limitations were related with the implementation and validation of OCL expressions in the model which lead us to use the UML-based Specification Environment (USE) tool in a complementary, fashion, as will be explained later.

3.5.1 UML Profile for ITIs

As mentioned before, we used the extensions capabilities of UML to create the ITI Profile (itiPM). The profile is a collection of additional stereotypes and tagged values, among other properties, applied to elements, attributes, methods and connectors, which together allow the design of models in the domain of ITIs. The increased precision facilitated by the use of a formal metamodel specified in itiPM, is rendered possible by specifying rules and constraints upon it using OCL clauses. The profile is specified in XML file format. Once created, the XML files can be imported into Sparx EA and the modeler can use drag and drop functionalities to use ITI elements and create ITI diagrams. The following sections detail the extensions mechanisms used.

3.5.1.1 ITI Profile packages

Mainly for organizational purposes, we used the packages to organize the elements of itiPM and to control the itiPM version. The packages worked much like folders in an operating system and they can contain sub-packages diagrams and elements.

The first step to create the ITI profile consists in creating the profile package. To be able to create a new package, sparx EA requires the existence of a project file and the package requires the stereotype *«profile»*, the diagram group has to be *UML Structural* and diagram type should be *Class*. The package structure of the profile is as presented in figure 3.15.

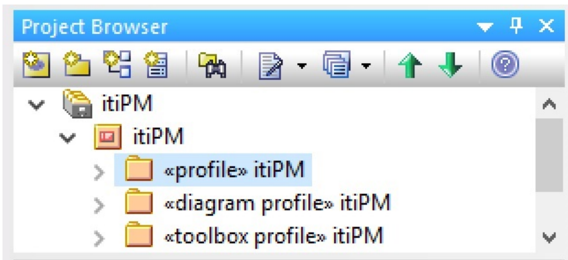


Figure 3.15: itiPM structure

3.5.1.2 ITI Profile toolboxes

The diagram toolboxes are an essential feature that we wanted to include in the ITI profile to simplify their usage since they include and give access to all ITI elements and connectors. Each toolbox can include hidden sub-menus and icons for items. To create toolbox pages, it is required to create an extension connector for diagrams and then between each stereotype element and the *ToolboxPage* metaclass element, as can be seen in figure 3.16.

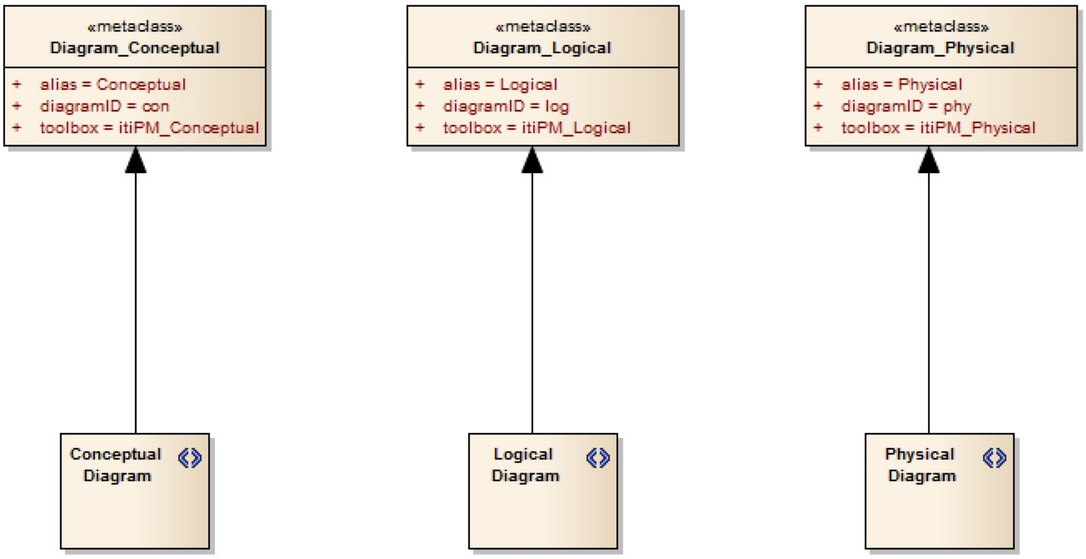


Figure 3.16: Profile toolboxes

In our profile, we decided to create the following types of ITI diagrams:

- **Conceptual.** A conceptual IT Infrastructure is used to map out areas of technology into a structure and framework. This is used to define, name, and position these areas for a common understanding between the IT supplier and the organizations using the technology and to ensure that all the technology areas required to implement an organization's operational or nonfunctional requirements are defined and available to the organization.
- **Logical.** Logical infrastructure diagram is used to create a model that describes the servers, network, storage communication paths in a way that is easily shared among system architects, infrastructure architects, developers, and IT operations staff.
- **Physical.** The physical diagram presents details of physical communication links, such as cable length, grade, and approximation of the physical paths of the wiring, analog, and Integrated Services Digital Network (ISDN) lines. It represents also servers, with computer name, IP addresses, server role, domain membership, location of devices such as printers, hubs, switches, modems, routers and bridges, and proxy servers that are on the network.

For each stereotype of type *ToolboxPage* we have defined the name (e.g. *Network Devices*) and included all the related stereotypes. This feature was important not only to organize elements and connectors, but also to decrease the effort of designing ITIs.

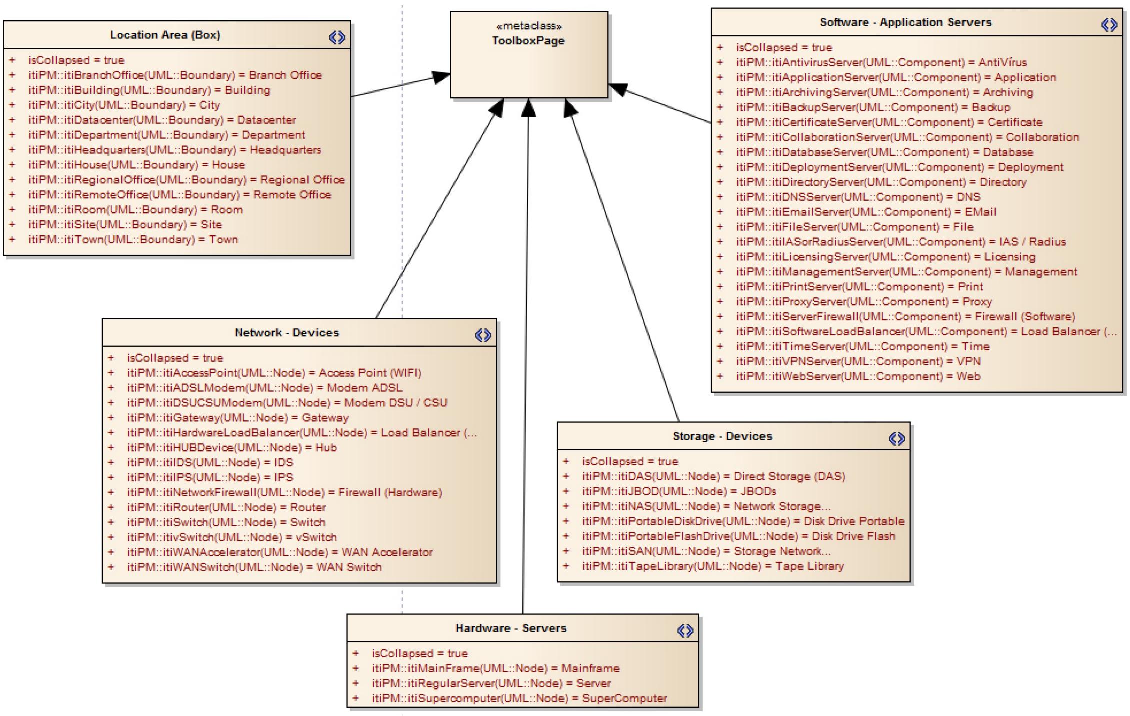


Figure 3.17: Subset of itiPM physical toolbox

3.5.1.3 ITI Profile stereotypes and metaclasses

Stereotypes are an extension mechanism that can be used to change the meaning, display or syntax of a model element. The stereotypes can be applied to elements (such as classes and objects), attributes and operations and connectors such as associations and generalizations. The stereotypes can either extend **Core UML objects (Metaclasses)** or extend **non-UML objects (Stereotypes)**. The stereotypes can extend metaclasses in several ways:

- One stereotype extending one metaclass.
- One Stereotype extending more than one metaclass.
- Several Stereotypes extending one metaclass.

For understandability reasons, the stereotypes can also be associated with new shapes to change their appearance, using image files or shape scripts.

3.5.1.4 ITI Profile Tagged values

To enrich the profile, it was required to define additional meta-information for each ITI stereotype element, by adding various types of tagged values, which will represent attributes of each element. The tagged values can be assigned to a range of model objects, including elements, object instances, connectors, attributes and operations.

It is possible to customize existing tagged values or create new tagged values based on predefined or system-provided types. The tagged values can be:

- Complex and based on predefined types.
- Composite containing other tagged values.

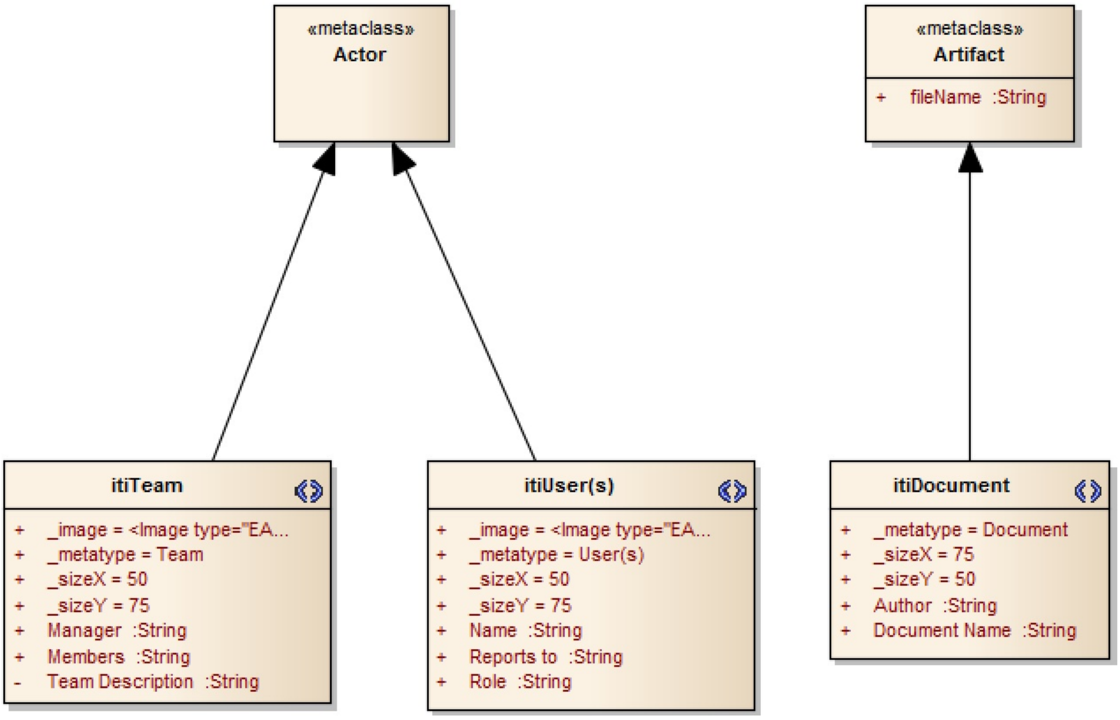


Figure 3.18: Extending metaclasses

- Used to return values from the various reference data tables.
- Masked to insert user-provided data into a text string.

Most of the ITI stereotypes were created with enumeration elements, such as the ones presented in figure 3.19, to make the process of designing ITIs faster, since a drop-down list is presented for each attribute of the ITI element and also has the benefit of restricting possible values, avoiding mistakes. When the element is dragged to the diagram, all the associated tagged values are added to the element as well.

Another important feature of tagged values, is their possibility of inheriting from parent classes. We have defined tagged values for most of ITI elements and we change their appearance based on the specified values.

3.5.1.5 ITI Profile Notation and Shape Scripts

Instead of using the standard UML notation in the ITI profile, we decided to use a Sparx EA scripting language to specify custom shapes. We have associated a different shape script to each stereotype, so that each element can be easily identified. For simplicity, we have defined the following main rules for the elements of the ITI profile:

- **Blue.** Applies by default to all components of type device, such as servers.
- **Amber.** Applies by default to all components of type network.
- **Green.** Applies by default to all components of type storage, such as SANs.
- **Yellow.** Applies by default to all components of type software.

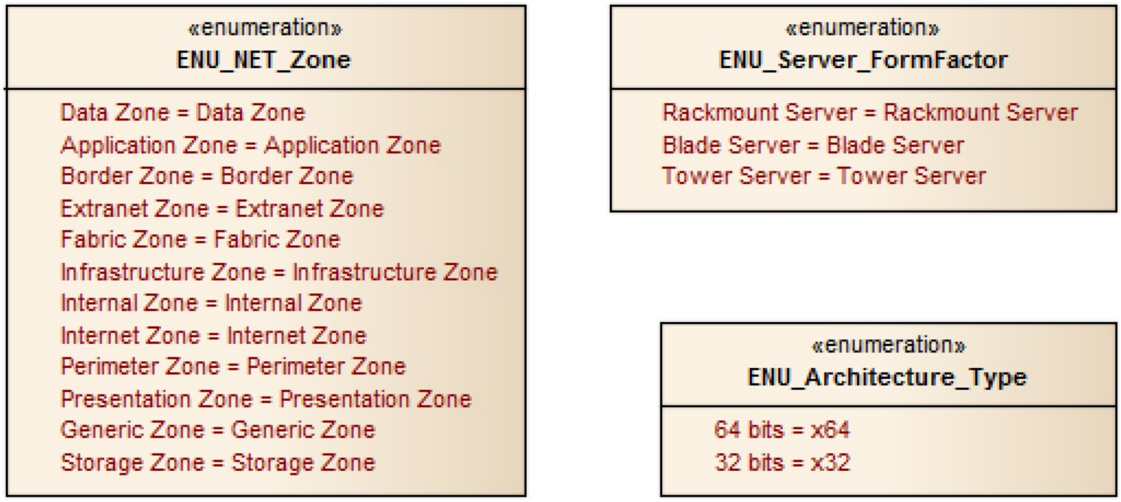


Figure 3.19: Enum tagged values

For connections, we used the script provided in figure 3.20, to define the line type, connection name and the line color.

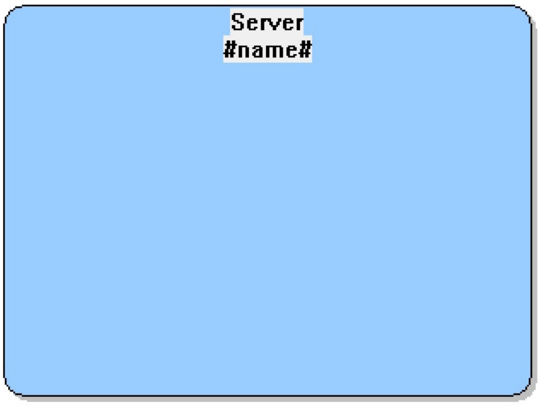

```
shape main
{
noshadow = "true";
h_align = "center";
editablefield = "name";
setpencolor(34,177,66);
setpenwidth(1);
print("#name#");
lineto(100,0);
}
```

Figure 3.20: Notation for connections

The shape script associated for nodes, such as servers, is displayed in figure 3.21. For all the other shapes, according to the rules above, only the *setfillcolor* and *name* properties change.

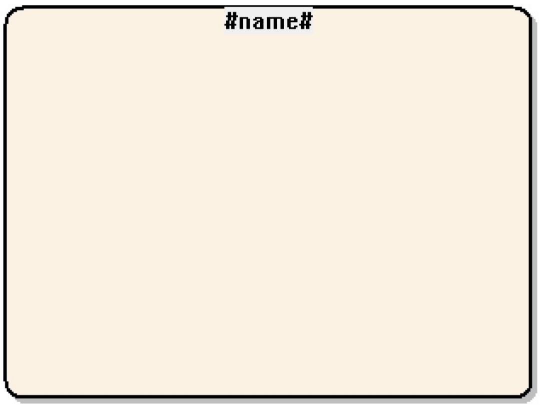
There are cases where we need to change the element color based on the information provided in the tagged values. An example is the classification of the infrastructure network zones (e.g. Perimeter Network should be red, while all the other zones green). The script available in figure 3.22 implements this functionality.

We used a similar approach for network segments. We wanted to define a set of properties based on the network segment type, selected in the tagged value. We used the method *HasTag* which returns true if the associated element has a tag value with the name *tagname*. The value of the tag has to be equal to *tagvalue* for the method to return *true*. A subset of the script implemented in the ITI profile is available in listing 3.1.




```
shape main
{
noshadow = "false";
h_align = "center";
editablefield = "name";
setfillcolor(153,204,255);
roundrect(0,0,100,100,25,25);
println("Server");
println("#name#");
setfillcolor("White");
}
```

Figure 3.21: Notation for shapes




```
shape main{
    SetPenWidth(2);
    RoundRect(0, 0, 100, 100, 20, 20);
    addsubshape("namecompartment", 100, 10);
    shape stereotypecompartment
    {
        h_align = "center";
        editablefield = "stereotype";
        println("«#stereotype#»");
    }

    shape namecompartment
    {
        h_align = "center";
        editablefield = "name";
        println("#name#");
    }
}
```

Figure 3.22: Notation based on user input

Listing 3.1: Network segments shape scripts

```
shape main{
  layoutType=border;
  h_align = center;
  rectangle(0,0,100,100);

  if(hasTag(Segment_Type, Public Network))
  {
    setfillcolor(238,50,17);
    rectangle(0,0,100,100);
    println(Public Network);
    return;
  }

  if(hasTag(Segment_Type, Private Network))
  {
    setfillcolor(255,128,0);
```

```
    rectangle(0,0,100,100);
    println(Private Network);
    return;
  }

  if(hasTag(Segment_Type, Internal Network))
  {
    setfillcolor(209,253,130);
    rectangle(0,0,100,100);
    println(Internal Network);
    return;
  }
  if(hasTag(Segment_Type, External Network))
  {
    setfillcolor(64,128,128);
    rectangle(0,0,100,100);
    println(External Network);
    return;
  }
  ...
```

3.5.1.6 Quick Linker

To decrease the effort in designing ITIs, we used the *Quick Linker* functionality which is a simple and fast way to create new elements and connectors on a diagram and their relationships. When an element is selected in a diagram, the *Quick Linker* icon is displayed in the upper right corner of an element. Simply clicking and dragging the icon enables the creation of new connectors and elements. For each element, we have provided a list of valid connections.

3.5.1.7 MDG Technologies

The Model Driven Generation (MDG) Technologies is a core functionality of Sparx EA, allowing the ITI profile distribution and any EA user to import and access their resources. The MDG Technologies can include UML Profiles, code modules, patterns, images, tagged value types, Rich Text Format (RTF) report templates, linked document templates, toolbox pages and task pane pages. We included those resources of the itiPM implementation/deployment as an MDG Technology, as can be seen in figure 3.23.

3.5.1.8 ITI Profile Creation - Step by Step

The process of creating an UML Profile for ITIs on Sparx EA, comprise a number of steps as follows:

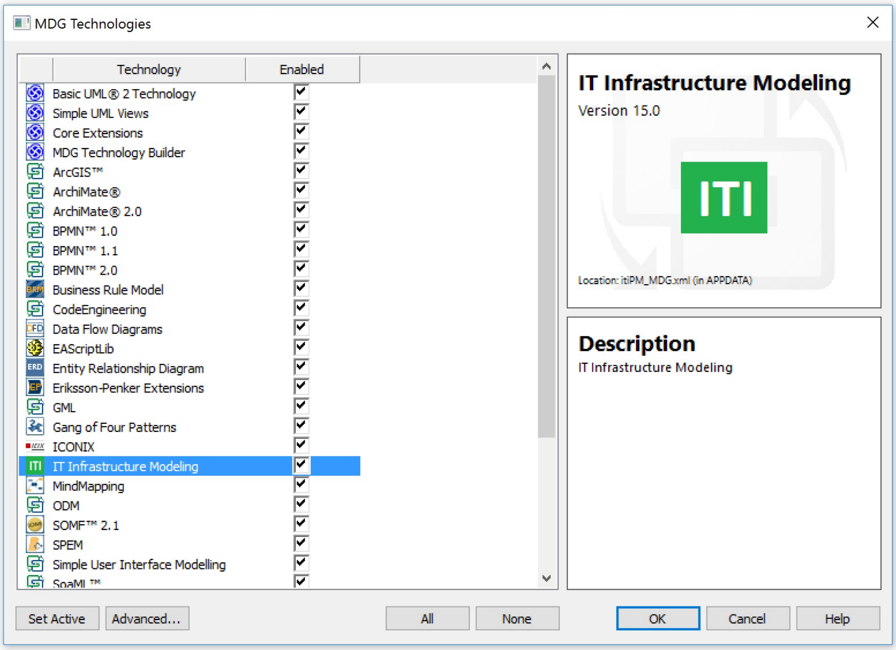


Figure 3.23: MDG technologies for ITI modeling

1. Creation of a **profile package** that has the stereotype «profile» in a technology development model.

2. Add **stereotype and metaclass** elements to the child diagrams of the profile package.

3. Define additional meta-information for a stereotype by adding various types of **tagged values**.

4. Define drop-down list of values for a tagged value on the stereotype element. This can be done with the creation of an **enumeration element**.

5. Define appearance for each UML element and connectors in terms of shape, font, color and labeling. This can be performed with **shape scripts** for stereotype elements.

6. Provide **Quick Linker** definitions in the Profile to allow suggesting elements that can be connected to a specific element. This is done using a Comma Separated Values (CSV) file and incorporated later into the Profile.

7. **Save package** or the diagrams as Profile, and export it to disk. The Profile can then be integrated with an MDG Technology and deployed to other models for use.

8. Incorporate the Profile into an **MDG Technology** and deploy the technology file.

3.5.2 itiPM Semantics Enforcement

Currently there are several tools available from both commercial companies and universities that support OCL. Some examples are ModelRun from Boldsoft, OCL compiler from University of Dresden and OCL compiler from Cybernetic Intelligence GmbH, Octopus from Klasse Objecten, USE from University of Bremen and many others. Among the several existing UML tools that can be used to achieve our goals in terms of expressing and detecting any violations of ITI best practices, we decided to work with the *USE* [Gog+02]. This decision was based upon the fact that it fulfills the requirements for evaluating ITIs and we already had knowledge and good experiences in using it in other research projects [Sil08; SA10c]. The main drawback is the current lack of integration between *Sparx EA* and *USE*, which forced us to perform a manual export and import process.

The USE tool was developed by Mark Richters at the University of Bremen with the purpose of creating information systems specifications. A specification in USE consists of textual descriptions of the modeling elements in an UML model and additional integrity constraints specified in OCL [Gog+07]. The OCL language is currently a standard and was adopted as a formal specification language within UML to define constraints or validation rules that we can apply to our *itiPM* metamodel. OCL expressions can be used in for several purposes such as the specification of the initial value of an attribute or association, the specification of the body of an operation, indication of a guard condition and, most of all, to specify the following types of constraints on the itiPM metamodel:

- **Invariants**. An invariant is a constraint that states a condition that must always be met by all instances of the class, type, or interface. An invariant is described using an expression that evaluates to true if the invariant is met. Invariants must be true at all times;
- **Preconditions**. A precondition to an operation is a restriction that must be true before the operation is called. Fulfilling a precondition is an obligation for the operation caller;
- **Postconditions**. A postcondition to an operation is a restriction that must be true when the operation ends its execution. Having a postcondition fulfilled is a right for the operation caller.

We have the ITI metamodel expressed as a UML class diagram, upon which well-formedness rules and best practices are defined using OCL invariants. With USE we were able to load the itiPM metamodel and the *itiLib* library (which will be detailed later) which contains meta-classes, meta-associations, meta-attributes, meta-operations and constraints. Once this metamodel has been loaded, we can add the meta-objects and initialize their attributes. Each time a change in the system's state occurs, an automatic checking of the invariants is carried out. For example, when a new object is introduced, it is checked that it fulfills the specification of the itiPM model and all the constraints defined.

3.5.2.1 itiPM metamodel consistency

To ensure the consistency of our **itiPM**, we enforce some semantics in the metamodel through the definition of Well-Formedness Rule (WFR), which are expressed as OCL invariants. As mentioned, the latter are OCL expressions that must be true for all instances of that type at any time. The number and type of WFRs to create depends on the domain. Based on this formalization, we were able to perform the detection of best practices violation, as will be described later in this chapter.

To give an example, the OCL expression available on listing 3.2 defines that all existing locations in a particular ITI must contain at least an object of type server. This is considered a best practice and is applied to ITI directory services to validate that there are no empty sites in the ITI.

Listing 3.2: Each site contains at least one server

```
context itiDirectoryServer
inv SiteHasAtLeastOneServer:
  IsSite() implies itiContain.allInstances ->
  select(ServerMember.oclIsTypeOf(itiDirectoryServer))->
  exists(SiteMember=self and
  ServerMember.oclAsType(itiDirectoryServer).IsServer())
```

Usually a directory service is a core component of an ITI, since it consists in a shared repository which includes most ITI components available in the network, such as devices, servers, sites (aka locations), printers, volumes, users, groups, phone numbers and e-mail addresses, among many other information. A directory service is a critical component of an ITI for locating, managing, administering and organizing network resources and this service is provided by directory servers that consider each network resource on the network as an object. The information about a particular object is stored as a collection of attributes associated with that object.

3.5.2.2 Concrete syntax examples

In the domain of ITIs, a member cannot be multiple contained (e.g. a server cannot belong simultaneously to the site Lisbon and to the site Paris.). The expression isUnique specifies whether the return parameter is unique or not. The OCL expression to enforce this semantic is shown in listing 3.3.

Listing 3.3: A member is unique within a container

```
context itiContain
  inv noMultipleContainment:
    itiContain.allInstances->isUnique(ServerMember)
```

Another example is enforcing that servers and clients, or some parents and members, cannot be null (e.g. a server has to belong to a specific location) as specified in listings 3.4 and 3.5.

Listing 3.4: The Parent and Member definitions cannot be null

```
context itiContain
  inv noNullParentOrMember:
    not(SiteMember.isUndefined() or ServerMember.isUndefined())
```

Listing 3.5: The server and client definitions cannot be null

```
context itiConnect
  inv noNullServerOrClient:
    not(DestinationReplication.isUndefined() or
    SourceReplication.isUndefined())
```

Another example of an invariant for semantic enforcement regards the membership of objects. Listing 3.6 specifies that ITI servers must be contained by a site and listing 3.7 specifies that a site cannot have a container. Using a similar approach, also a subnet must be contained by a site. The subnet is a property of a site and we want to ensure that all subnets are allocated to sites. The OCL expression for subnets is available in listing 3.8.

Listing 3.6: A server must be contained by a site

```
context itiDirectoryServer
  inv ServerContainedBySite:
    IsServer() implies Container().IsSite()
```

Listing 3.7: A site cannot have a container

```
context itiDirectoryServer
  inv SiteHasNoContainer:
    IsSite() implies Container().isUndefined()
```

Listing 3.8: A subnet must be contained by a site

```
context itiSiteSubnet
  inv SubnetContainedBySite:
    IsSubnet() implies Container().IsSite()
```

When defining an ITI with distributed components linked via communication links, it is important to avoid errors such as having a specific server or site communicating with itself. The OCL expression to ensure that there are no communications where the source and destination are the same is shown in listing 3.9.

In a communication, we want also to ensure that this communication takes part between two servers and not between one server and one site or between one subnet and one server. This is accomplished using the OCL expression in listing 3.10.

Listing 3.9: Neither a site nor a server can be linked to itself

```
context itiDirectoryServer
  inv NotMutualConnection:
    not itiConnect.allInstances->exists(Client()=self and Server()=self)
```

Listing 3.10: A Connection must be established between two servers

```
context itiConnect
  inv ConnectionBetweenServers:
    IsNTDSConnection() implies Server().IsServer() and Client().IsServer
        ()
```

From a site perspective, we want to ensure that the source and destination are objects of the type site, to avoid having sites communicating with servers and vice versa (listing 3.11) and that all connections are defined to ensure, for instance, that in a connection between a server and a client only one component is connected (listing 3.12).

Listing 3.11: A sitelink must be established between two sites

```
context itiConnect
  inv ConnectionBetweenSites:
    IsSiteLink() implies Server().IsSite() and Client().IsSite()
```

Listing 3.12: An Endpoint must have a container

```
context EndpointDefinition
  inv noNullContainer:
    not(ContainmentDef().isUndefined())
```

3.5.2.3 Best practices enforcement

This section presents best practices formally defined using OCL. Each best practice will be defined as an invariant in the context of the **itiPM**.

"A site in an ITI should have at least one server". The invariant *SiteHasAtLeastOneServer* (Listing 3.13) says that if a site exists, it should have at least a member of the type server.

Listing 3.13: A site should have at least one server

```
context itiDirectoryServer
  inv SiteHasAtLeastOneServer:
    IsSite() implies itiContain.allInstances->
    select(ServerMember.oclIsTypeOf(itiDirectoryServer))->
    exists(SiteMember=self and
    ServerMember.oclAsType(itiDirectoryServer).IsServer())
```

"A site in an ITI should have at least one subnet defined". The invariant *SiteHasAtLeastOneSubnet* (Listing 3.14) says that if a site exists, it should have at least a member of the type subnet.

Listing 3.14: A site should have at least one subnet

```
context itiDirectoryServer
  inv SiteHasAtLeastOneSubnet:
    IsSite() implies itiContain.allInstances->
    select(ServerMember.oclIsTypeOf(itiSiteSubnet))
    ->exists(SiteMember=self and
    ServerMember.oclAsType(itiSiteSubnet).IsSubnet())
```

"When an organization has more than one site in the same ITI, they should be interconnected". The invariant *SiteLinkedToAtLeastAnotherSite* (Listing 3.15) says that if more than one site exists, they should be connected.

Listing 3.15: A site should be linked to at least another site

```
context itiDirectoryServer
  inv SiteLinkedToAtLeastAnotherSite:
    IsSite() implies itiConnect.allInstances->
    exists(Client()=self and Server().IsSite())
```

"To allow information interchange, servers in the same ITI should be interconnected". The invariant *ServerConnectedToAtLeastAnotherServer* (Listing 3.16) says that when more than one site exists, the servers have to be connected to other servers.

Listing 3.16: A server should be connected to at least another server

```
context itiDirectoryServer
  inv ServerConnectedToAtLeastAnotherServer:
    IsServer() implies itiConnect.allInstances->
    exists(Client()=self and Server().IsServer()) and
    itiConnect.allInstances->exists(Server()=self and Client().IsServer
        ())
```

In the following section we will apply each of these invariants to a real example to check if there are best practices violations. An invariant can be evaluated to true, false or not applicable (n/a). A falsified invariant indicates an invalid system state which can be inspected by double-clicking the invariant name in the USE tool.

3.5.2.4 Detecting best practices violations

To demonstrate the applicability of OCL expressions in the context of designing ITIs, we will evaluate an entire ITI to detect if there are some problems. We defined several OCL expressions, that according to our experience, consisted in best practices for ITIs. One of these best practices is related with having all infrastructure connected and available. So, if for instance, we have part of an ITI not connected, according to our approach this is a violation of a best practice.

To demonstrate the feasibility of our approach, we will load the infrastructure of figure 3.24 and we will use the USE tool, to detect if there are best practices violations. This is a real example, and each box represents a *site* of the ITI with information such as the location, servers and their names, subnets, IP addresses and domains, among other ITI details and each line represents the connections among *sites*. The diagram is presented in a small size and blurred fashion to guarantee the confidentiality of the data origin. With a different color, there is one isolated site in the figure 3.24 that is not connected to the rest of the ITI that will be called Site_Lisbon.

The result of checking the application of the best practices against this infrastructure gives the following results.

Listing 3.17: Check best practices violations in ITI designs

```
use> check
checking structure...
checking invariants (using 5 concurrent threads)...
checking invariant (1)
itiDirectoryServer::ServerConnectedToAtLeastAnotherServer:OK.
checking invariant (2)
itiDirectoryServer::SiteHasAtLeastOneServer:OK.
checking invariant (3)
itiDirectoryServer::SiteHasAtLeastOneSubnet:OK.
checking invariant (4)
itiDirectoryServer::SiteLinkedToAtLeastAnotherSite:FAILED.
-> false : Boolean
Instances of itiDirectoryServer violating the invariant:
-> Set(@Site_Lisbon) : Set(itiDirectoryServer)
checked 4 invariants in 0.080s, 1 failure.
use>
```

With the exception of invariant *SiteLinkedToAtLeastAnotherSite* all are OK, meaning that most of the best practices are in place in this organization. The **FAILED** invariant

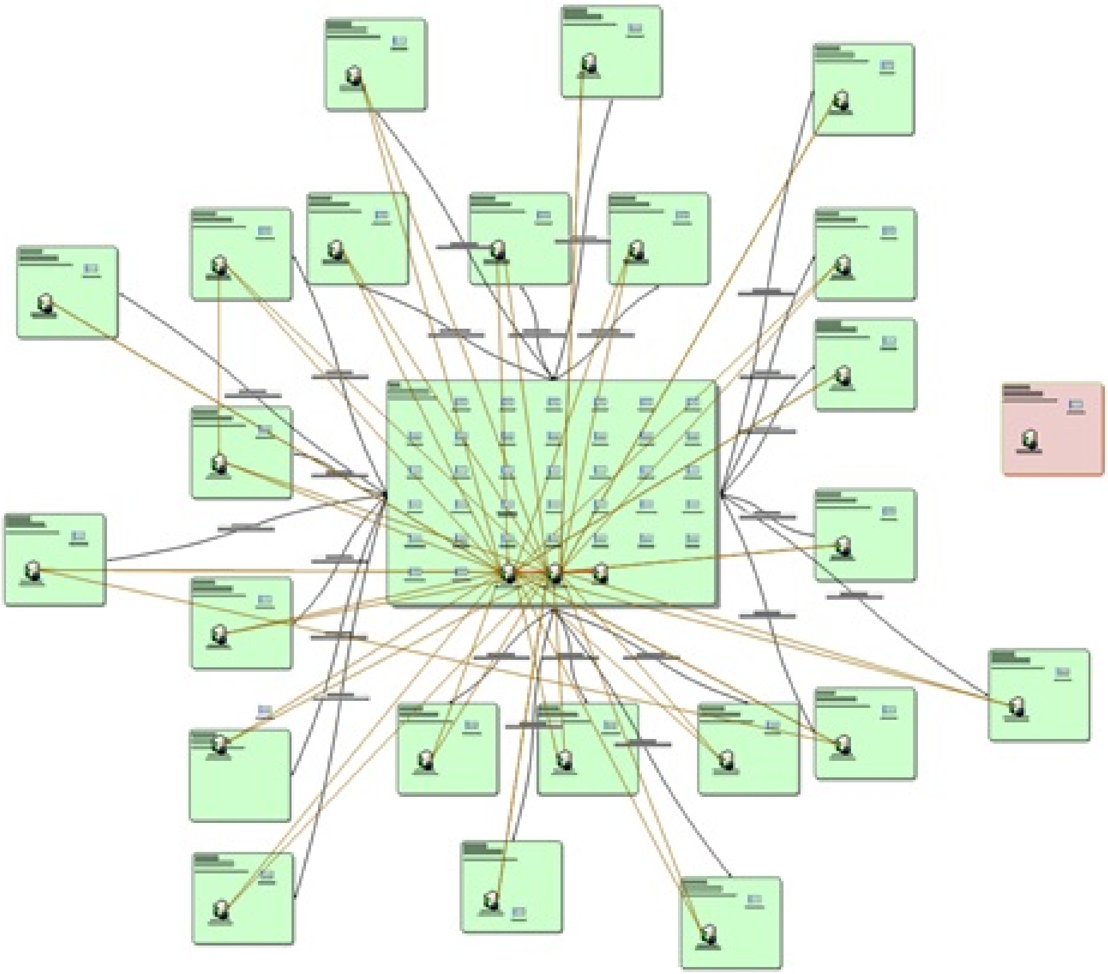


Figure 3.24: Best practices violation (blurred for confidentiality reasons)

(line 11) is due to the fact that there is an isolated site in the infrastructure. There are several reasons that can lead to problems like this. In our experience, the most common problems are anomalies or human errors.

3.5.3 itiPM Library for ITIs

We defined several operations based on OCL expressions to collect relevant information from the meta-data and support the evaluation of ITIs and we grouped them in a library called *itiPM Library for ITIs (itiLib)*. The *itiLib* is an extension of the *itiPM* metamodel.

The *itiLib* contains several metrics that can be used not only to evaluate best practices (as described earlier) but also to evaluate ITIs using different perspectives such as their sizing or their level of complexity. We followed best practices defined by [AC94] that indicate that metrics should be:

- Formally defined;
- System size independent (non-size metrics);
- Dimensionless or expressed in some consistent unit system;

- Obtainable early in the life cycle;
- Down scalable;
- Easily computable;
- Language independent.

It is also important to mention that in the development of these metrics we tried to create them with the following characteristics: [PW95]:

- **Simple:** definition and use of the metric is simple;
- **Objective:** different people will obtain identical values from the same entities being evaluated, guaranteeing consistency and prevent individual bias;
- **Easily collected:** the cost and effort to obtain the metric is reasonable;
- **Robust:** the metric is insensitive to irrelevant changes; allows for useful comparison;
- **Valid:** the metric measures what it is supposed to; this promotes trustworthiness of the metric.

There are different types of operations defined in the *itiLib*. The operations are organized in the following groups:

- Generic operations
- Operations for sites
- Operations for servers
- Operations for sites and servers
- Operations for subnets
- Operations for siteLinks
- Operations for connections

3.5.3.1 Generic itiLib operations

Some of the operations available in *itiLib* were classified as generic and allow, for instance, the quantitative evaluation of ITIs. Listing 3.18 presents an example of generic operations, which are used to support other operations. Since they are Boolean, when invoked these operations return true or false, depending on the condition. The attribute *objectClass* was defined by us in the *itiPM* schema for simplicity reasons.

Listing 3.18: objectClass attribute in itiLib operations

```
IsSite(): Boolean = objectClass = site
IsServer(): Boolean = objectClass = server
IsSubnet(): Boolean = objectClass = subnet
IsSiteLink(): Boolean = objectClass = siteLink
IsNTDSConnection(): Boolean = objectClass = nTDSConnection
```

The *CommunicationDefinition* relationship is used to represent communication links for sites and servers. Both send and receive data trough *endpoints*. *IsInput()* and *IsOutput()*

are boolean operations created with the purpose of identifying if a given connection is used to receive information or to send information.

Listing 3.19: *IsInput()* and *IsOutput()* itiLib operation

```
IsInput(): Boolean = itiConnect.allInstances->exists(
    DestinationReplication = self)
IsOutput(): Boolean = itiConnect.allInstances->exists(
    SourceReplication = self)
```

The *ContainmentDef()* operation is defined in class *itiDirectoryServer* and is used to identify all containment relationships for a specified object. This operation relies on the operation *first()* that, as the name implies, returns the first element. The operation *Container()*, uses the *ContainmentDef()* itiLib operation to identify all container relationships for a specific *itiDirectoryServer* object. As the name suggests, the *SourceReplication* and *DestinationReplication* attributes are used to identify the source and destination of the information.

Listing 3.20: *ContainmentDef()* and *Container()* itiLib operations

```
ContainmentDef(): itiContain = itiContain.allInstances->select(
    ServerMember = self)->asSequence->first()
Container(): itiDirectoryServer = ContainmentDef().SiteMember.
    oclAsType(itiDirectoryServer)
```

To allow the analysis of the ITI objects, we defined a set of operations that can be used to work with dates. The class Date defined in figure 3.21 has four operations, *isBefore()*, *isAfter()*, *isEqual()* and *isBetween()*.

Listing 3.21: Date class

```
class Date
  attributes
    day:Integer
    month:Integer
    year:Integer

  operations
    isBefore(t:Date): Boolean =
      if self.year = t.year then
        if self.month = t.month then
          self.day < t.day
        else
          self.month < t.month
        endif
      else
        self.year < t.year
      endif
```

```

    isAfter(t:Date): Boolean =
      if self.year = t.year then
        if self.month = t.month then
          self.day > t.day
        else
          self.month > t.month
        endif
      else
        self.year > t.year
      endif

    isEqual(t:Date): Boolean =
      self.year  = t.year and
      self.month = t.month and
      self.day   = t.day

    isBetween(startDate: Date, endDate: Date): Boolean =
      self.isEqual(startDate) or
      self.isEqual(endDate) or
      self.isAfter(startDate) and self.isBefore(endDate)

    yearsSince(t:Date): Integer =
      if self.month < t.month or
        self.month = t.month and self.day < t.day then
        self.year - t.year -1
      else
        self.year - t.year
      endif
end --Date
```

The operation *TotalAddedSitesInPeriod()* receives two dates as input and returns the number of sites added to the ITI between these two dates. Notice that this operation relies on the operation *isBetween()* defined earlier in class Date.

Listing 3.22: Total sites added and changed in periods of time

```
TotalAddedSitesInPeriod(firstDate: Date, lastDate: Date): Integer =
    Sites()->select(CreationDate.isBetween(firstDate, lastDate))->size
TotalAddedSitesNameInPeriod(firstDate: Date, lastDate: Date): Bag(String
    ) = Sites()->select(CreationDate.isBetween(firstDate, lastDate)).
    Name
TotalChangedSitesInPeriod(firstDate: Date, lastDate: Date): Integer =
    Sites()->select(ChangedDate.isBetween(firstDate, lastDate))->size
TotalChangedSitesNameInPeriod(firstDate:Date,lastDate:Date):Bag(String)=
     Sites()->select(ChangedDate.isBetween(firstDate, lastDate)).Name
```

The operation *TotalAddedSitesNameInPeriod()* is similar to the previous operation, however, instead of returning the number of sites added, returns the name of the sites added to the ITI in the specified dates. Instead of knowing the number or the name of the sites added to the ITI in a specified period, it may be required to identify the number of changes in sites or the name of the sites changed in a specified period. The following operations *TotalChangedSitesInPeriod()* and *TotalChangedSitesNameInPeriod()*, allow us to do this.

The same requirements may be required for servers. It may be interesting to know the number or the names of servers added or changed in a specified period of time.

Listing 3.23: Total servers added and changed in periods of time

```
TotalAddedServersInPeriod(firstDate: Date, lastDate: Date): Integer =
    Servers()->select(CreationDate.isBetween(firstDate, lastDate))->size
TotalAddedServersNameInPeriod(firstDate:Date,lastDate:Date):Bag(String)=
     Servers()->select(CreationDate.isBetween(firstDate, lastDate)).Name
TotalChangedServersInPeriod(firstDate: Date, lastDate: Date): Integer =
    Servers()->select(ChangedDate.isBetween(firstDate, lastDate))->size
TotalChangedServersNameInPeriod(firstDate:Date, lastDate:Date):Bag(
    String)= Servers()->select(ChangedDate.isBetween(firstDate, lastDate
    )).Name
```

As for sites and servers, the same applies to subnets. It may be interesting to know the number or the names of subnets added or changed in a specified period of time.

Listing 3.24: Total subnets added and changed in periods of time

```
TotalAddedSubnetsInPeriod(firstDate: Date, lastDate: Date): Integer =
    Subnets()->select(CreationDate.isBetween(firstDate, lastDate))->size
TotalAddedSubnetsNameInPeriod(firstDate:Date,lastDate:Date):Bag(String)=
     Subnets()->select(CreationDate.isBetween(firstDate, lastDate)).Name
TotalChangedSubnetsInPeriod(firstDate: Date, lastDate: Date): Integer =
    Subnets()->select(ChangedDate.isBetween(firstDate, lastDate))->size
TotalChangedSubnetsNameInPeriod(firstDate:Date,lastDate:Date):Bag(String
    )= Subnets()->select(ChangedDate.isBetween(firstDate, lastDate)).
    Name
```

To understand what connections were changed or created in a specified period of time, the same principles were applied to connections among servers.

Listing 3.25: Total siteLinks added and changed in periods of time

```
TotalAddedSitelinksInPeriod(firstDate: Date, lastDate: Date): Integer =
    Sitelinks()->select(CreationDate.isBetween(firstDate, lastDate))->
    size
TotalAddedSitelinksNameInPeriod(firstDate: Date, lastDate: Date): Bag(
    String) = Sitelinks()->select(CreationDate.isBetween(firstDate,
    lastDate)).Name
```

```
TotalChangedSitelinksInPeriod(firstDate: Date, lastDate: Date): Integer
    = Sitelinks()->select(ChangedDate.isBetween(firstDate, lastDate))->
    size
TotalChangedSitelinksNameInPeriod(firstDate: Date, lastDate: Date): Bag(
    String) = Sitelinks()->select(ChangedDate.isBetween(firstDate,
    lastDate)).Name
```

To identify the connections added or changed in the ITI in a specified period of time, we have created the following operations.

Listing 3.26: Total connections added and changed in periods of time

```
TotalAddedConnectionsInPeriod(firstDate:Date,lastDate:Date): Integer=
    Connections()->select(CreationDate.isBetween(firstDate, lastDate))->
    size
TotalAddedConnectionsNameInPeriod(firstDate: Date, lastDate: Date): Bag(
    String) = Connections()->select(CreationDate.isBetween(firstDate,
    lastDate)).Name
TotalChangedConnectionsInPeriod(firstDate:Date,lastDate:Date):Integer=
    Connections()->select(ChangedDate.isBetween(firstDate, lastDate))->
    size
TotalChangedConnectionsNameInPeriod(firstDate: Date, lastDate: Date):
    Bag(String) = Connections()->select(ChangedDate.isBetween(firstDate,
     lastDate)).Name
```

Listing 3.27: Operations used to work with paths

```

nextServerPath(): Set (itiDirectoryServer) =
  self.ConnectedServers().ConnectedServers()->asSet-> excluding(self)
longestPath(): Set(itiDirectoryServer) =
  self.comparePaths(oclEmpty(Set(itiDirectoryServer)), oclEmpty(Set(
      itiDirectoryServer))) -> excluding(self)
comparePaths(lgPath: Set(itiDirectoryServer),  path: Set(
    itiDirectoryServer)): Set(itiDirectoryServer) =
  if self.nextServerPath()->isEmpty() then
    if path->size+1 > lgPath->size then
      path->including(self)
    else
      lgPath
    endif
  else
    self.nextServerPath()->iterate(elem: itiDirectoryServer; acc:  Set(
        itiDirectoryServer) = path->including(self) |
    if acc->excludes(elem) then
      elem.comparePaths(lgPath, acc)
    else
      if acc->size > lgPath->size then
```

```
        acc
      else
        lgPath
      endif
    endif)
  endif
```

3.5.3.2 Manage Sites

We have defined in itiLib a set of operations to quantitative evaluate different aspects of ITI objects of *'Site'* kind. The operation with the name *Sites()* return all ITI objects of *'Site'* kind that are present in an ITI. In the context of itiPM, objects of *'Site'* kind are represented as instances of type *itiDirectoryServer*. The operation *Sites()* relies on the operation *IsSite()* described earlier to identify from all instances of type *itiDirectoryServer* those that are of the *'Site'* kind. The operation *IsSite()* looks into the value of the attribute *objectClass* to select the *itiDirectoryServer* objects of the *'Site'* kind.

Listing 3.28: itiLib operations for sites

```

Sites(): Set(itiDirectoryServer) = itiDirectoryServer.allInstances->
    select(IsSite())
SitesName(): Bag(String) = Sites().Name
```

The operation *SitesName()* was created to identify the name of objects of *'Site'* kind. The attribute *Name* is indirectly inherited from the definition meta-class.

As presented earlier, the objects of type *itiDirectoryServer* can correspond to ITI objects of *'server'* kind or to ITI objects of *'Site'* kind. The operation *SystemMembers()* identifies the servers for a given site.

Listing 3.29: Identify sites

```

SystemMembers(): Set(itiDirectoryServer) =
  if (IsSite()) then
    itiContain.allInstances->select(SiteMember = self).ServerMember.
    select(oclIsKindOf(itiDirectoryServer)).oclAsType(itiDirectoryServer
        )->asSet
  else
    oclUndefined(Set(itiDirectoryServer))
  endif
SiteServers() : Set(itiDirectoryServer) = SystemMembers()
```

All the ITI servers must belong to a site, so to identify the servers of a given site, we used the *ContainmentDefinition* relationship. The operation *SiteServers()* was created to simplify the usage of the operation *SystemMembers()*.

Since all ITI subnets must belong to a site, we used a similar process to identify the ITI objects of *'Subnet'* kind for a given site. The main difference resides in the utilization of the object of type *ResourceDefinition*, instead of *itiDirectoryServer*.

Listing 3.30: Identify subnets

```

ResourceMembers(): Set(itiSiteSubnet) =
  if (IsSite()) then
    itiContain.allInstances->select(SiteMember = self).ServerMember.
    select(oclIsKindOf(itiSiteSubnet)).oclAsType(itiSiteSubnet)->asSet
  else
    oclUndefined(Set(itiSiteSubnet))
  endif
  SiteSubnets() : Set(itiSiteSubnet) = ResourceMembers()
```

The operation *SiteSubnets()* was created to identify the subnets and simplify the usage of the operation *ResourceMembers()*.

The operation *Connections()* was created to identify all the connections among servers and sites in ITI. The *CommunicationDefinition* relationship display the results and rely on the operation *IsSite()* described earlier to identify, from all instances of type *itiDirectoryServer*, those that are of the *'Site'* kind.

Listing 3.31: Identify connections

```

  Connections(): Set(itiConnect) = itiConnect.allInstances->select(
      IsNTDSConnection())
  ConnectionsName(): Bag(String) = Connections().Name
  NTDSConnections(): Set(itiConnect) = Connections()->select(
      IsNTDSConnection())
  IsNTDSConnection(): Boolean = objectClass = nTDSConnection
```

The operation *ConnectionsName()* was created to identify the name of objects of *'Connection'* kind. The operation *Connections()* returns objects of *CommunicationDefinition* relationship type, which can be objects used to communicate with servers or sites. The *NTDSConnections()* operation returns only the connections used to communicate with servers. *IsNTDSConnection()* is a boolean operation that returns *true* if the value of the attribute *objectClass* is *'nTDSConnection'* . The value *'nTDSConnection'* is used to identify the connections among servers.

The operation *Sitelinks()* returns objects used to communicate with sites. This operation relies on *IsSiteLink()* boolean operation to identify if a particular connection is related with sites.

Listing 3.32: Identify links among Sites

```

Sitelinks():Set(itiConnect)=itiConnect.allInstances->select(IsSiteLink()
    )
IsSiteLink(): Boolean = objectClass = siteLink
```

The value *'siteLink'* of attribute *objectClass* in *IsSiteLink()* operation is used to identify the connections among sites.

In ITI sites with more than one server, as the ones presented in some of the case studies, the servers can communicate to servers in the same site, or they can communicate with servers in other sites. The operation *Intrasite_connections()* was created to identify the objects of *'connection'* kind that are used to connect to other servers in the same site.

Listing 3.33: Identify connection type (intra or inter)

```

Intrasite_connections() : Set(itiConnect) =
  if (IsSite()) then
    NTDSConnections()->select(IsConnectionInternalInSite(self))
  else
    oclEmpty(Set(itiConnect))
  endif

Intersite_connections() : Set(itiConnect) =
  if (IsSite()) then
    IncomingConnections()->union(OutgoingConnections())
  else
    oclEmpty(Set(itiConnect))
  endif
```

The operation *Intersite_connections()* was created to identify the objects of *'connection'* kind that are used to connect to servers in another site. From the list of connections among sites in an ITI, the *IncomingSiteLinks()* operation returns only those connections that are incoming (used to receive information) from other sites.

Listing 3.34: List site connections type (incoming or outgoing)

```
IncomingSiteLinks() : Set(itiConnect) =
  if (IsSite()) then
    SiteLinks()->select(IsIncoming(self))
  else
    oclEmpty(Set(itiConnect))
  endif

OutgoingSiteLinks() : Set(itiConnect) =
```

```
  if (IsSite()) then
    SiteLinks()->select(IsOutgoing(self))
  else
    oclEmpty(Set(itiConnect))
  endif
```

The *OutgoingSiteLinks()* operation returns only those connections that are outgoing (used to send information) to another sites.

Depending on the ITI topology, a site may be connected to one or more sites. The *ConnectedSites()* operation is used to identify these sites.

Listing 3.35: Connections for a specific site

```
ConnectedSites() : Set(itiDirectoryServer) =
  if (IsSite()) then
    IncomingSiteLinks().Client()->union(OutgoingSiteLinks().Server())->
        asSet
  else
    oclEmpty(Set(itiDirectoryServer))
  endif
```

3.5.3.3 itiLib operations for Servers

We have defined in *itiLib* a set of operations to quantitative evaluate different aspects of ITI objects of *'Server'* kind. The operation with the name *Servers()* return all ITI objects of *'Server'* kind that are present in an ITI. In the context of *itiPM*, objects of *'Server'* kind are represented as instances of type *itiDirectoryServer*. The operation *Servers()* relies on the operation *IsServer()* described earlier to identify, from all instances of type *itiDirectoryServer*, those that are of the *'Server'* kind.

Listing 3.36: All servers

```
Servers(): Set(itiDirectoryServer) = itiDirectoryServer.allInstances->
    select(IsServer())
```

The operation *IsServer()* looks into the value of the attribute *objectClass* to select the *itiDirectoryServer* objects of the *'Server'* kind.

Every server in the ITI must have a parent of the *'Site'* kind. This relationship is expressed with the *ContainmentDefinition* relationship type. The *ServerSite()* operation returns the site for a given server.

Listing 3.37: Identify the site of a given Server

```
ServerSite(): itiDirectoryServer =
  if (IsServer()) then
```

```
    itiContain.allInstances->select(ServerMember = self).SiteMember.
    select(oclIsKindOf(itiDirectoryServer)).oclAsType(itiDirectoryServer
        )->asSequence->first()
  else
    oclUndefined(itiDirectoryServer)
  endif
```

As explained earlier, the communication among servers or sites requires the existence of *Endpoints*, which can be classified as input or output, depending on if they are used to receive or to send information. The *InputEndpoints()* and *OutputEndpoints()* operations identify the type of *Endpoints*.

Listing 3.38: Endpoint type (client or server)

```
InputEndpoints(): Set(EndpointDefinition) = EndpointMembers()->select(e:
    EndpointDefinition | e.IsInput())
OutputEndpoints(): Set(EndpointDefinition) = EndpointMembers()->select(e
    : EndpointDefinition | e.IsOutput())
```

In an ITI, servers are connected in the sense that they communicate. The number of servers that a specific server is connected to differs from ITI to ITI and is dependent of the topology. The *ConnectedServers()* operation returns the servers connected to a given server.

Listing 3.39: Servers connected to a specific server

```
ConnectedServers() : Set(itiDirectoryServer) =
  if (IsServer()) then
    IncomingConnections().Client()->union(OutgoingConnections().Server()
        )->asSet
  else
    oclEmpty(Set(itiDirectoryServer))
  endif
```

When multiple servers exists in the same site, each server in the site can send or receive information from multiple servers. The *Intraserver_FanIn()* and *Intraserver_FanOut()* operations return the number of connections used to receive information from other servers in the same site and to send information to servers in the same site.

Listing 3.40: Number of server connections

```
Intraserver_FanIn(): Integer =
  if (IsServer()) then
    ServerSite().Intrasite_connections()->select(IsIncoming(self)) ->
        size
  else
    0
```

```
  endif

Intraserver_FanOut(): Integer =
  if (IsServer()) then
    ServerSite().Intrasite_connections()->select(IsOutgoing(self)) ->
        size
  else
    0
  endif
```

The *SSEN()* operation is based on the *SiteServers()* and is used to count the number of servers in a site.

Listing 3.41: Count the number of servers in a site

```
SSEN(): Integer =
  if (IsSite()) then
    SiteServers()->size
  else
    0
  endif
```

3.5.3.4 itiLib operations for Sites and Servers

The *EndpointMembers()* operation is applicable to servers and sites and returns the connections (*nTDSConnection* or *siteLink*) for servers or sites.

Listing 3.42: All Endpoints

```
EndpointMembers(): Set(EndpointDefinition) =
  if (IsServer() or IsSite()) then
    itiContain.allInstances->select(SiteMember = self).ServerMember.
    select(oclIsKindOf(EndpointDefinition)).oclAsType(EndpointDefinition
        )->asSet
  else
    oclUndefined(Set(EndpointDefinition))
  endif
```

The *IncomingConnections()* operation returns objects of type relationship definition. This operation can be applied to servers or sites and returns the connections used to receive information. If a site is provided, the incoming connections relatively to the current site are returned; otherwise, the incoming connections relatively to the server are returned. The *OutgoingConnections()* operation also returns objects of type relationship definition and can be applied to servers or sites. These operations is used to return the connections used to send information.

Listing 3.43: Connections type for a specific site (incoming or outgoing)

```
IncomingConnections() : Set(itiConnect) =
  if (IsSite()) then
    -- Returns the incoming connections relatively to the current site
    NTDSConnections()->select(IsConnectionIncomingInSite(self))
  else
    if (IsServer()) then
    -- Returns all incoming connections (even from other sites)
      NTDSConnections()->select(IsIncoming(self))
    else
      oclEmpty(Set(itiConnect))
    endif
  endif

OutgoingConnections() : Set(itiConnect) =
  if (IsSite()) then
    -- Returns the outgoing connections relatively to the current site *
    NTDSConnections()->select(IsConnectionOutgoingInSite(self))
  else
    if (IsServer()) then
      NTDSConnections()->select(IsOutgoing(self))
    else
      oclEmpty(Set(itiConnect))
    endif
  endif
```

The *FanIn()* operations is used to count the number of connections used to receive information and (*FanOut()*) counts the number of connections used to send information. These operations can be applied to servers or sites.

Listing 3.44: List connections that send or receive data (FanIn or FanOut)

```
FanIn(): Integer = Connections()->select(IsIncoming(self)) -> size
FanOut(): Integer = Connections()->select(IsOutgoing(self)) -> size
```

3.5.3.5 itiLib operations for Subnets

The *SubnetSite()* operation allows the identification of the site corresponding to a given subnet. This operation searches all containment relationships where the member is the subnet and returns the site.

Listing 3.45: List the site of a particular subnet

```
SubnetSite(): itiDirectoryServer =
  if (IsSubnet()) then
    itiContain.allInstances->select(ServerMember = self).SiteMember.
```

```
    select(oclIsKindOf(itiDirectoryServer)).oclAsType(itiDirectoryServer
        )->asSequence->first()
  else
    oclUndefined(itiDirectoryServer)
  endif
```

The *SSBN()* operation relies on the *SiteSubnets()* operation to count the subnets in a specified site.

Listing 3.46: Count subnets in a specific site

```
SSBN(): Integer =
  if (IsSite()) then
    SiteSubnets()->size
  else
    0
  endif
```

To list all existing subnets, which are from type *ResourceDefinition*, the concept is similar to sites and servers. However, instead of using *itiDirectoryServer*, we have to use *ResourceDefinition* and look for subnet in the value of the attribute *objectClass*.

Listing 3.47: List all subnets

```
Subnets(): Set(itiSiteSubnet) = itiSiteSubnet.allInstances->select(
    IsSubnet())
```

3.5.3.6 itiLib operations for SiteLinks

It may be important to identify the name of all existing sites within an ITI. The following operations make that possible.

Listing 3.48: List all site names

```
Sitelinks():Set(itiConnect)=itiConnect.allInstances->select(IsSiteLink()
    )
SitelinksName(): Set(String) = Sitelinks().Name->asSet
```

From the list of all existing connections in the ITI, we may want to know to what server does one specific connection belongs to.

3.5.3.7 itiLib operations for Connections

In the communication definition we already defined operations such as *isNTDSConnection()* to identify the connections of the servers and *IsSiteLink()* to identify the connections among sites. However, in the cases with more than one server per site, we may also want to identify the connections to other servers in the same site and the connections to servers in other sites. In order to achieve this, we have created the *IsConnectionInternalInSite()* and *IsConnectionOutgoingInSite()* predicates. The latter relies on the operations *Client()* and *Server()* to help on the identification.

Listing 3.49: Connections to servers in the same site

```
IsConnectionInternalInSite(theSite: itiDirectoryServer): Boolean =
  (Client().ServerSite() = theSite) and (Server().ServerSite() =
      theSite)
IsConnectionIncomingInSite(theSite:itiDirectoryServer): Boolean =
  (Client().ServerSite() <> theSite) and (Server().ServerSite()=
      theSite)
IsConnectionOutgoingInSite(theSite:itiDirectoryServer): Boolean=(
    Client().ServerSite() = theSite) and (Server().ServerSite() <>
    theSite)
IsIncoming(target: itiDirectoryServer): Boolean = Server() = target
IsOutgoing(target: itiDirectoryServer): Boolean = Client() = target
Client(): itiDirectoryServer = SourceReplication.Container()
Server(): itiDirectoryServer = DestinationReplication.Container()
```

3.6 Bootstrapping the Application of a Real Example

As described in section 3.3, to be able to represent an ITI using the M2DM approach, we need a metamodel in UML format to express the concepts and the relations of the domain to measure (in our case ITIs). As mentioned before, to be able to accomplish our goals we created the itiPM metamodel described earlier. With the itiPM metamodel we were able to formally express semantics enforcement and metrics for ITIs using the OCL language.

To better demonstrate the applicability of itiPM metrics, we decided to use an ITI with hundreds of sites as the one provided in figure 3.25. The figure is presented in a very small size to guarantee the confidentiality of the data origin, since the data was gathered from a real organization in the finance industry. In this particular case, we used the data from a Lightweight Directory Access Protocol (LDAP) repository[9]. The LDAP repository is an existing stereotype of **itiPM** available in package **ITI Software**. Using the data from figure 3.25 instantiated in the itiPM metamodel enables any metric defined in itiPM to be run against this ITI.

Each of the green boxes in represents a site and the lines represent connections among sites. Each site has objects such as servers and subnets inside. The topology of this ITI is centralized, what means that every site is connected to a central site. To use the metrics defined in our itiPM metamodel, there are five main steps that should be performed:

[9]Application protocol for accessing and maintaining distributed directory information services over an Internet Protocol network.

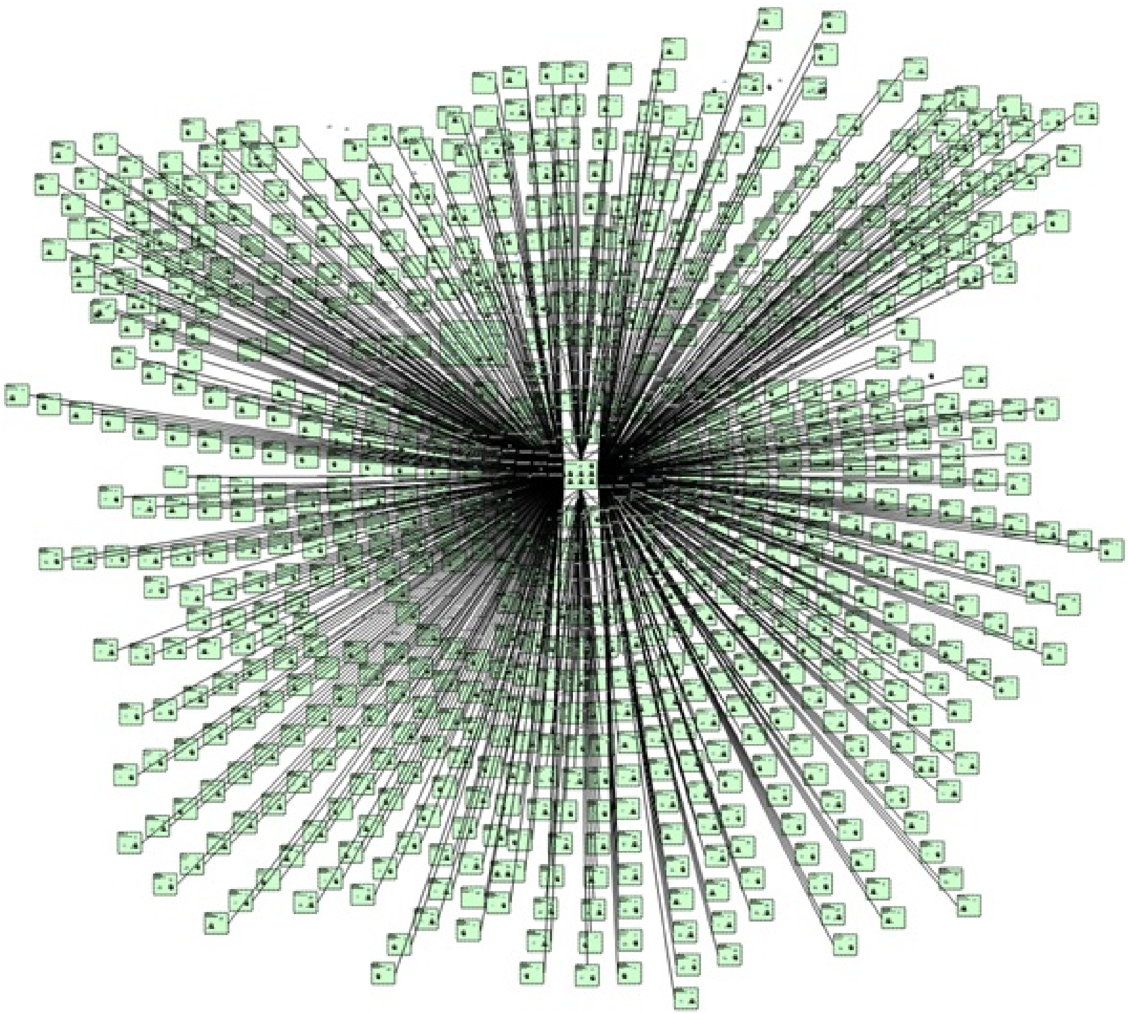


Figure 3.25: ITI with a large number of sites (blurred for confidentiality reasons)

- **Step 1 - Gather ITI data.** The process of gathering ITI data can be performed manually, by looking at the infrastructure and creating and defining each object and their attributes and values (only feasible in relatively small ITIs) or automatically by running queries to a directory service repository. The example provided in this section was obtained automatically by querying Active Directory which is a Microsoft implementation of LDAP service.
- **Step 2 - Transform and prepare data.** Using an automated way to gather ITI data, requires some work to transform and prepare the data obtained in CSV format, compatible with **itiPM**. We called this data ITI meta-instances since they will be used to instantiate our metamodel.
- **Step 3 - Compile itiPM.** This step consists in compiling the metamodel component comprising all the meta-classes, constraints and stereotypes.
- **Step 4 - Load ITI meta-instances.** After loading the **itiPM**, we loaded the ITI meta-instances generated in step 2. This process consists in loading data and instantiating this data into the **itiPM** compiled in step 3;

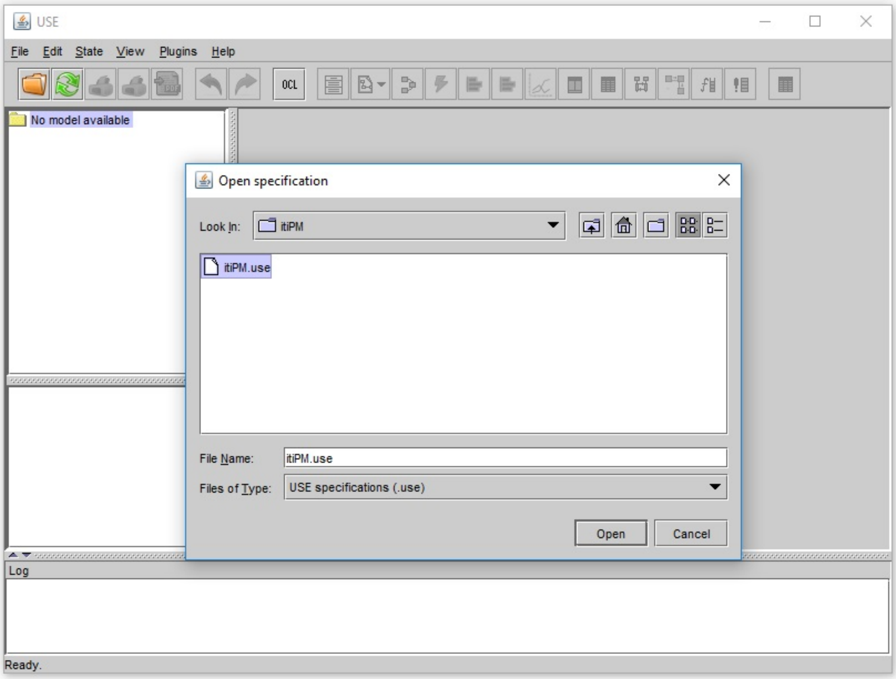


Figure 3.26: Open USE specification (Step 3a)

- **Step 5 - Run queries.** With all the data instantiated, we are able to query data, obtain quantitative information, check constraint violations, among other aspects.

The following discussion assumes we already have generated the meta-instances corresponding to the ITI in figure 3.25. This means that we already performed the steps 1 and 2 outlined in our approach. The next steps are to load the itiPM metamodel into USE and then load the meta-instances corresponding to this ITI.

To load the itiPM metamodel we open the application USE tool, and we open the window "Open Specification" to be able to select our "itiPM.use" as illustrated in figure 3.26. Notice in the left pane of the USE tool the empty model, meaning that no model is currently loaded. This step will only succeed if our itiPM.use file has no errors. After loading the specification, the USE tool automatically compiles the file and displays in the log pane the result of the compilation process and the number of classes, associations, invariants pre-/postconditions and operations as illustrated in figure 3.27. Notice that our itiPM.use metamodel contains 104 operations created in the context of this dissertation, corresponding to the *itiLib* library presented earlier.

With the itiPM metamodel loaded, it is possible to have an overview of the existing UML classes, their attributes and relationships. Figure 3.28 shows the itiPM classes loaded into USE.

With the initial steps of our approach already performed, the next step (step 4) consists in loading the meta-instances corresponding to the ITI of figure 3.29. Since we already have the meta-instances generated and stored in a file (with the name ITIdetails.cmd) we load the ITI data into USE.

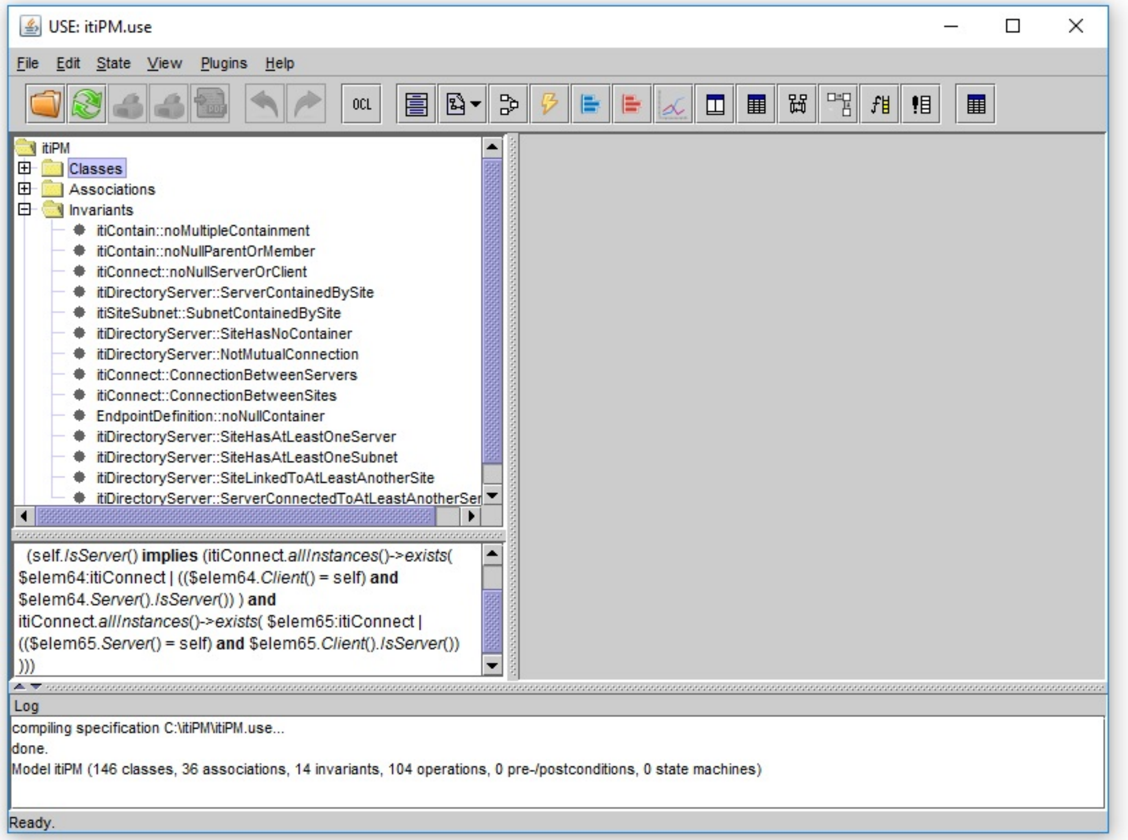


Figure 3.27: Load itiPM metamodel into USE (Step 3b)

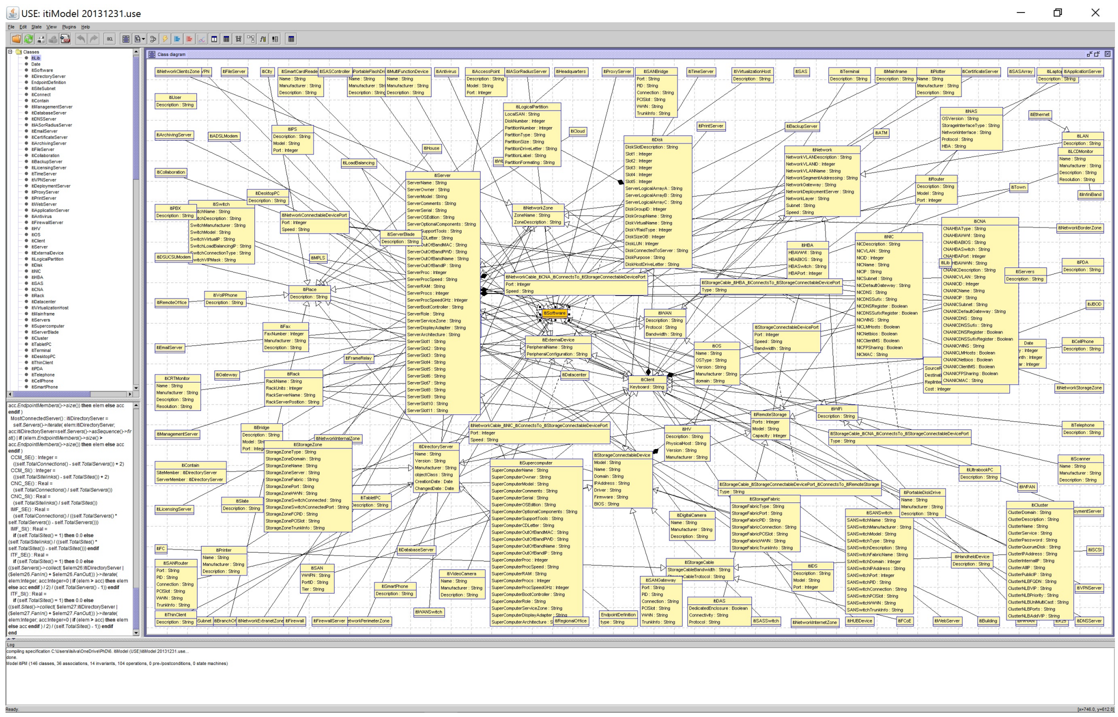


Figure 3.28: itiPM class diagram visualization (step 3c)

Figure 3.29: Load ITI meta-instances (Step 4a)

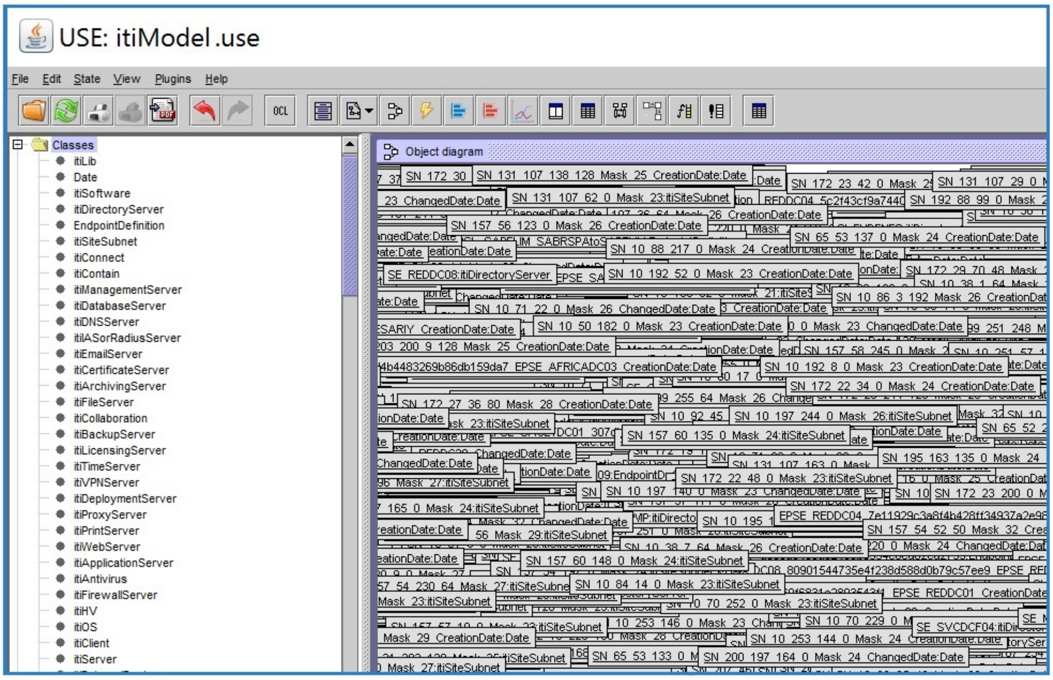


Figure 3.30: ITI object diagram (Step 4b)

With all ITI data loaded, it is possible to perform queries and analyze results (step 5). Some of the operations that are implemented and can be run against this ITI are:

- **Total number of sites.** This metric was defined in itiLib as *TotalSites()* and returns the total number of sites in a particular ITI;
- **Total number of servers.** This metric was defined in itiLib as *TotalServers()* and returns the total number of servers in a particular ITI;
- **Total number of subnets.** This metric was defined in itiLib as *TotalSubnets()* and returns the total number of subnets in a particular ITI;
- **Total number of connections.** This metric was defined in itiLib as *TotalConnections()* and returns the total number of connections among servers or nTDSconnections in a particular ITI;
- **Total number of sitelinks.** This metric was defined in itiLib as *TotalSiteLinks()* and returns the total number of connections among sites or sitelinks in a particular ITI;
- **The average of servers by site.** This metric was defined as *AverageServersPerSite()* and returns a real number corresponding to the average of servers per site;
- **The average of subnets per site.** This metric was defined as *AverageServersPerSite()* and returns a real number corresponding to the average of subnets per site;
- **The average of connections per server.** This metric was defined as *AverageConnectionsPerServer()* and returns a real number corresponding to the average of server connections per server;
- **The maximum number of servers in a single site.** This metric was defined as *MaxServersPerSite()* and returns the maximum number of servers in a single site. This metric discovers which site has the highest number of servers and returns that maximum.

We created the class *ITIoperations* to define all these counting or sizing metrics. Most of the operations that return the total number of objects were created with the prefix 'Total' for understandability reasons. Now that the sizing metrics are defined, we can apply them to obtain sizing information regarding the ITI presented in figure 3.25. In order to do that, we will continue using the command line provided by the USE tool. The commands use to evaluate expressions in USE start with a question mark (?). The available commands in USE were presented in figure 3.29.

The output presented in listing 3.50 is the result of the application of each sizing metric in the ITI presented before. There are several variations to these sizing metrics in itiLib, such as to provide the name of servers instead of the number or name of the site with the maximum number of servers, instead the maximum number of servers in a site. These results complete the last step of our approach to evaluate ITIs.

Listing 3.50: ITI analyses

```
use> ! create op:ITIoperations
use> ? op.TotalSites()
```

```
-> 1167 : Integer
use> ? op.TotalServers()
-> 1205 : Integer
use> ? op.TotalSubnets()
-> 4264 : Integer
use> ? op.TotalConnections()
-> 1928 : Integer
use> ? op.TotalSitelinks()
-> 2184 : Integer
use> ? op. AverageServersPerSite()
-> 1.0325621251071122: Real
use>
use> ? op.AverageSubnetsPerSite()
-> 3.6538131962296486: Real
use>
use> ? op.AverageConnectionsPerServer()
-> 1.6 : Real
use>
use> ? op.MaxServersPerSite()
-> 7 : Integer
```

As can be seen, this approach allows the evaluation of several dimensions of an ITI. In this case study, and with the ITI information loaded into USE, it was relatively straightforward to obtain the total number of sites, servers, subnets, connections among other sizing parameters as demonstrated in the listing above.

Based on the image of 3.25 and according to our expectations, the number of sites of the provided ITI is large. However, most sites have only one server in it.

3.7 Summary

This chapter described the approach proposed to modeling ITIs. It started with an metamodel evaluation, describing the characteristics of each metamodel, in terms of their capacity to express ITIs. After the evaluation and due to the lack of expressiveness in the domain of ITIs of existing metamodels, it was decided to extend UML and create an UML Profile for ITIs. The ITI metamodel creation, structure, notation and packages are presented in section 3.4.

Section 3.5.1 describes the main components of the ITI profile, including notation, toolboxes, stereotypes, metaclasses and tagged values. The steps performed to implement the ITI profile in Sparx EA tool are also discussed. Section 3.5.2 explains the rationale for using the USE tool and presents concrete syntax examples, an exercise on how the best practices can be enforced and how to detect best practices violation.

The library itiLib is described in section 3.5.3, which consists in a set of operations to collect relevant information from the meta-data and support the evaluation of ITIs. Section 3.6 describes the collection of itiPM metrics in a real ITI with hundreds of sites.

Chapter 4

ITI Patterns

Contents

4.1 Introduction

In the ITI domain, lurking behind recurring problems, there are recurring solutions, and we argue that the use of design patterns and pattern languages can be beneficial, if correctly written and organized.

4.2 History of Design Patterns

According to the etymology dictionary[1] the word "pattern" dates back to the early 14th century and originates from the old French *"patron"* and directly from Medieval Latin *patronus* and it was used to mention something from which a copy is made; a model or outline. In the 15th century, the term was used to refer to someone or something seen as an example to be imitated; an exemplar. The difference in form and sense between *patron* and pattern was not firm until 1700s when their meaning started to be used as a "model or design" to be imitated, mainly in the area of dressmaking. The concept of pattern has been, from the very beginning, generic and applied to multiple areas, which explains the reason why patterns are nowadays used in many domains.

More recently (between the 1960s and the 1970s) the architect Christopher Alexander, used the concept of patterns in their own theories focused on the nature of human-centered design, which had notable and unexpected impacts across many fields beyond Architecture, including Urbanism, Computer Science, Sociology and other fields. His book, *"Notes on the Synthesis of Form"* published in 1964, [Ale64] about the process of design was considered "one of the most important contemporary books about the art of design, what it is, and how to go about it. His paper *"A City is Not a Tree"* [Ale65] was considered recently as "one of the classic references in the literature of the built environment and related fields."

With the collaboration of Murray Silverstein, Shlomo Angel, Sara Ishikawa, and Denny Abrams, Christopher Alexander published the book "The Oregon Experiment"[Ale+75] which describes an experimental approach to campus community planning at the University of Oregon, in Eugene. This work resulted in the two-best known Christopher Alexander books, which were *"A Pattern Language: Towns, Buildings, Construction"* [Ale+77] to empower anyone to design and build at any scale and *"The Timeless Way of Building"* [Ale79] which introduced the concept of "pattern" in the area of Architecture. These books had a huge influence on creative thinking, especially in the areas of Architecture and Computer Science on the field of software design [Lea94].

Alexander define patterns as a *"...problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over and over, without ever doing it the same way twice"* [Ale79; Ale+77]. Each design pattern provides systematic documentation of a proven solution to a recurrent problem, by systematizing lessons learned. Some patterns are

[1] http://www.etymonline.com/ visited in Apr 4, 2017

loosely coupled, but others are more interwoven. A set of highly coupled patterns is often called a pattern language. Generally speaking, a pattern language is a practical network of important related ideas, that provides a comprehensive treatment of a subject, using a common vocabulary and understanding. These networks of ideas are the heart of a pattern language since they provide more value than a single pattern.

4.2.1 Patterns in Computer Science

In Computer Science, the interest in the use of design patterns began in the 1980's with the proposal of *literate programming* by Donald Knuth in 1984 which is a programming paradigm that used natural language to explain the program logic [Knu84]. The main idea was to treat program as a literature understandable to human beings. A few years later, [CB86; Ewi86] had successful experimentations with the development of elegant user interfaces in Smalltalk and with the creation of patterns as guidelines for using Smalltalk inspired on Christopher Alexander's ideas.

The first session about patterns was organized by Bruce Anderson at OOPSLA[2] in 1990, where Erich Gamma and Richard Helm met and discovered their common interest. They were later joined by Ralph Johnson and John Vlissides. Together they created the book "Design Patterns: Elements of Reusable Object-Oriented Software" that was published in 1994 and sold more than 500000 copies in English and in 13 other languages. The authors are often referred to as the *Gang of Four (GoF)* [Gam+94].

The patterns quickly became a type of literature used to solve common problems in different Computer Science areas. Jim Coplien worked in organizational patterns and he created a pattern language to cover management of organization and projects with the focus on teamwork [CS95]. Later Frank Buschmann worked on software architectures and proposed the concept of a *system of patterns* [Bus+96]. Martin Fowler published a book on *analysis patterns* that covered requirements analysis and analytical modeling [Fow97]. Alistair Cockburn's book covered the area of object-oriented projects and included a set of *project management patterns* [Coc98]. Fernando Brito e Abreu published *patterns for pedagogical purposes* [Abr97; Abr98]. Robert Binder published the use of *patterns for software testing* [Bin99]. Markus Völter, Alexander Schmid and Eberhard Wolff published *server component patterns* for building server-side component infrastructures [Völ01]. James Lin in the context of web sites, integrated patterns into a design-sketching environment [Lin+02]. James Noble and Charles Weir published the book Small Memory Software [NW05] where they presented a set of *patterns for software development* in the context of limited memory resources [NW05]. Since then, many books and papers have been published in journals and conference proceedings such as the EuroPLoP (European conference on software patterns) and in other Pattern Languages of Programs (PLoP) conferences (e.g. ChiliPLoP, KoalaPLoP, MensorePLoP, SugarLoafPLoP).

[2] Object-Oriented Programming, Systems, Languages and Applications

4.2.2 Patterns Applied to ITIs

In the realm of ITIs, and more specifically in their design, the concept of design patterns did not become popular and is not widely used. A serious effort related with the use of design patterns was performed in the context of web sites [Lin+02]. This approach integrates patterns into a design-sketching environment, allowing designers to drag and drop reusable components for recurring page elements, such as navigation bars, as well as conditionals, to illustrate and test transitions that depend on user's input. Other efforts such as [PI13] were performed in the domain of Enterprise Architectures, but at a higher-level to support services in the other Enterprise Architectures layers.

One of the major problems faced by ITIs is their increasing size and complexity, that may jeopardize the delivery of real business value [Ses08]. As mentioned before, the size and complexity are often the result of ITIs created, designed or adapted by non ITI experts such as business decision makers, consultants, administrators, developers, software engineers, solution architects and other individuals (sometimes with conflicting perspectives due to their own point of view) without ITI design guidelines. Designing or changing ITIs is a challenging task, mainly because it requires knowledge of existing organization processes, the views of different players, and the coordination of technical expertise in multiple ITI domains.

4.2.3 Patterns Organizations and Forms

The desideratum of using a given strategy to schedule identified problems and using an appropriate composition of solutions, basically corresponds to what differentiates a patterns catalog from a pattern language. However, in both cases, the patterns organization is important to help find the most adequate pattern for a given problem. There are several ways to organize patterns, such as by domain, by function, or problem type. For instance, in [Gam+94] they were categorized in three groups (Creational, Structural and Behavioral). The books Pattern-Oriented Software Architecture (POSA) [Bus+96] used criteria such as interactive and adaptable systems, organization of work, communication and access control. However, we should keep in mind, as observed by Fowler [Fow97], that it is better and more valuable to have poorly organized good patterns, than to have well organized weak patterns. *Pattern forms*, in turn, correspond to the pattern representation or the format used to write a pattern. Most authors tend to create their own pattern form. However, some pattern forms become better known than others. Some of the most well-known forms are:

- **Alexandrian**. Narrative form, with relatively few headings. It has the following sections: title, prologue, problem statement, discussion, solution, diagram, and epilogue. With this form of organization, it allows the reader to analyze a large number of patterns very quickly [Ale+77];

- **Coplien**. Frequently used in software to speed comparisons and lookups and it is one of the simplest forms. It has the following sections: name, alias (optional), problem, context, forces, solution example (optional), resulting context, rationale (optional), known uses, related patterns. This pattern form is also referred to as "canonical form"[Cop97];

- **GoF**. This form is one of the largest and most structured forms and has the following sections: intent, motivation, applicability, structure, participants, collaborations, consequences, implementation, sample code, known uses, and related patterns [Gam+94];

- **POSA**. Very structured form, similar to GoF with the sections: summary, example, context, problem, solution, structure, dynamics, implementation, example resolved, variants, known uses, consequences, and see also. An important part of this form is that patterns are preceded by a narrative section that summarizes the patterns and describes the overall topic [Bus+96].

4.3 ITI Patterns

The use of ITI patterns can be seen as a process to simplify the ITI design process, through the use of well-known solutions for recurrent problems. The solutions addressed by design patterns are not intended to be static and final. In fact, they are templates that can be customized and extended. Design patterns help mitigating ITI complexity by allowing architectural decisions to be taken at a higher abstraction level.

Based on our field experience and in the survey presented in chapter 2 we found that the problems faced in ITIs do not vary much from organization to organization. Having a non-proprietary structured approach that quickly allows the reuse of solutions in the design of ITIs, can be beneficial for organizations in many different ways. Some benefits that are expected to arise from the use of patterns in ITIs, were already briefly mentioned in section 1.2.3 and include building complex and heterogeneous solutions, comprising multiple source technologies, promoting the sharing of ideas, providing a guided approach to solving problems, improving design quality and efficiency [Tro+03]. Our experiments, described in chapter 5, and the results of the online survey, corroborate these findings.

4.3.1 Creating and Refining an ITI Pattern

Some of the ITI patterns produced, such as the **REMOTE SOFTWARE DISTRIBUTOR** that will be described later in this section were published and evaluated in a PLOP conference [SA10a]. After being accepted in the PLOP conference, they went through a process called "*shepherding*", in which we received feedback from experienced pattern authors during the ITI pattern creation. Based on this shepherding, we completely changed the way we used to write patterns. As an example, we always started producing the context section and then we realized that the process of writing patterns is an interactive process and the order to write the pattern is usually different from the order it will be presented. As described in [WF12], the recommended order of writing is *Solution*, *Problem*, *Consequences* and then *Forces* and *Context*.

4.3.2 Adopted ITI Patterns Organizations

In terms of pattern organization and forms, as stated before, there are many different description formats, and we decided to apply the Coplien pattern form, mainly due to popularity and the fact that all its sections satisfy us in terms of what we considered relevant for an ITI pattern. Most of the sections of the Coplien pattern form could be easily applied to ITI patterns, with some obvious exceptions such as "Sample Code".

Our ITI patterns are organized with the following sections:

- **Context.** Consists in the stage where the ITI pattern takes place or the situation in which the solution is applicable. The context describes the conditions under which the problem can be solved by the pattern.
- **Example.** Consists in a practical example that can describe the problem. It may be omitted depending on the context.
- **Problem.** Explains what the actual problem is, in the described context. The problem is the reason why the solution has to be applied.
- **Forces.** Describes why the problem is difficult to solve. A force explains the cause for a specific solution.
- **Solution.** The solution describes how the problem can be resolved.
- **Consequences.** Identifies what happens when you apply the solution in terms of benefits, costs, drawbacks, trade-offs and liabilities of the solution.
- **Example resolved.** Explains how the solution solves the problem in the example. It may be omitted depending on the context.
- **Related patterns.** Identifies relationships among patterns. The relationships are an important aspect to make a specific pattern part of a "pattern language" rather than just a set of isolated patterns.

4.3.3 ITI Patterns Catalog

We grouped all the ITI patterns in what we call an ITI patterns catalog. A pattern catalog is essentially a collection of related patterns in the domain of ITIs which are only informally related. The main difference between a catalog and pattern languages resides in the fact that pattern languages are more complete in the sense that they contain more patterns, their variations and all allow the creation of architectures by using multiple patterns. The creation of a pattern language for ITIs is important and considered future work. Table 4.1 shows our ITI pattern catalog with 50 ITI patterns along with their name and intention.

Table 4.1: ITI pattern catalog

#	Pattern Name	Intent
1	Application Network Zone	Defines the rules for a network zone that hosts all the applications main logic that expose interfaces like web services and others.
2	Border Router Security Lockdown	Provides security configurations that can be used in high availability environments to secure the border routers.
3	Border Switch Security Lockdown	Provides security configurations for switches located in border zones to improve security and high availability.
4	Corporate Branch Offices	Defines how to structure corporate branch offices and how to connect them in a secure way to headquarters.
5	Business-Layer Server	Describes the key guidelines for designing the business layer of an application and provide the configurations to integrate servers into a typical layered architecture.
6	Cloud Dedicated Connector	Defines how to securely connect a private infrastructure to a public cloud infrastructure, using a dedicated connection.
7	Cloud Public Connector	Describes how to connect corporate internal networks to Public Cloud networks, using a secure connection over the internet.
8	Cloud VPN Connector	Provides an high availability storage infrastructure with storage networks and LAN networks paths isolated.
9	CloudTraffic Business Applications	Describes best practices on how to configure and estimate the corporate internet bandwidth to access business applications in the cloud, without performance issues.
10	CloudTraffic Collaboration Applications	Provides for corporate internal users access to collaboration applications in a public cloud environment without performance issues by estimating the required bandwidth.
11	CloudTraffic Communication Applications	Provides the configurations required to allow the estimation of required internet bandwidth for communication applications.
12	CloudTraffic Estimator	Allows the estimation of required internet bandwidth when moving applications to a public cloud.

Continued on next page

Table 4.1 – continued from previous page

#	Pattern Name	Intent
13	CloudTraffic Shaper	Enables the understanding of network traffic by specifying different profiles and priorities to differentiate important cloud application traffic and lower-priority traffic.
14	Core Database	Defines security guidelines on how to define and deploy a database service for internal clients in an organization.
15	Core Directory	Defines guidelines on how to configure a directory service to allow internal users to access data, while avoiding unauthorized modification which could have an adverse effect on the organization.
16	Data Network Zone	Provides guidelines on how to create a clear separation of data within the network, to decrease the risk of compromising business critical information.
17	Data-Layer Server	Describes the key guidelines for configuring servers in the data layer and integrate them into a typical layered application architecture.
18	Directory System	Defines how to store, organize and provide access to information in a directory to corporate users.
19	External Proxies	Provides proxies configurations to handle recursive DNS services for outbound client traffic.
20	Failover Cluster	Provides a multiple server redundant solution that can be available, even when a complete server fails.
21	Fault Tolerant Server	Provide a full tolerant and scalable server configuration that can to be used in high availability solutions.
22	File and Print	Defines how to structure and configure the file and print services for internal use.
23	Border Security Firewall	Defines how to configure the internal interface on the border firewall between border public and the perimeter information services and perimeter name services zone.
24	Firewall Proxy	Defines the configurations for a firewall proxy server to mirror services, as if they were running on the end host and force sessions through the proxy.
25	HA Non-Converged Storage	Provide an high availability storage infrastructure with storage networks and LAN networks paths isolated.
26	Internal Access	Describes how to configure authentication of Virtual Private Network (VPN) clients, in a corporate environment.

Continued on next page

Table 4.1 – continued from previous page

#	Pattern Name	Intent
27	Internal Applications	Defines how to structure and configure application services for internal use, integrating them in a layered architecture.
28	Internal Zone	Defines how to architect an internal network to provide connectivity among IT infrastructure components.
29	ITI Border Zone	Provides best practices for the physical devices that support the communications infrastructure with public and perimeter zones.
30	ITI Fault Domain	Provides a solution to protect against hardware failures and increase application availability.
31	ITI Systems Management	Defines the configurations and how to structure the servers and components of a systems management infrastructure.
32	Load Balancing Server Farms	Provides an high scalable solution to be used in solutions that require both high availability and redundancy. The workload is distributed across two or more computers.
33	Messaging System	Describes how to design a messaging system taking into consideration specific service level agreements, related to availability, reliability, and scalability.
34	Non Fault Tolerant Server	Describes how to design a non fault tolerant server architecture that can however be scalable.
35	Perimeter Application	Defines how to structure and configure application services located in a perimeter network.
36	Perimeter DNS Services	Defines how to setup DNS services for queries made from public zones.
37	Tiered Perimeter DNS Servers	Provides the configuration for DNS servers hosted in a perimeter network with load balancing capabilities.
38	Perimeter Management	Defines how to architect and use management services securely in public zones.
39	Perimeter VPN	Defines how to provide remote access for internet-based clients using a VPN.
40	Perimeter Web Tier	Defines how to structure and configure load balancing for web servers located in a perimeter network.
41	Perimeter Zone	Describes the perimeter zone to allow the protection of internal assets by controlling all access from external networks.

Continued on next page

Table 4.1 – continued from previous page

#	Pattern Name	Intent
42	Presentation Network Zone	Defines how to design a presentation network zone to allow secure communication with other network zones.
43	Presentation-Layer Server	Provides the configurations for servers in the presentation network zone, to allow them to receive requests to other network zones which require authentication.
44	Private Zone	Defines the enterprise security best practices for corporate private zones.
45	Public Firewall	Defines the external interface configurations on the border firewall between public and the border public zone.
46	Public Zone Connectivity	Describes how to provide connectivity between the public and the border public zone.
47	RAS Firewall	Defines how to configure a border firewall between public and perimeter zone to provide remote access services.
48	Remote Software Distributor	Provide a solution to distribute software to remote locations.
49	Tiered Distribution	Defines how to structure servers and distribute load among them, in a complex distributed enterprise environment.
50	Site to Site VPN	Provides the guidelines to configure VPN services for client and site-to-site connections.

4.3.4 ITI Patterns Descriptions

During this research work we created and published the following ITI patterns, that allowed us to evaluate the executability of the approach:

- **Pattern Name:** Border Router Security Lockdown.
  Problem: *How to secure routers that connect internal LANs to the Internet through a WAN link?*
  Context: This pattern applies to organizations using routers to connect internal LANs to the internet through a WAN link and intend to secure routers to minimize the impact of a security violation. Since all data traffic from and to the internet passes through these routers, a security violation could lead to network failures or even theft of data. Routers facing the internet are typically known as edge or border routers.

- **Pattern Name:** Fault Tolerant Server

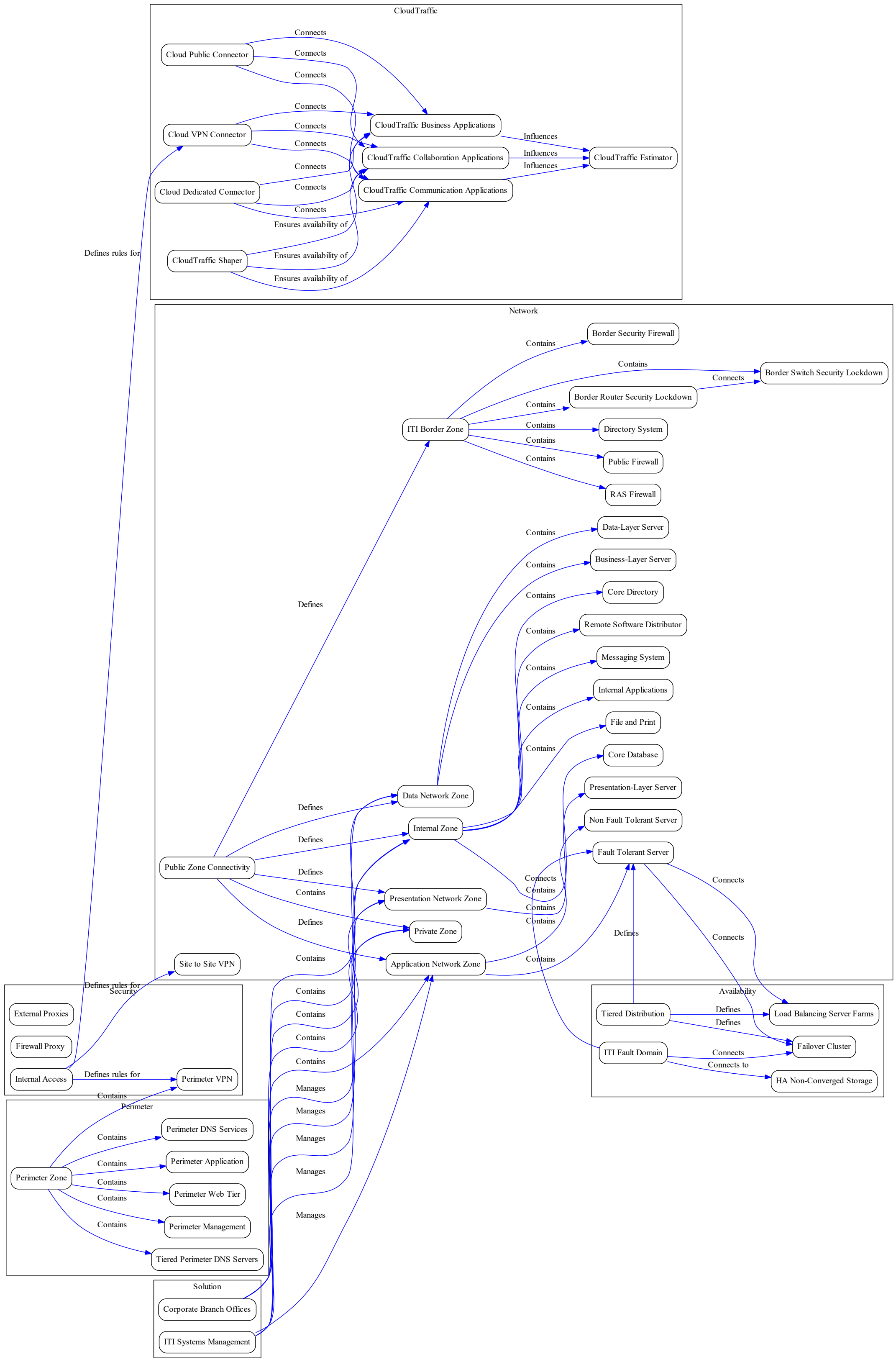


Figure 4.1: Relationships among ITI patterns

Problem: *Which hardware components should a server have to support business critical applications?*
Context: The definition of the required server hardware to support a new service, application or solution is a common problem for IT professionals, since it is a challenge to find the right hardware balance. For instance, over-used servers result in performance degradation and loss of end user productivity, while under-used servers can cause higher capital and operations expenses. The problem is even bigger when the application is critical for business.

- **Pattern Name:** Remote Software Distributor
  Problem: *How do you distribute software to devices in remote locations?*
  Context: This pattern applies to organizations that distribute software from a central place and have hundreds or even thousands of devices (e.g. servers, desktops, laptops, etc.) connected with local area networks (LANs) and wide area networks (WANs) in multiple remote locations. WANs are used to connect LANs and other types of networks together, to allow devices in one location to communicate with devices in other locations.

- **Pattern Name:** ITI Border Zone
  Problem: *How to provide access to the Internet to all the end-users without compromising the security?*
  Context: The ITI constitutes the foundation upon which organizations can deliver services to customers, partners and its employees. The network is part of ITI and combines a number of devices (e.g. routers, switches, load balancing devices, firewalls, etc.) that should be able to communicate in an efficient, reliable, and secure manner to provide high quality services. The border zone is the zone of the ITI that provides the connection to external networks, such as the internet.

- **Pattern Name:** ITI Fault Domain
  Problem: *How to protect against hardware failures and increase applications availability?*
  Context: This pattern applies to organizations that embrace cloud computing and require that an internal ITI has attributes such as scalability, elasticity, and fault-tolerance to support applications requiring high degree of availability. One of the main differences between traditional data centers and private clouds reside in the abstraction of physical resources (e.g. servers, disks, networks) which are logically grouped and mapped to the physical ITI, to allow efficient and intelligent provisioning and management.

- **Pattern Name:** CloudTraffic Estimator
  Problem: *How to estimate the required Internet bandwidth when moving applications to a public cloud?*

Context: The level of connectivity to access a public cloud is crucial to make the cloud deliver the best services. Different organizations have different sizes, require different services and have different needs. Depending on the bandwidth of the internet connection and the amount of data exchanged with the cloud provider, the experience for end users may vary. Most organizations simply assume they need a broadband IP VPN, whether or not the latter delivers the performance, reliability, availability and security required to access the public cloud.

In the following section, each of ITI patterns above will be presented in detail.

4.3.5 ITI Pattern - BORDER ROUTER SECURITY LOCKDOWN

4.3.5.1 Context

This pattern applies to organizations using routers to connect internal LANs to the Internet through a WAN link and intend to secure routers to minimize the impact of a security violation. Since all data traffic from and to the Internet passes through these routers, a security violation could lead to network failures or even theft of data. Routers facing the Internet are typically known as edge or border routers.

4.3.5.2 Problem

How to secure routers that connect internal LANs to the Internet through a WAN link?

4.3.5.3 Forces

The following forces influence the solution:

- **Secure Connectivity:** The border router should provide routing traffic and allow both inbound traffic coming from the Internet to the internal network and outbound traffic coming from the internal network to the Internet.
- **Authorized access only:** The solution should take into consideration that border routers, as well as other devices, are not public devices and should only be accessed by authorized network engineering staff.
- **Extendability:** Solution should provide means to allow future growth, through the addition of new devices, while be consistent in terms of security.
- **Security:** The border zone router provides secure access to the Internet. The solution should take into consideration that border router is normally the device more exposed to threats.

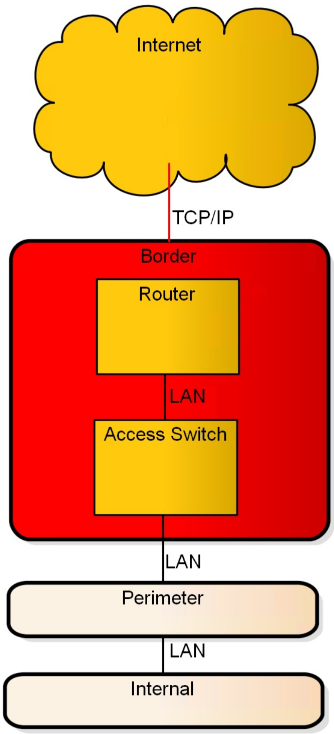


Figure 4.2: Border router

4.3.5.4 Solution

Apply tight, extended Access Control Lists (ACLs) on the border routers to secure the network traffic allowed in the perimeter network and create and maintain a security policy that identifies management activities.

The ACLs are important to restrict access from unknown, untrusted users, from the Internet into the internal network (figure 4.2). When defining border router ACLs, the following should be considered:

- Turn off unneeded services on the border router.
- Deny Internet Control Message Protocol (ICMP) to transit the border router, because support for the ping command and similar capabilities can lead to potential attacks.
- Allow Border Gateway Protocol (BGP) traffic that uses TCP/179 if there are packets sourced from adjacent routers.
- Deny all protocols, TCP/UDP ports and IP addresses through the router, except the protocols, ports and IP addresses corresponding to services in use.
- Block incoming packets that claim to have the same destination and source address.
- Deny tampered packets structured to appear from a different location. The anti-spoofing ACLs features ensure that tampered packets are rejected.

- Allow only internal traffic to enter the router from the internal network, and allow only Internet traffic to enter the router from the external network.

The management activities should be defined in a security policy. The security policy should include:

- Change standard users and administrator's names and passwords.

- Define multiple usernames and passwords for different levels of access. The policy should clearly identify who is allowed to log in to the router, who is allowed to configure and update it and who is allowed to read logs and other statistical information.

- Passwords should be complex by incorporating a mix of uppercase and lowercase letters and numbers and having the minimum length standards (for example, six characters minimum). More flexibility can be achieved by using a centralized secure login system such as RADIUS or Kerberos.

- Only secure clients that use the SSH protocol and from a specific network and source IP address can establish router console sessions.

- Turn on the router's logging capability, ensuring that includes time information, and use it to log errors and blocked packets.

- Having offline master copies of border router configuration files simplifies the process of identifying changes and the introduction of new devices with similar configurations.

4.3.5.5 Consequences

The use of **BORDER ROUTER SECURITY LOCKDOWN** pattern presents the following benefits:

- **Efficiency:** Disabling unneeded router services and restricting the packets passing through the router turn devices more efficient since they have less packets to process.

- **Access Control:** Only well-defined users and administrators can perform router management activities.

- **Simple configuration:** Having offline copies of router configuration can simplify the process of deployment of new devices or device reconfiguration, due to a malfunction, for example.

- **Reduce risk of attack:** Locking down border routers devices and actively controlling and monitoring those devices can minimize the risk of attack.

On the other hand, the use of **BORDER ROUTER SECURITY LOCKDOWN** pattern carries several liabilities:

- **Less flexibility:** The deployment of services requiring new ports or protocols require changes in router configurations.
- **Management activities:** Since for security reasons the administration must be performed from specific locations, the time and effort required to perform a configuration is higher;
- **Denial of service:** The process of locking down router does not minimize the risk of a denial of service attack.
- **Expensive:** The use of a centralized secure login may require the acquisition of new systems.

4.3.5.6 Related Patterns

This pattern is related with the following patterns:

- **TIERED DISTRIBUTION:** Tiered Distribution organizes the system infrastructure into a set of physical tiers, to optimize server environments for specific operational requirements and system resource usage.
- **BORDER SWITCH SECURITY LOCKDOWN:** similar to border router security lockdown this pattern defines the security measures that should be applied to switches in the border zone, to maximize security.

4.3.6 ITI Pattern - FAULT TOLERANT SERVER

4.3.6.1 Context

The definition of the required server hardware to support a new service, application or solution is a common problem for IT professionals, since it is a challenge to find the right hardware balance. For instance, over-utilized servers result in performance degradation and loss of end user productivity, while under-utilized servers can cause higher capital and operations expenses. The problem is even bigger when the application is critical for business.

4.3.6.2 Problem

Which hardware components should a server have to support business critical applications?

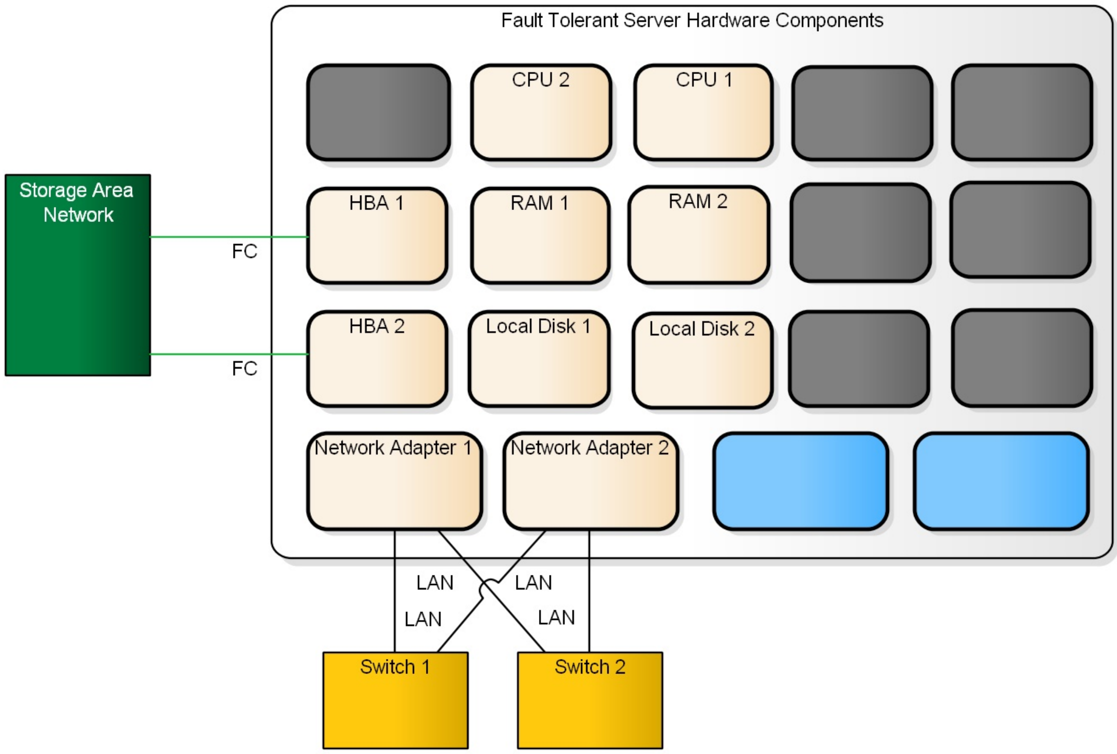


Figure 4.3: Fault tolerant server hardware components

4.3.6.3 Forces

The following forces influence the solution:

- **Failures:** The failure of a single component should not affect server availability. The server should continue supporting services, applications or solutions.
- **Scalability:** The server should be designed to allow scaling up (e.g. adding more resources, such as memory).
- **Performance:** The server should be performant to support business critical applications.

4.3.6.4 Solution

Use a server with redundant hardware components to minimize the impact of failures and provide free slots, to allow the addition of new hardware components.

The server should have the following hardware components:

- **Multiple CPUs:** In terms of CPU speed, since most applications are single-threaded, the number of CPUs is less important than the speed of CPUs. The faster the CPU, the more data will be able to process in a set period of time. The benefits that faster CPUs can provide to database systems, for instance, are obvious. However, if a server has multiple slower processors, the impact of CPU speed on database systems performance can often be mitigated.

- **Memory:** The service, application or solution requirements are also the main factors to define the right amount of memory. In what concerns memory, and despite the fact that there is a correlation between number of clients to support and the total server memory, there is no ratio defined to identify the maximum amount of memory required by a server. For instance, the amount of memory for a database system depends upon the previous factors and also the demands of the databases. The best practice is to configure the server with a reasonable amount of memory (depending on the requirements), and monitor whether memory is sufficient to address the server workload.

- **Network Adapters Speed:** Forgetting to define requirements for network adapters is a common mistake in the design of ITI servers, since they have a significant influence on the server performance. Having the right amount of memory and faster CPUs cannot eliminate the bottleneck introduced by not having fast network adapters. Sometimes, organizations perform investments to improve the performance of server through acquisition of faster processors or more memory and afterwards they realize that performance is still poor. Often the performance problems rely in network interface cards (NICs) and not with the amount of memory or CPUs. Having a Gigabit NIC is normally recommended for most applications since these cards offer performance gains that are higher than the 10x throughput, for example.

- **Multiple Network Adapters:** The use of multihoming (multiple network adapters) servers permits the use of a teaming network[3]. In the later, a single server can have two or more network adapter ports, each connected to a physically different switch, as presented in figure 4.3. In case of failure of one adapter, the server continues to work. Other advantages are the higher performance through the separation of traffic (e.g. traffic to Internet and traffic to internal servers) and increase of security with each interface connected to a separated network segment, which facilitates for instance, port filtering. The disadvantages of using multihoming servers are the additional complexity, and cost associated with multiple network adapters, switches and cables.

- **Storage:** Most applications, services or solutions also require storage to maintain information, images, files and media among other things. The most common requirement is database storage. The space required can normally be segregated into the space required for database (e.g. client data), space required for processing, file storage and sometimes space required to backups to disk. When planning disk size,

[3]teaming network or NIC teaming, also known as load balancing and failover Load Balancing and FailOver (LBFO), allows multiple network adapters on a computer to be placed into a team for bandwidth aggregation or traffic failover to prevent connectivity loss in the event of a network component failure.

the plan should also allow for data growth over time. Disks sized only for the initial data may quickly outgrow. Other important aspect besides space is performance, often measured in terms of throughput. The throughput can be defined as, how many requests the server can process per second. More users normally require higher throughput. In order to handle the required throughput, multiple disks and disk arrays should be used or multi-channel controller cards. It is recurrent to have servers with enough memory and CPUs and with poor performance, due to disk configurations. Since most applications require databases, there are two main components that require attention, when discussing disk performance for database servers, which are the database itself and database logs that should be in different disks. The common configuration to address this performance problem is to have at least three spindles, one for the operating system, one for database logs, and one for the database itself.

4.3.6.5 Consequences

The use of **FAULT TOLERANT SERVER** pattern presents the following benefits:

- **Availability:** The failure of most components does not affect business critical application. There are some server components (e.g. mouse, keyboard, and monitor) where a failure does not produce any direct effect in server availability.
- **Scalability:** There is enough room to scale, since there are free slots for CPUs, memory and network adapters, among other components.
- **Applicability:** The unique characteristics of this server make it well suited for business application front-end servers, database servers, authentication and authorization servers, among other systems.

On the other hand, the use of **FAULT TOLERANT SERVER** pattern carries several liabilities:

- **Capacity Planning:** Depending on the requirements of the business application, service or solution, the number and speed of hardware components may vary. The general recommendation is to use recent hardware components.
- **Single Points of Failure:** There are certain components (e.g. motherboards) that are a single point of failure in a server since a single server cannot have more than one of these components.
- **Failure impact:** The failures of a component can degrade performance.
- **Multiple failures:** The failure of multiple components simultaneously can affect server availability.

4.3.6.6 Related Patterns

This pattern is related with the following patterns:

- **LOAD BALANCING SERVER FARMS:** Load balancing server farms can improve application performance for the current number of users by sharing the workload across multiple servers.
- **FAILOVER CLUSTER:** Failover clusters can increase availability by creating redundancy in the infrastructure

4.3.7 ITI Pattern - REMOTE SOFTWARE DISTRIBUTOR

4.3.7.1 Context

This pattern applies to organizations that distribute software from a central place and have hundreds or even thousands of devices (e.g. servers, desktops, laptops, etc.) connected with local area networks (LANs) and wide area networks (WANs) in multiple remote locations. WANs are used to connect LANs and other types of networks together, to allow devices in one location to communicate with devices in other locations.

4.3.7.2 Example

Suppose a medium size bank institution has 1 central location (headquarters) and approximately 1200 remote locations (branch offices) with an average of 10 devices per location. In order to have a centralized administration, the devices in the central location and in each remote location are connected with LANs. The LANs in each remote location are connected with the central location with 256Kb WANs. The WANs communication links support all the management activities (e.g. software distribution, antivirus updates, etc.) from the central location. In each remote location, there is a critical business application called Financial Terminal, responsible for performing all financial operations and uploading data to the central location at the end of the day. Mainly because of this, the WAN communication link availability, speed and available bandwidth are very important. In such a scenario, for instance the distribution of 20MB of updates for each of the 10 devices in the remote location, corresponds to 200MB, which will take approximately 2 hours to complete, assuming 256Kb WAN communication link for each of the 1200 remote locations.

4.3.7.3 Problem

How do you distribute software to devices in remote locations?

4.3.7.4 Forces

The following forces influence the solution:

- Limited bandwidth – Communication links have low bandwidth available. The software distribution activities should minimize the link usage.
- Management from a central location – The software distribution activities should be performed from a central location.
- Devices in multiple locations – The organization devices are spread over different locations.
- Automation – The software distribution must be automated.
- Existing applications – Software distribution activities should not affect other applications running in remote locations.
- Fast distribution – The software should be distributed as fast as possible to devices in remote locations.

4.3.7.5 Solution

In each remote location deploy a Remote Software Distributor that receives software packages from the central software distribution solution and transfers them to all local devices as shown in figure 4.4.

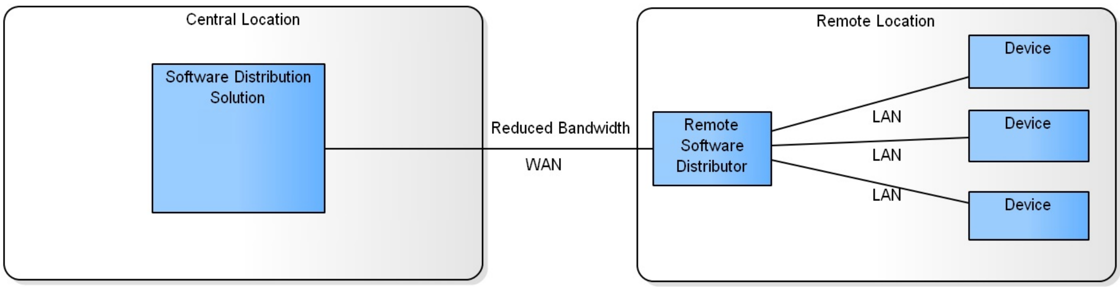


Figure 4.4: Devices pull/push software packages

Using Remote Software Distributor, software distribution packages are created in the central location and are distributed once from the central location to the Remote Software Distributor over the WAN link. Then, the software is transferred to the devices using the LAN. The complete solution encompasses the following components:

- Software distribution solution: Represents the main server(s) responsible for managing software distribution activities such as creating software distribution packages, transferring software to devices, receiving status regarding software distribution success, among other activities.

- Devices: Each server, desktop, laptop or other equipment that runs one or more applications and can be managed (e.g. receive software updates) remotely. Normally these devices have a piece of software running, called agent, responsible for allowing remote activities such as software distribution.
- Remote Software Distributor: Component that separates software distribution solution from devices. It acts as an intermediate between these two entities to provide efficient use of resources in operations such as software distribution, hardware and software inventory, among others. This component resides in a remote location.

4.3.7.6 Consequences

The use of Remote Software Distributor pattern presents the following benefits:

- Less effort – The team responsible for software distribution will have significantly less effort to distribute software when compared to manual software installation in each of the remote devices.
- Central – All the software distribution is performed from a central location.
- Fewer impact – The impact of software distribution in applications or services already deployed in the remote location that may require the availability of the link to work properly are less affected.
- Less Traffic – The bandwidth required to distribute software is significantly lower.
- Faster distribution – Significantly less time to distribute software to remote devices over a limited or reduced link.

On the other hand, the use of Remote Software Distributor pattern carries several liabilities:

- More servers – The number of servers to be deployed can be considerable higher, depending on the number of remote locations.
- Higher costs – More servers may also require changes in the existing licensing agreement.
- Higher maintenance – Since there are more servers to manage, the regular management activities such as backups, patch management, etc. will increase.
- Required logistic conditions – Some remote locations may not be prepared to have servers. Typical problems that may arise are unavailability of power supply, lack of networking equipment to support a server or simply space constraints.

4.3.7.7 Example Resolved

The use of the Distributor in each of the remote locations connected with 256Kb communication links allowed the software distribution from the central location to remote locations faster with less traffic and consequently less impact to the applications running in the remote location. By applying this pattern, the software distribution of 20MB of updates for each of the 10 devices in the remote location took approximately 10 minutes instead of 2 hours.

4.3.7.8 Related Patterns

The **ITI REMOTE SOFTWARE DISTRIBUTOR** pattern is related with the following patterns:

- **LOAD BALANCING SERVER FARMS:** Load balancing server farms can improve application performance for the current number of users by sharing the workload across multiple servers.
- **FAILOVER CLUSTER:** Failover clusters can increase availability by creating redundancy in the infrastructure

4.3.8 ITI Pattern - ITI BORDER ZONE

4.3.8.1 Context

The ITI constitutes the foundation upon which organizations can deliver services to customers, partners and its employees. The network is part of ITI and combines a number of devices (e.g. routers, switches, load balancing devices, firewalls, etc.) that should be able to communicate in an efficient, reliable, and secure manner to provide high quality services. The border zone is the zone of the ITI that provide the connection to external networks such as the internet.

4.3.8.2 Problem

How to connect to external networks to efficiently deliver services for customers, partners and employees without compromising the security?

4.3.8.3 Forces

The following forces influence the solution:

- **Secure connectivity:** The outbound traffic should be controlled by the routers and the inbound traffic will be allowed only to a specific set of services, depending on the needs of the organization.

- **High availability:** All components should be redundant in the border zone network to increase availability in case of hardware failures, or problems with a specific internet service provider.
- **Communication:** The ITI border zone has to provide communication among devices in an efficient, reliable, and secure manner securely.
- **Threats:** The border zone is the most exposed zone to threats.

4.3.8.4 Solution

Security is a responsibility of all components of ITI and not only network. Usually the answer consists in the application of defense in depth strategy (multiple layers of defense) to minimize the risk of being compromised. From a network perspective, this concept can be mapped into different security zones being the border zone the more exposed.

To provide service in an efficient way to all the end-users without compromising the security, define a zone called border to control all the connections with external networks. Deploy redundant routers and switches in the border zone and apply security lockdown configurations as defined in the ITI patterns BORDER ROUTER SECURITY LOCKDOWN and SWITCH SECURITY LOCKDOWN.

The number and type of network segments in the border zone are important for a variety of reasons such as to provide connectivity, improve performance through broadcast traffic containment, improve security through flow control among other aspects. The network segments can be physical (LANs) or logical (VLANs), but since the border zone is considered the most exposed to threats it is recommended to deploy physical network segments with no connectivity between them. Even if an attacker breaks one physical device, this does not mean that he can break the others.

The connections from the border zone to the Internet should be redundant (multiple internet service providers) and all paths among the components of the border zone should also be redundant. The routers should be configured to allow outbound from internal network to the internet through proxy server and will be responsible for routing traffic from the internet to the organization.

4.3.8.5 Consequences

The **ITI BORDER ZONE** provides the following benefits:

- **Security:** Using physical network segments has the advantage of being simpler to setup and more secure in the sense that are multiple physical networks.
- **Increased capacity:** Since for security reasons the number of devices are duplicated in the border zone, the number of requests that can transverse the border zone is higher.

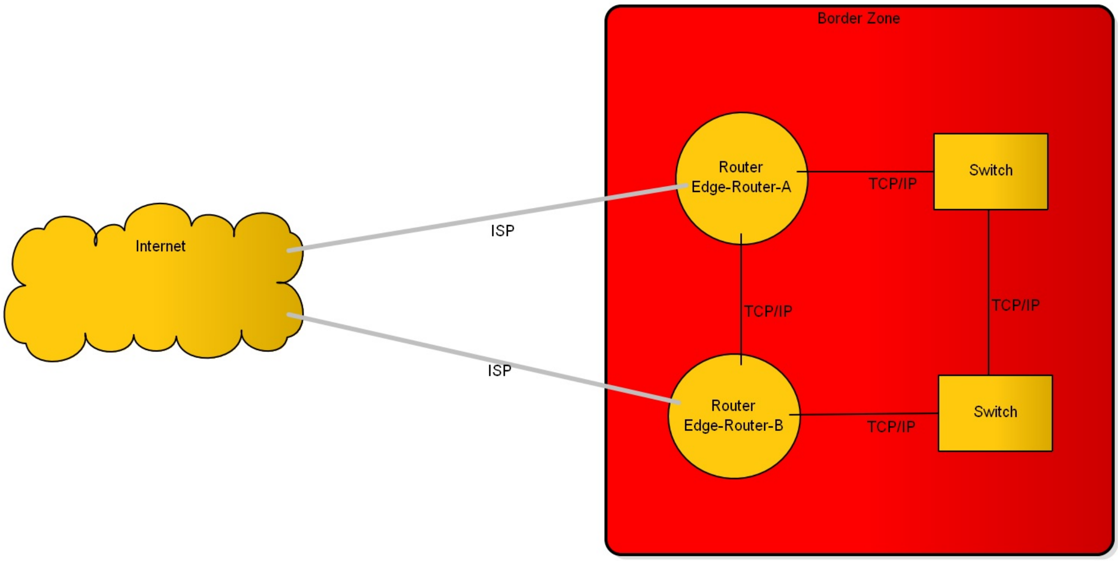


Figure 4.5: ITI border zone

- **High performance:** This solution offers higher performance and enables organizations to plan for potential future growth through the addition of new devices.

On the other hand, the pattern carries several liabilities:

- **Effort to manage:** The use of multiple physical networks, increases the effort to manage networks (since they are more), and in the case of a change in the topology, it will be more difficult to reconfigure.
- **Network analysis effort:** Depending on the number of services and applications the network performance analysis may take some time.
- **Expensive:** Since for security and high availability reasons, the networks are physical and all devices in the border zone should be redundant, an increase in the cost is expected.

4.3.8.6 Related Patterns

- **BORDER ROUTER SECURITY LOCKDOWN:** This pattern defines the security measures that should be applied to routers in the border zone, to maximize security.
- **BORDER SWITCH SECURITY LOCKDOWN:** similar to border router security lockdown this pattern defines the security measures that should be applied to switches in the border zone, to maximize security.

4.3.9 ITI Pattern - ITI FAULT DOMAIN

4.3.9.1 Context

This pattern applies to organizations that embrace cloud computing and require that an internal ITI have attributes such as scalability, elasticity, and fault-tolerance to support applications requiring high degree of availability. One of the main differences between traditional datacenters and private clouds reside in the abstraction of physical resources (e.g. servers, disks, networks) which are logical grouped and mapped to the physical ITI to allow efficient and intelligent provisioning and management.

4.3.9.2 Problem

How to protect against hardware failures and increase application availability?

4.3.9.3 Forces

The following forces influence the solution:

- **Failures:** The failure of hardware components such as disk crash, network fault or server death should not affect application availability.
- **Scalability:** The solution should be designed to allow scaling out.
- **Performance:** The solution should be performant to support business critical applications.

4.3.9.4 Solution

Define multiple units of failure, by organizing a set of hardware components (e.g. servers, switches, routers) that share a single point of failure. As an example, a rack of servers as represented in figure 4.6 can be considered a fault domain, since all servers connected to the same rack and a power outage in the rack will affect all servers in the rack.

After the creation of multiple ITI fault domains, it is important to distributed the data and different instances of the application to at least two ITI fault domains. This logical organization ensure that an application remains available even if a server or the whole rack hosting an application instance fails.

4.3.9.5 Consequences

The **ITI FAULT DOMAINS** provides the following benefits:

- **Availability:** The failure of servers within a fault domain does not affect applications.
- **Scalability:** More fault domains can be added to the farm to provide more capacity.

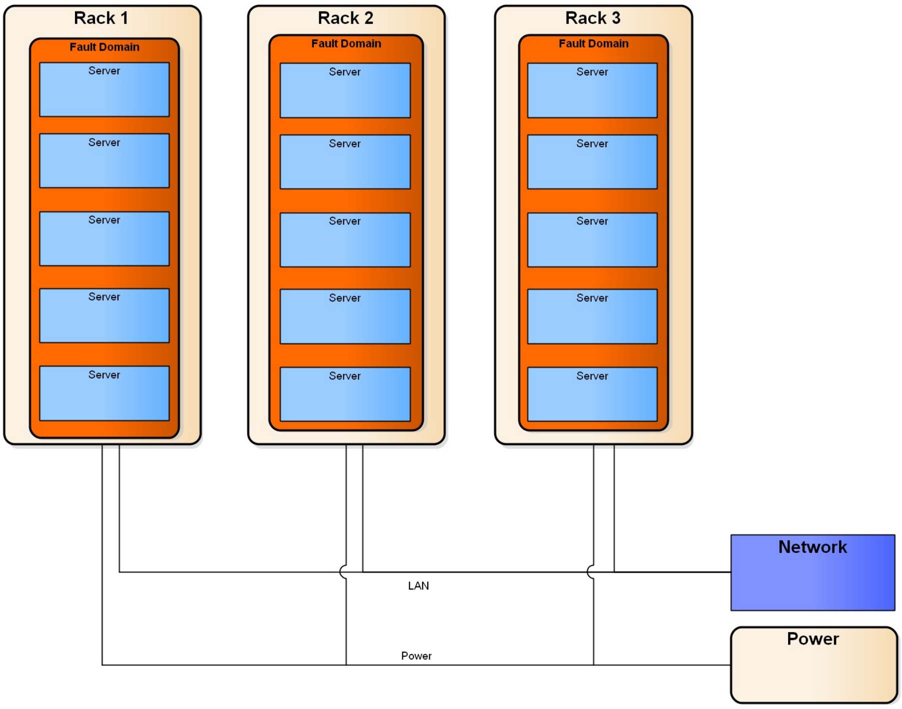


Figure 4.6: ITI fault domain

- **Minimize failure impact:** By separating an application into logical instances the failure of hardware is minimized.
- **Applicability:** The unique characteristics of this farm make it well suited for multiple business critical application such as front-end servers, database servers, authentication and authorization servers among others.

On the other hand, the pattern **ITI FAULT DOMAINS** carries several liabilities:

- **Expensive:** The creation of a fault domain when more capacity is required is expensive.
- **Single Points of Failures:** Within a fault domain, there is no fault tolerance. The fault-tolerance is only applicable when more than one ITI fault domain are managed as a whole.
- **Complexity:** The creation of ITI fault domain increases ITI complexity.
- **Multiple failures:** The failure of more than one ITI fault domain could compromise application availability. It is assumed that only one fault domain will fail at a time.

4.3.9.6 Related Patterns

The **ITI FAULT DOMAINS** pattern is related with the following patterns:

- **ITI UPDATE DOMAINS:** The ITI update domain provides the possibility to update subset of servers within a farm without compromising application or solution availability.
- **FAULT TOLERANT SERVER:** The fault tolerant servers present the characteristics that a server must possess to be fault tolerant.
- **LOAD BALANCING SERVER FARMS:** Load-balanced clusters can improve application performance for the current number of users by sharing the workload across multiple servers.
- **FAILOVER CLUSTER:** Failover clusters can increase availability by creating redundancy in the infrastructure.

4.3.10 ITI Pattern - CLOUDTRAFFIC ESTIMATOR

4.3.10.1 Context

The level of connectivity to access a public cloud is crucial to make the cloud deliver the best services. Different organizations have different sizes, require different services and have different needs. Depending on the bandwidth of the internet connection and the amount of data exchanged with the cloud provider, the experience for end users may vary. Most organizations simply assume they need a broadband IP VPN, whether or not the latter delivers the performance, reliability, availability and security required to access the public cloud.

4.3.10.2 Example

An organization has several applications hosted in internal ITI and an internet connection to provide employees access to the internet and to some internal applications, such as corporate e-mail that need to exchange messages with other corporations via internet and "customer facing" applications. There are also some applications developed in-house which are business critical and heavy utilized during some periods. Mainly to reduce computing costs, achieve a more flexible computing environment and ensure capacity is there when needed, this organization decided to embrace Cloud Computing and evaluate the impact on internet connection bandwidth of moving some of these applications to a public cloud.

4.3.10.3 Problem

How to estimate the required internet bandwidth when moving applications to a public cloud?

4.3.10.4 Forces

The following forces influence the solution:

- **Traffic variability:** The internet traffic produced by users, applications or network devices, may vary according to a period of the day, week, and month, what influences the required bandwidth.
- **Multiple purposes:** The internet connection is shared between users and applications and network devices. The available bandwidth has impact in the user's experience.
- **Capacity:** The number of users, applications and network devices influence the internet bandwidth requirements. More applications in cloud tend to require more internet connection bandwidth.
- **Operations:** The internet bandwidth requirements are influenced by the type of operations performed. Different users, applications or network devices perform different operations and may require different bandwidth requirements.

4.3.10.5 Solution

Evaluate network performance by using traffic generators, network analyzers and active measurement tools and create a network capacity plan.

The network capacity plan is an important instrument to define which services and applications can be migrated to the cloud and what will be the impact on the existing infrastructure. The network capacity plan should be integrated with the cloud adoption strategy, which has detailed information regarding business objectives, effort, business impact and cost analysis, risks, among other aspects. The general objectives of the network capacity plan are:

- Understand current services and applications network capacity requirements.
- Document assumptions regarding requirements and workload forecasts.
- Define the required network capacity forecast for services and applications.
- Provide network recommendations to ensure that there is sufficient network capacity to support the forecasted workload.

It is important that the network capacity plan includes the *(i)* definition of service level requirements that should categorize services and applications, quantify user's expectations, define workloads and identify service levels for each workload, the *(ii)* analysis of current capacity to understand services and applications requirements on internet connection bandwidth, and the (iii) planning for future capacity to forecast future needs and system requirements by determining future processing requirements to maintain the

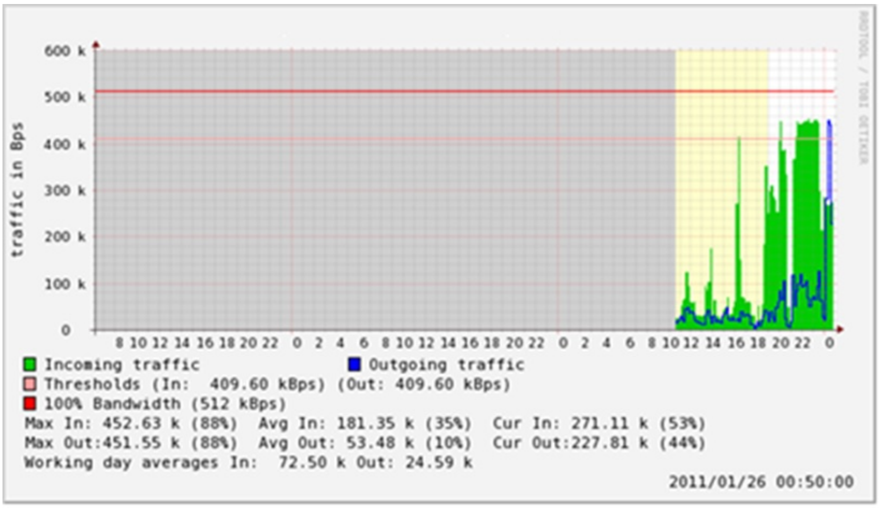


Figure 4.7: Incoming and outgoing traffic captured using network analyzers

service levels. To evaluate network performance, there are two types of tools that can help to analyze current capacity and plan future capacity which are *(i)* traffic generators and *(ii)* network analyzers. The traffic generators also called load generators are used to generate dummy packets and keep track of the packet delivery in the network and are useful to view and analyze the performance and capacity of existing devices, network topologies and internet connections. Network analyzers are important to provide more information regarding ITI internal applications and determine connection requirements needed to support users through a network traffic analysis. This analysis must capture incoming and outgoing traffic data to and from each application. A typical system to perform these analyses has three components: *(i)* sensor (or sensors), *(ii)* collector and *(iii)* reporting system. The sensor is also known as a probe, and is an agent that listens to the network and captures traffic data. The sensor may capture traffic from switches, routers and firewalls, among other devices. The collector is a server that receives and stores data from sensors. The reporting system is responsible for analyzing stored data and producing network traffic reports (figure 4.7).

Multiple networking factors can have impact and affect the delivery of data between the organization and the cloud. The network capacity plan should include details regarding latency, packet loss, retransmission and throughput, network devices performance (e.g. endpoints, routers, switches and hubs), bandwidth usage, type of circuit(s) used, upstream and downstream transfer rate, number of remote locations and their access to the cloud. To ascertain which workloads are the major, which help to narrow the attention to the workloads that are making the greatest demand on internet connection, the network capacity plan should identify, for each service and application, the following aspects:

- How many people are using the service or application.
- Current response time (e.g. excellent, fair, poor) from multiple ITI locations.

- Application or service peak usage (time of day, number of users).
- Type of service or application (e.g. e-mail, office, voice, video-conferencing).
- Application or service usage patterns (some applications have seasonal workloads such as end of week, end of month.).
- Impact on application or service in the internet connection bandwidth.
- Devices used to connect to services or applications.

Due to the demand on internet connection, the internet access to address cloud requirements is frequently dedicated, high-end with low latency and high bandwidth. It is normally recommended to establish a service level agreement between the organization and the internet service provider to define acceptable services. The service level agreement should be defined in terms of networking performance metrics such as response time and throughput.

4.3.10.6 Consequences

The **CLOUDTRAFFIC ESTIMATOR** provides the following benefits:

- **Decrease Costs:** Organizations will have the opportunity to have an internet connection bandwidth according to their needs.
- **Cloud Strategy:** Identify services and applications that consume most network resources. Understand how much bandwidth each service and application is using, before moving to the cloud, helps in the cloud strategy definition.
- **Improve Quality:** Identify what will be the services and application requirements bandwidth to provide the expected service quality. These requirements are important to help in the definition of service level agreements and to have a reliable and optimized connectivity.

On the other hand, the pattern carries several liabilities:

- **Time Consuming:** Depending on the number of services and applications the network performance analysis may take some time.
- **Increase Costs:** Based upon the network capacity plan analysis it may be necessary to have a different type of internet connection or a bandwidth increase.
- **Point in time analysis:** New services and application are implemented *decommissioned*, so the network analysis is a “current point in time”.

4.3.10.7 Example Resolved

Based upon the capacity planning analysis, the organization was able to understand the impact that existing services and applications have in the internet connection and the amount of traffic associated with each application deployed in ITI (that can benefit from being in a public cloud). The amount of traffic associated with each application was very important to classify applications according to their internet connection requirements and better evaluate where the public cloud delivered strategy provides the most benefits and gives the highest return value.

4.3.10.8 Related Patterns

The **CLOUDTRAFFIC ESTIMATOR** is related with the IT Infrastructure Patterns for Cloud Computing presented in figure 4.8.

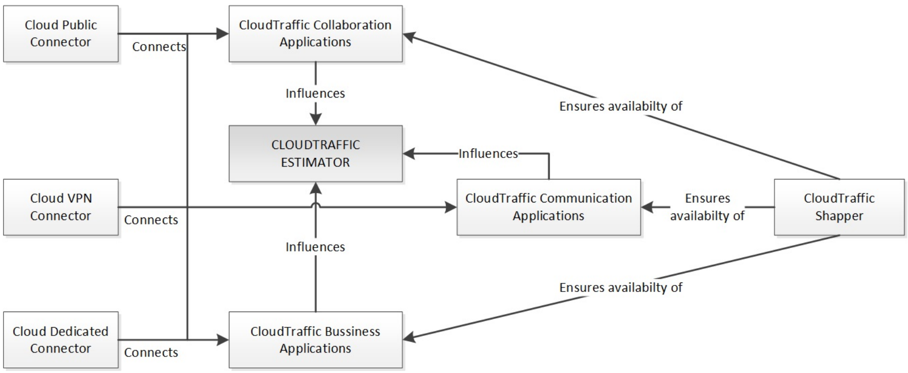


Figure 4.8: Related ITI design patterns for cloud computing

4.4 ITI Patterns Methodological Support

As mentioned before, design patterns play an important role in capturing expertise and making the design process simple, faster more efficient and accessible to everyone, because even experts do not solve problems from scratch. Instead, they tend to reuse solutions that have worked for them in the past.

As the benefits of design patterns become more and more evident, the number of design patterns tend to growth and one of the challenges is how to make this knowledge accessible in a way that can easily be used. Most Computer Aided Software Engineering (CASE) tools support the use of UML but do not support design patterns to simplify and assist in the creation of ITIs. As we did with the profile for modeling ITIs, we decided to use the same approach and extend Sparx EA to incorporate the ITI patterns instantiation process.

4.4.1 ITI Patterns Creation

For Sparx EA a pattern is seen as a group of objects/classes abstracted from a general set of modeling scenarios that describe how to solve an abstract problem. The ITI patterns are essentially XML files that can be imported as UML resources.

The process of creating an ITI Pattern on Sparx EA, comprises a number of steps as follows:

1. Create a standard UML Diagram.
2. Create a Model of the ITI representing the Pattern, using ITI constructs defined previously in **itiPM**. All stereotypes, tagged values and constraints should be added to the diagram.
3. When the model is finished, it has to be saved as a Pattern Diagram and the name of a XML file where the diagram will be saved has to be specified along with the Pattern Name, Category, Version and Notes.
4. Import the ITI Pattern into the resources by introducing the XML filename defined and pressing the import option.
5. Import the ITI Pattern into the ITI Patterns toolbox by selecting the appropriate toolbox diagram and adding the ITI Pattern. (Figure 4.9)

The ITI patterns have been created in a way that can be instantiated when designing ITIs as will be demonstrated in section 4.4.3.

4.4.2 Using ITI Patterns

The use of ITI patterns in Sparx EA enables the ITI designer to rapidly design ITI solutions using drag and drop functionalities. To be able to use ITI patterns in Sparx EA, the following steps must be taken:

1. Creation of an EA project or open an existing project.
2. Create a new diagram where the ITI Pattern will be placed or open an existing diagram.
3. Expand the toolbox ITI Patterns where all the ITI patterns are available and select the desirable ITI Pattern.
4. Drag and drop the pattern into the diagram as the example provided in figure 4.10.
5. Instantiating the ITI Pattern, by changing element names and adding or removing information from tagged values.

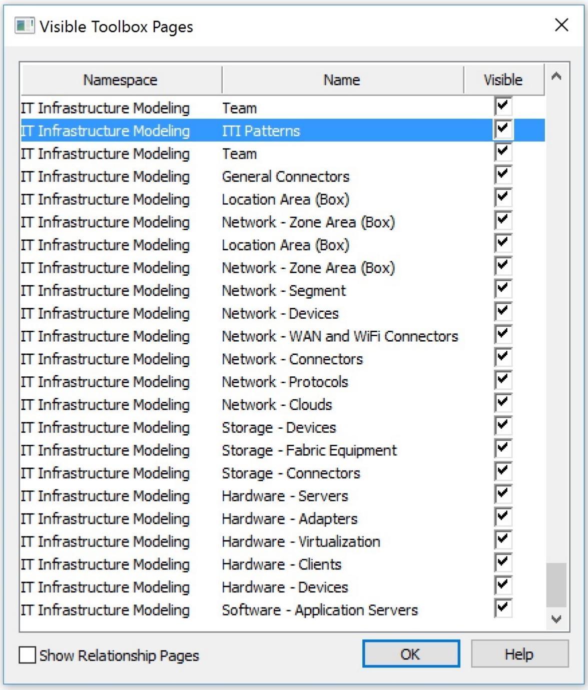


Figure 4.9: EA toolbox pages visibility

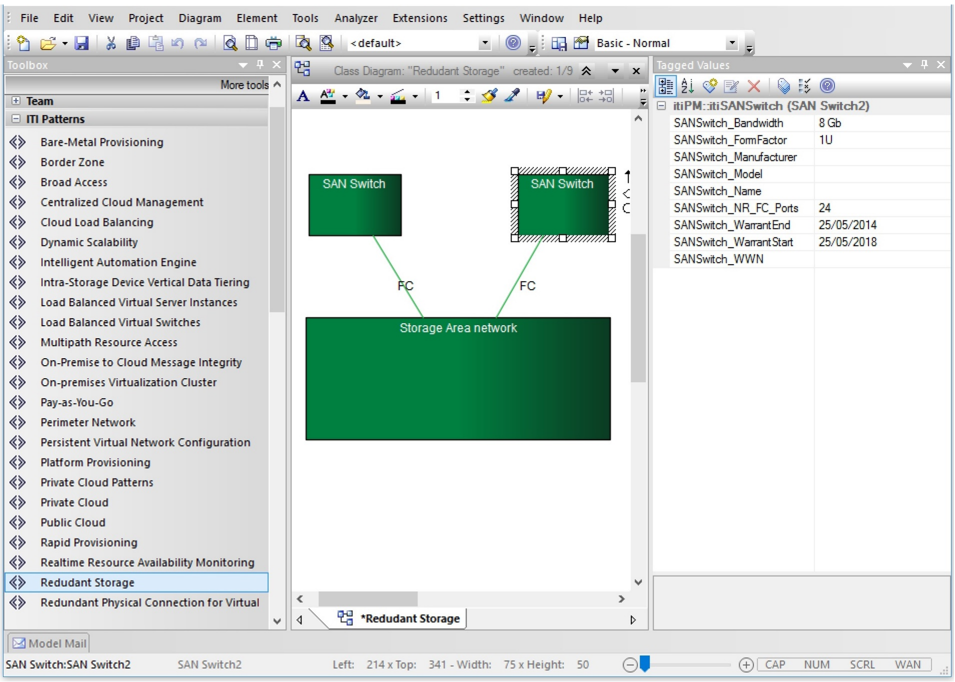


Figure 4.10: Drag and drop ITI patterns from toolbox

Note: To be able to preview the ITI Pattern, as well as their elements, types and names, the ITI Pattern can also be imported to the diagram using the UML resources as shown in figure 4.11.

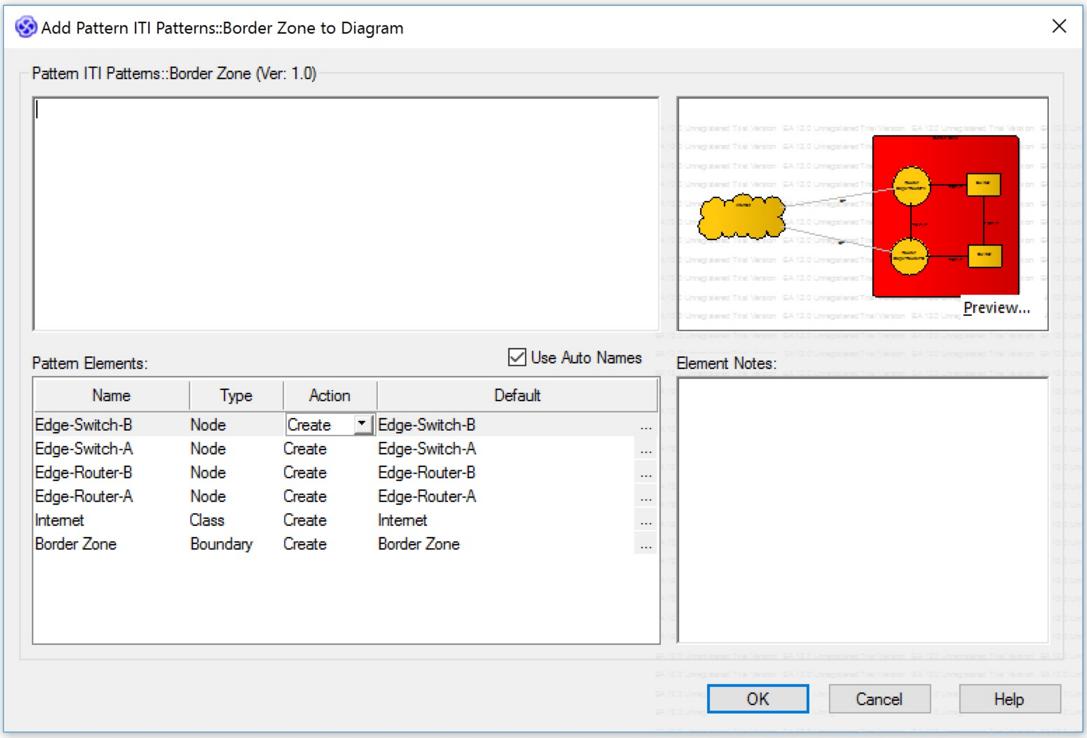


Figure 4.11: Add ITI pattern from resources

4.4.3 ITI Pattern Instantiation

As mentioned, the ITI patterns are technology agnostic, which means that it is possible to delegate the responsibility of designing ITIs to others than infrastructure architects, while decreasing the complexity and time required to design ITIs, among other benefits. This section shows how one of the ITI patterns can be instantiated to design a private cloud infrastructure consisting in two different physical servers, connected to a SAN with all components redundant. The pattern will be instantiated with technology from different manufacturers.

To demonstrate the applicability of our ITI patterns, we decided to use the ITI Pattern *High Availability (HA) Non-Converged Storage*. This pattern is frequently used when a storage investment in a SAN was already made and there is a requirement to deploy a new physical infrastructure.

All components of this pattern are redundant to provide high availability, and it is *Non-converged*, which means that storage network and LAN network paths are isolated, using dedicated I/O adapters. Fiber Channel is used as the primary connectivity method to the shared storage network, but the access can also be performed using iSCSI protocol (variant of this ITI pattern called *HA Converged Storage*). The servers are deployed in cluster, TCP/IP network adapters are configured with multiple network adapters, the access to storage is performed via multiple paths to achieve the failover and scalability requirements.

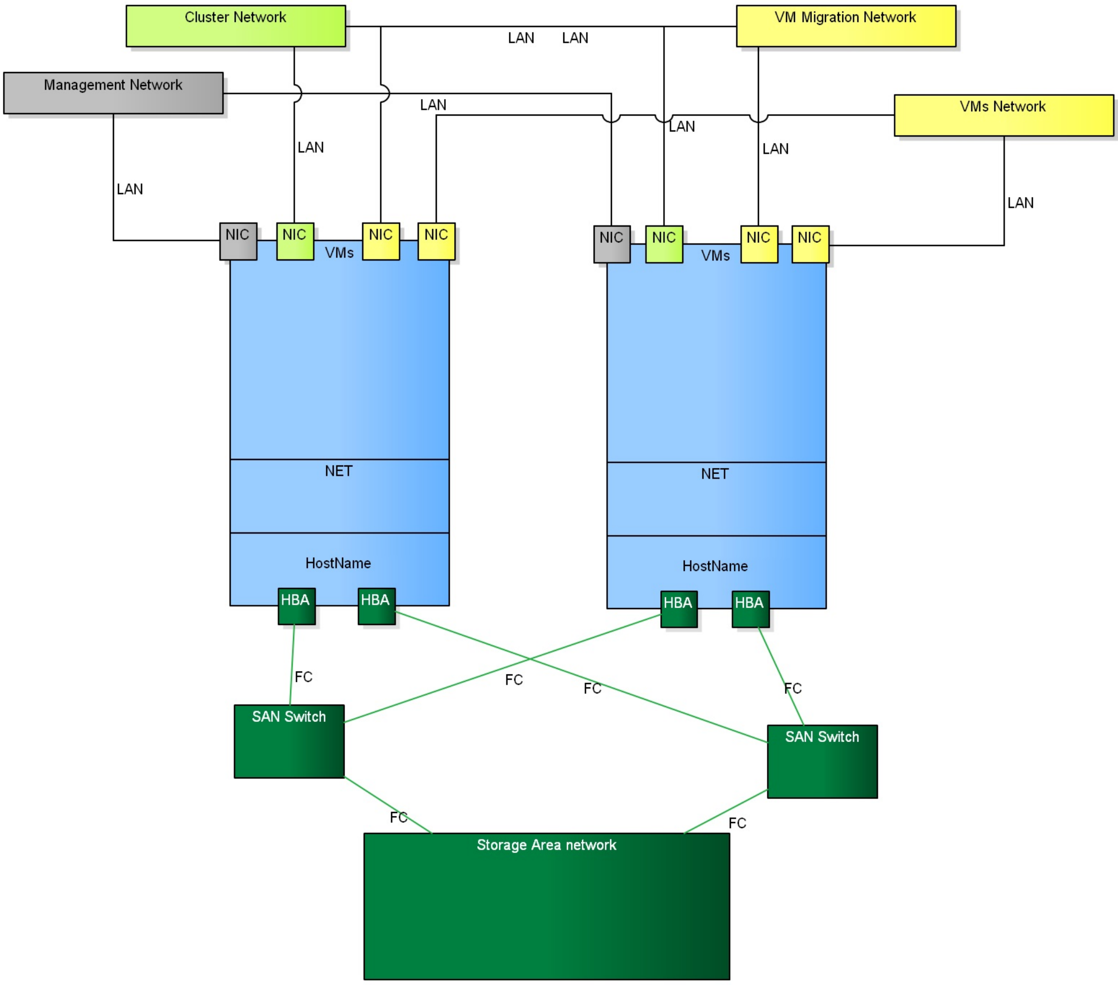


Figure 4.12: HA Non-converged storage

The diagram in the figure 4.12 is a private cloud and belongs to ITI pattern with the name *HA Non-Converged Storage*. It consists of two virtualization hosts that are configured in cluster to provide high availability. Both hosts connect to different networks using dedicated network adapters identified with different colors. Each host has capacity to have virtual network components such as *vSwitches* to provide internal connectivity and multiple virtual machines. In terms of storage, both hosts are connected to a SAN via fiber channel using multiple host based adapters and redundant paths.

As was mentioned before, ITI patterns are completely agnostic in terms of software, hardware, network or storage manufacturers allowing their instantiation with technology from multiple vendors. To demonstrate how the ITI pattern can be instantiated, we used only a limited number of software and hardware components from multiple manufacturers. The hardware and software specifications are available in tables 4.2 and 4.3. Hardware components are from *Dell*, hypervisor software from *VMware* and virtual machines from *Microsoft*. The diagram instantiated is presented in figure 4.13.

For understandability reasons, not all attributes and properties that can be included in

Table 4.2: Hardware specifications

Component	Description	Roles
Server	PowerEdge R710 server	Host virtual machines
FB/iSCSI Storage	EqualLogic PS6000XV	Provide shared storage
LAN switch	PowerConnect 6248 switches	Support LAN networks
SAN switch	PowerConnect 6248 switches	Support FB/iSCSI traffic
SAN	EqualLogic ps6210xs	Support FB/iSCSI traffic

Table 4.3: Software specifications

Component	Description	Roles
Hypervisor	vSphere ESXi 5.5	Hypervisor software on both hosts
vSwitches	Int. connectivity	vSwitch software on both hosts
Host 1 name	PowerServer01	Name of virtualization host 1
Host 2 name	PowerServer02	Name of virtualization host 2
Virtual Machine 1	VM in Host 1	Win. 2008 OS with Anti-virus
Virtual Machine 2	VM in Host 2	Win. 2008 R2 OS with DNS
Management Network	192.168.8.0	Host 1
Management Network	192.168.1.0	Host 2
Cluster Network	192.168.7.0	Host 1
Cluster Network	192.168.3.0	Host 2
VM Migration Network	192.168.6.0	Host 1
VM Migration Network	192.168.2.0	Host 2
VMs Network	192.168.5.0	Host 1
VMs Network	192.168.1.0	Host 2

the diagram are visible. One example is the details of virtual machines in terms of sizing, network configurations and operating systems that not visible in the diagram. In the figure 4.13 most of the configurations can be seen in the diagram including the vSwitch network component, virtual machines and applications, names of ITI components among other details.

4.5 Summary

This chapter is focused on design patterns and how they can be adapted and applied to ITIs. It starts with the motivation, a brief history of design patterns and how they can be organized in section 4.2.

Section 4.3, describes the process behind the creation of ITI patterns and provide the reasons that lead to the selection of Coplien pattern form to organize patterns. A catalog of ITI patterns with a brief description and their relations is also presented. This section finalizes with six different ITI patterns created and some of them published during this research work.

To make this knowledge associated with ITI patterns, accessible in a way that can

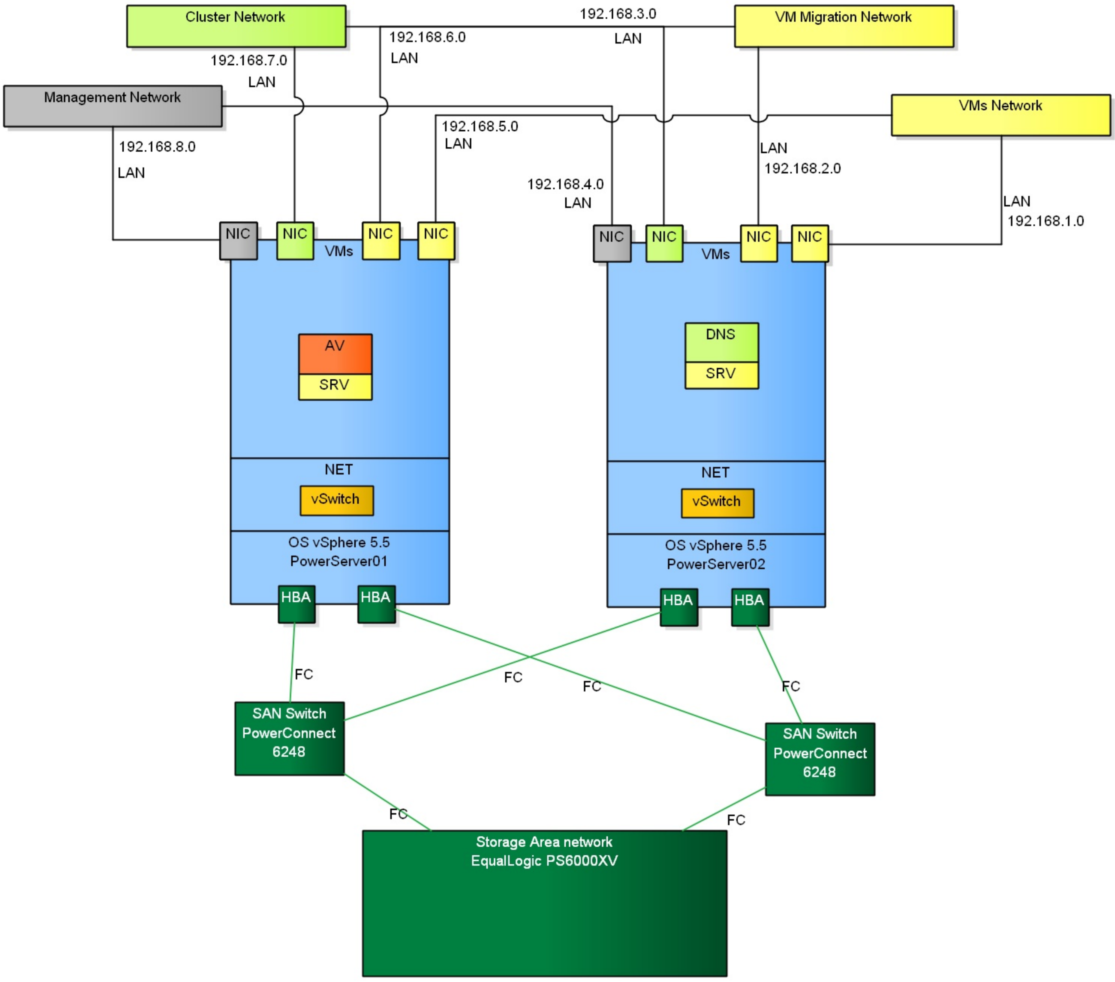


Figure 4.13: HA Non-converged storage instantiated

easily be used, it was decided to extend Sparx EA to incorporate the concepts of ITI patterns. The process of creation and use ITI patterns in Sparx EA is described in section 4.4. This chapter closes with the use of ITI patterns to support the design of a private cloud infrastructure consisting in two different physical servers, connected to a SAN with all components redundant. The pattern was instantiated with technology from different manufacturers.

CHAPTER 5

VALIDATION

Contents

5.1 Introduction

The starting point of our research process was to formulate the main problems we are tackling (section 5.2). This drove us to the research questions (section 5.3). To answer these questions, we had to formulate some hypotheses. The explanatory theory of the problems posed by the research questions is embodied in the research hypotheses (section 5.4). For validating our research contributions, we had to set up some experiments (section 5.6). Based on the experiments conducted, we were able to draw conclusions about the benefits and effectiveness of our proposals (section 1.2.3).

5.2 Research Problems

As briefly presented in section 1.2.1, the aim of our research consists in providing contributions for the following major problems[1] faced in the realm of ITIs design:

- RP1: Informal notation used to represent ITIs.
- RP2: Inability to express best practices in ITIs design.
- RP3: Absence of methodological support for a pattern-based approach to ITI design.

5.2.1 Informal Notation Used to Represent ITIs (RP1)

Our experience in the field has shown that most organizations depict their ITIs informally, either using some *ad-hoc* templates or informal notation, not supported by a standard or framework, resulting in ambiguous models without any kind of traceability. The main reason for such a poor state-of-the-art is the absence of a modeling language capable of representing ITIs at the required level of abstraction. In fact, candidate languages do not provide adequate modeling constructs. The use of informal modeling notations renders ambiguity, creates communicational problems among stakeholders and ultimately makes ITIs suffer the same problems of legacy software: undocumented decisions, redundancy, inconsistencies and increased cost of ownership.

5.2.2 Inability to Express Best Practices in ITI Design (RP2)

Unlike other more mature IT domains such as Software Engineering, where there are formal and systematic processes of capturing, storing and transmitting information (e.g. design patterns), the design of ITIs is based on proprietary product-oriented reference architectures and implementation guides (known as *blueprints*) defined by vendors to

[1] For the sake of traceability with research questions the research problems are labeled as RP1, RP2 and RP3

facilitate the configuration of a particular solution, based upon their services and products portfolio [HP12b; Tre+13]. The lack of provider independent solutions to support the ITI design process and the existence of *ad-hoc* models which are ambiguous to stakeholders, turn unfeasible the use of these models to express best practices in ITI design. Therefore, models are unable to play their role as first class entities in the design, deployment and maintenance of ITI activities.

5.2.3 No Methodological Support for a Pattern-based Approach to ITI Design (RP3)

ITI patterns are abstract descriptions of solutions to often recurring problems in the domain of ITIs. They are a means to communicate experience in ITI design. Despite the popularity with the use of design patterns in general, one of the problems identified is how to enable people to use them. In other words, having well-defined ITI patterns available during infrastructure design activities is not enough and a methodological support must exist to aggregate and simplify the use of ITI patterns during infrastructure design. In this context, the term methodological support is used to refer to computational support (tooling).

5.3 Research Questions

To address the research problems of section 5.2, the following research questions were formulated:

- RQ1: How to model ITIs to capture more information?
- RQ2: How to express and reuse ITI best practices?
- RQ3: Which methodological approach can we use to create a new ITI, based upon patterns?

The RQ1 is related with RP1, RQ2 with RP2 and RQ3 is related with RP3. Each of the above research questions is also related with the research hypotheses in section 5.4.

5.4 Research Hypotheses

This section includes the definition of hypotheses tests and the definition of variables used in the experiment. The hypotheses presented include a null hypothesis, denoted H0i, and its corresponding alternative hypothesis, denoted H1i. By testing the research hypotheses against empirical evidence, the intention is to validate the degree of accomplishment of our research work. As will be described later in this chapter, the experiment uses five different assignments available in appendix E.2, that were presented to experts in the area. To measure the results achieved with each experiment, there are three main dependent variables detailed in section 5.6.1.2, that will be responsible for measuring the completeness of the assignment (*Attributes coverage*), the time to execute each requirement correctly (*Design efficiency*) and the number of requirements correctly addressed (*Requirements coverage*).

5.4.1 Hypotheses About Modeling ITIs

To evaluate the impact of a concrete representation of the same language in the design of ITIs, the following null hypotheses and corresponding alternative hypotheses were defined:

H01: The usage of the proposed concrete representation during the ITI design process has no influence on *Attributes coverage*, when compared with *ad-hoc* methods.

H11: The usage of the proposed concrete representation during ITI design process has an impact on *Attributes coverage*, when compared with *ad-hoc* methods.

H02: The usage of the proposed concrete representation during the ITI design process has no influence on *Requirements coverage*, when compared with *ad-hoc* methods.

H12: The usage of the proposed concrete representation during ITI design process has an impact on *Requirements coverage*, when compared with *ad-hoc* methods.

The hypotheses Hi1 and Hi2 are related with RP1 described in section 5.2.1 and with RQ1 described in section 5.3.

5.4.2 Hypotheses About Best Practices Reuse

To validate the impact of the use of ITI patterns in the design of ITIs, the following null hypotheses and corresponding alternative hypotheses were defined:

H03: The usage of ITI patterns during ITI design process has no influence on the *Attributes coverage*, when compared with *ad-hoc* methods.

H13: The usage of ITI patterns during ITI design process has an impact on the *Attributes coverage*, when compared with *ad-hoc* methods.

H04: The usage of ITI patterns during ITI design process has no influence on the *Requirements coverage*, when compared with *ad-hoc* methods.

H14: The usage of ITI patterns during ITI design process has an impact on the *Requirements coverage*, when compared with *ad-hoc* methods.

The hypotheses Hi3 and Hi4 are related with RP2 described in section 5.2.2 and with RQ2 described in section 5.3.

5.4.3 Hypothesis About Methodological Support

To understand and validate the impact of methodological support we evaluate the following hypotheses:

H05: The use of the methodological support has no influence in the *Design efficiency*, when compared with *ad-hoc* methods.

H15: The use of the methodological support has an impact on the *Design efficiency*, when compared with *ad-hoc* methods.

The hypothesis Hi5 is related with RP3 described in section 5.2.3 and with RQ3 described in section 5.3.

5.5 Online Survey

This section introduces the survey that was used also to validate the results obtained. Surveys are probably the most commonly-used research method worldwide. A survey is a form of empirical study that can be used to collect different thoughts and information for a range of issues [Fin03; PK01]. Since survey-based research is not a straightforward process and can easily lead to wrong or inconclusive results, we decided to follow the guidelines for survey design, construction, administration and analysis outlined by Barbara Kitchenham and Shari Pfleeger [Kit+02; KP02a; KP02b; KP02d; PK01].

The other decision was to conduct the survey online to obtain more information based on the practical experiences of designing ITIs, based on the increasing number of virtual groups and communities online and with hundreds of thousands of people regularly participating in discussions about almost every topic.

There are several advantages in using online surveys. The main advantages are:

- **Efficiency and economy.** An online survey offers the possibility to collect data in far less time when compared with other type of surveys such as pencil and-paper surveys, telephone interviews or traditional postal surveys [Oat06].
- **Data Analysis.** The results are presented in a digital format which allows easier analysis of data.
- **Groups of interest.** With social networks, it is easy to find people with expertise in the area of ITIs.
- **Convenience.** The survey respondents can decide when and where to complete the survey.

There are also disadvantages in conducting online survey that should be considered:

- **Privacy.** The process used to communicate with potential respondents is frequently using discussion groups, chat rooms, or direct emails, that may be considered rude, offensive or Sending and Posting Advertisement in Mass (SPAM) [HB04] by the members of the online community and can lead the community owner or manager to delete the unwanted post.
- **Access.** The process to access some online communities requires approval, which can delay the whole process.

- **Sampling.** Most of the communities are based on the common interests for a topic and they do not require much information from the participants to join other than the email address and other demographic information. When the data is self-reported, there is no guarantee that the respondents provided accurate demographic information. There is also no guarantee regarding multiple responses from the respondents, and invalid/inactive email addresses make random sampling online a problematic method [And+03].

5.5.1 Survey Design

As stated before the design process is key and starts with reviewing the objectives that, in this case, consists in investigating the benefits of using ITI Patterns in the design of ITIs. This process also includes defining the target population. Approaches to obtain the information needed, a strategy to promote the largest possible response rate[Kit+02] among other steps that will be detailed next.

The purpose of the survey consists in understanding better the current practices in the design of ITIs, and assessing the impact of using the ITI profile and ITI patterns created in the process of designing ITIs. The purpose is supported by the hypothesis described in section 1.2.2. To obtain the information, and according to the survey guidelines that will be described in section 5.5.2 the survey is divided in the following six sections:

- **Section I - Introduction.** The survey begins by introducing the purpose of the survey and describing the survey structure and an estimate of the time needed to complete the survey, in a study letter that was followed by a consent letter.
- **Section II - Background.** The questions in this section are focused on understanding the current state of the art in the design of infrastructures.
- **Section III - Modeling IT Infrastructures.** In this part, we investigate the experience of respondents with the design of ITIs and the impact of the ITI Profile in the ITI design process.
- **Section IV - IT Infrastructure Patterns.** In this part, we investigate the impact of the ITI Patterns in the ITI design process.
- **Section V - Tools.** In this part, we asked the respondents to provide information regarding the tools that they use to design infrastructures.
- **Section VI - About you.** This section asks respondents for some personal information and give respondents the opportunity to comment about their experience with the design of Infrastructure as well as some comments on the questionnaire.

5.5.2 Constructing the Survey Instrument

The survey instrument in our case is a questionnaire, that according to [KP02b] should be constructed using the following steps:

- Search relevant literature.

- Building the questionnaire.
- Evaluate the questionnaire.
- Document the questionnaire.

The first three steps will be detailed next. Notice that the last step in the construction of the questionnaire is the documentation which is normally performed when the questionnaire is concluded [BF95]. Documenting the survey includes details regarding the objective(s) of the study, description of the evaluation process, who were the questionnaire respondents, how the questionnaires were administered and how the completed questionnaires were processed. This step is crucial when dealing with traditional surveys where there is the need to wait for a long time for the distribution of the questionnaires, the respondents to reply and the researchers to analyze the results and there the risk of forgetting details of instrument creation and administration. Since the questionnaire will be performed online and all the information is in the survey platform we skipped this procedure.

5.5.2.1 Literature search

As described in section 5.5, to understand if similar study were already performed, we searched for a SLR related with ITIs. Since we were unable to find, we decided to create a questionnaire to understand the state of the art on ITIs and validate the results.

5.5.2.2 Building the questionnaire

There are several guidelines that should be applied to questionnaires to maximize the number of answers. These guidelines were defined mainly for traditional surveys. However, many of them can be easily adapted to online questionnaires. To construct the questionnaire, we considered fifteen best practices rules adapted from traditional surveys [KP02d]. The same approach was performed to construct the questions, to obtain the information needed in the most effective way. The best practices rules to construct the questions were adapted from [ES14; KP02d] and we defined sixteen question guidelines.

Survey guidelines: To construct the questionnaire we considered the following **S**urvey **G**uidelines (**SG**$_n$ Where n = 1,2,3...), many of them adapted from [KP02d] to comply with online surveys:

- **SG1 - Instructions.** Instructions to complete each part of the questionnaire are clearly written.
- **SG2 - Demographic.** Information about the respondent are positioned at the end of the questionnaire.
- **SG3 - Length.** The total number of questions are appropriate.
- **SG4 - Purpose.** The purpose of the study is explained.

- **SG5 - Time.** The time to complete the questionnaire is realistic.
- **SG6 - Selection.** The respondents understand why they were chosen.
- **SG7 - Comments.** There is space to comment on the questionnaire.
- **SG8 - Layout.** The space between questions is enough to not confuse respondents.
- **SG9 - Split.** The questions are not split between pages.
- **SG10 - Vertical.** The vertical answer format was used to maximize clarity.
- **SG11 - Color.** The color or contrast used does not affect interpretation.
- **SG12 - Font.** The font type is easy to read, and a font size is 10-12.
- **SG13 - Motivation.** The respondents are motivated to answer.
- **SG14 - Objectives.** The answers obtained address the study objectives.
- **SG15 - Knowledge.** The target population has enough knowledge to answer questionnaire.

Questions guidelines: To construct the questions to the questionnaire we follow the rules outlined in [ES14; KP02d] and we created the following **Q**uestion **G**uidelines (**QG**$_n$ Where $n = 1,2,3...$):

- **QG1.** The question relates directly to objectives.
- **QG2.** Question can be answered easily and accurately.
- **QG3.** Respondents have sufficient knowledge to answer the question.
- **QG4.** If applicable the question, considers a timeframe.
- **QG5.** The question is part of a topic, that addresses a particular objective.
- **QG6.** The question can be answered with standardized response format.
- **QG7.** The question is purposeful so that the respondent can see relationship between the intention of question and the survey objective.
- **QG8.** The question is precise and unambiguous
- **QG9.** The question does not contain more than one idea (two-edged).
- **QG10.** The question uses conventional language.
- **QG11.** The question or statement is not negative.
- **QG12.** In close questions of type categorical the answers include "None of the above" or "Don't know".
- **QG13.** In close questions of type ordinal, the answers include a neutral response.
- **QG14.** The question is not unnecessarily long.
- **QC15.** The question is not a leading question that uses phrasing meant to bias the response.
- **QC16.** The question is not intrusive or impertinent.

The complete survey with the answers are available in section 5.5.5.

5.5.2.3 Questionnaire evaluation

Before administering the questionnaire and gathering the resulting data it was essential to evaluate it through pretesting. The pretesting main goals were to:

- Check that questions are understandable.
- Assess the likely response rate.
- Evaluate the reliability.
- Select appropriate data analysis techniques, using the pre-test data where adequate.

To perform the pretest, we used a pilot study of the survey, that consisted in the same procedures but administered to a smaller sample. The goal of the pilot was to identify problems with the questionnaire itself, as well as with the response rate and follow-up procedures. The pretest was also important to assess the reliability in terms of making sure that we get the same answers even if we administer the survey to a different group.

5.5.3 Survey Pilot (PreTest)

After the construction of the online questionnaire, we invited eight experts to assess the quality of the questionnaire. Based on [KP02d] we defined fifteen overall guidelines and sixteen question guidelines. The goal was to have as many comments as possible from the reviewers to allow us to improve the overall quality of questionnaire. The guidelines were described in section 5.5.2. We used the participating approach (we informed respondents that it was a pretest) and interviewed each expert and we ask them to think out loud to make sure that the questions were asking what we needed to know.

5.5.4 Population and Samples

To have a sample considered valid, we have to have more than a set of responses from the questionnaire. To obtain the sample, it is also crucial to start with the definition of the target population that in this research is the people responsible for designing ITIs.

Our goal was to eliminate the subjectivity and obtain a sample that is both unbiased and representative of the target population. To allow us to make statistical inferences from our data we used probabilistic sample methods. The method used was cluster-based sampling since the questionnaire was applicable primarily to a defined group. Additionally, because the target population is very specific, of limited availability and difficult to identify outside of the chosen group we used also a non-probabilistic sampling method called snowball because we also requested people who have participated to nominate other people with ITI design experience.

As will be detailed in section 5.5.5 we sent the survey to a LinkedIn group of infrastructure architects and to 545 authors of ITI papers. Our sample size was 123 responses, but only 117 were considered valid.

5.5.5 Survey Results

After completing the survey pre-test and improved the survey based on the feedback received, we decided to distribute the survey to the infrastructure architects group available in the LinkedIn with more than 12.000 members. We defined the strategy with the group moderators and to maximize responses we encouraged the respondents to share the survey link with others. We also mentioned that for those interested we could share the results and give each respondent access to the modeling tool.

Unfortunately, even with the help of group moderators we had only 20 responses and we had no access to people's private e-mails. Despite the number of posts published in the group, after a deeper analysis we concluded that the interaction among members was very low with multiple threads over the years with little or no response.

To increase the number of responses we decided to send a personalized e-mail message for all the authors of papers related with ITIs and published in 2000 or later. For each paper, the paper name, author and e-mail was extracted and then we used the mail merge feature to send the message for everyone. There were 545 different authors. We repeat the process every week for those who had not responded. We realized that the initial message was classified as SPAM or blocked by mail filters for some recipients and other users claimed that they did not trust the URL provided, and we had to contact them and sent a screenshot taken from the *Qualtrics*[2] platform.

We collected 123 responses in total and after removing invalid responses there were 117 responses in our database. Before analyzing the results, it was important to analyze demographic questions to better understand the conditions where the survey was taken and the results.

As mentioned before the platform used to create the survey was *Qualtrics* that has the possibility to have hidden questions with meta information to help tracking and report basic information about respondents, such as their operating system and browser. This information was gathered mainly for troubleshooting and understanding if the users that did not complete the survey all had the same configurations.

5.5.5.1 Section I - Introduction (Answers)

The survey supported the most common browsers (e.g. Chrome, Internet Explorer and Firefox) and worked in multiple operating systems to make sure that the experience was similar in all environments. The respondents could answer the questions on a device similar to a PC or on a mobile phone. More than 88% of the respondents used computers.

The physical location of respondents may be important when analyzing results. After evaluating the *qualtrics* platform we decided not to have direct questions asking for location since, there is a hidden field with the IP address information. We used this information only to detect and delete duplicate answers. Our survey was distributed globally,

[2] `https://www.qualtrics.com`

and we had answers from multiple locations (table 5.1). As can be seen in figure 5.1 most respondents were located in Europe.

Table 5.1: Locations where there were answers

Location	Flag	Location	Flag
Africa		**Europe**	
Algeria		Austria	
Morocco		Belgium	
Namibia		Bosnia And Herzegovina	
Asia		France	
India		Germany	
Malaysia		Greece	
Myanmar		Ireland	
Taiwan, Province Of China		Italy	
Middle East		Netherlands	
Saudi Arabia		Portugal	
North America		Romania	
Canada		Slovakia	
United States		Spain	
Oceania		Sweden	
Australia		Switzerland	
South America		Ukraine	
Brazil		United Kingdom	

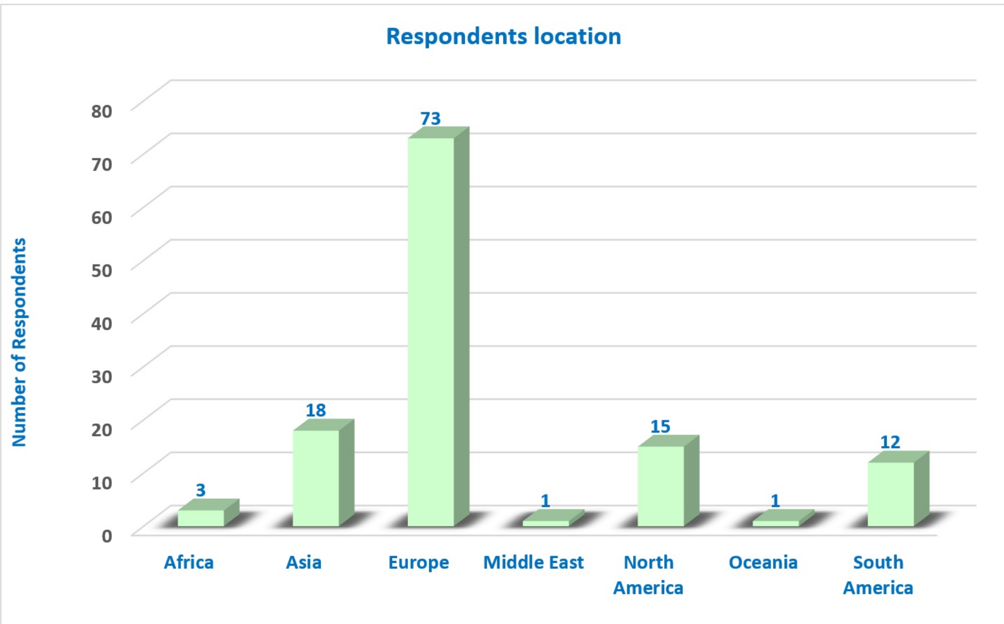


Figure 5.1: Respondents location

5.5.5.2 Section II - Background (Answers)

This section presents the questionnaire answers to a group of questions that we classified as background on ITIs.

Question 1: How effective are the current ITI representations to **STORE** or **CONTAIN** all the required ITI details/configurations?

Table 5.2: Q1 results: effectiveness of ITI representation

Effectiveness	%	Count
Completely Ineffective	9.6%	7
Ineffective	43.8%	32
Neither Effective nor Ineffective	27.4%	20
Effective	17.8%	13
Very Effective	1.4%	1

Most of the respondents confirmed that the existing practices to represent ITIs are not effective to store or contain all the configuration and details of ITI information, which is completely aligned with our expectations and confirmed our claims.

Question 2: How frequently do you **NEED ACCESS** to ITI representations (e.g. sketches, diagrams, pictures)?

Table 5.3: Q2 results: importance of ITI representation

Answer	%	Count
Never	4.2%	3
Less than Once a Month	11.1%	8
Once a Month	18.0%	13
2-3 Times a Month	22.2%	16
Once a Week	13.9%	10
2-3 Times a Week	15.3%	11
Daily	15.3%	11

The main purpose of this question was to understand how important the ITI representation are for the ITI professionals. Table 5.3 shows that the majority of ITI professionals need to access the ITI representations several times a week. These results indicate that the representations are important since they are used very often.

Question 3: According to your experience, **WHICH ROLES** are likely to create or adapt ITI representations (e.g. sketches, diagrams, pictures) ?

Table 5.4: Q3 results: roles creating ITI representation

Answer	%	Count
IT Architects	93.1%	67
Consultants	54.2%	39
IT Managers	43.1%	31
Developers	30.6%	22
Business Decision Makers	12.5%	9
IT Decision Makers	33.3%	24
IT Administrators	55.6%	40
Others (Please Specify)	9.7%	7

We claimed earlier that some of the ITI architectural problems may be related with low expertise on the ITI domain of some professionals. This question allowed us to confirm that there is a significant number of non ITI professionals such as developers, business decision makers and IT administrators representing ITIs.

Other roles creating ITIs (question 3)

- *Researchers*
- *Systems Engineers*
- *Pretty much all employees draw one of those at some point.*
- *Anyone with the need to gain knowledge on existing infrastructures and/or analyzing data flows within such infrastructure.*

Question 4: How do you classify the task of creating IT infrastructures representations with ITI components from **DIFFERENT VENDORS**, each with their own specifications?

Table 5.5: Q4 results: difficulty of representing ITIs

Answer	%	Count
Very Difficult	14.5%	10
Difficult	33.3%	23
Somewhat Difficult	27.5%	19
Neutral	18.8%	13
Somewhat Easy	2.9%	2
Easy	2.9%	2
Very Easy	0.0%	0

ITIs are made of components from different vendors which sometimes causes compatibility and integration issues. This question confirmed that most respondents face the same challenge.

Question 5: How do you classify the **OVERALL QUALITY** (in terms of completeness and consistency) of the ITIs representations (e.g. sketches, diagrams, pictures), that you have come across?

Table 5.6: Q5 results: ITI representation quality

Answer	%	Count
Very Low	8.3%	6
Low	38.9%	28
Fair	45.8%	33
High	5.6%	4
Very High	1.4%	1

Most respondents mentioned that the quality associated with existing ITI representations low or fair. The answers to this question were totally aligned with our claims related with the low quality of ITI representations.

Question 6: What levels of **FORMALITY** are normally applicable to the ITI representations (e.g. sketches, diagrams, pictures) that you have come across?

Table 5.7: Q6 results: level of formality of ITI representation

Answer	%	Count
Very Informal	21.4%	15
Informal	28.6%	20
Somewhat Formal	28.6%	20
Formal	20.0%	14
Very Formal	1.4%	1

According to our expectations the level of formality when designing ITI are very informal, so the results of this question were unexpected since 50% of the respondents consider that most ITI representations are somewhat formal. A possible explanation was the added note explaining that formal representations are representations normally supported by metamodels and associated with notations, with capacity to detect errors were unnoticed.

5.5.5.3 Section III - Modeling ITIs (Answers)

This section presents the questionnaire answers to the questions related with Modeling ITI representations.

Question 7: Using your current approach, how **MUCH EFFORT** do you estimate it would be required to model an ITI like the one represented in the figure (appendix B) and add to it detailed technical information such as the one represented for one of the modeling elements?

Table 5.8: Q7 results: effort to model ITI representations

Answer	**%**	**Count**
More than 15 days	14.7%	10
6 to 15 days	32.4%	22
3 to 5 days	32.4%	22
1 to 2 days	19.1%	13
Less than 1 day	1.5%	1

The answers to this question allowed us to confirm that the process of representing ITIs is not only difficult, but is also time consuming with the vast majority of respondents taking between 5 to 15 days to represent an ITI.

Effort required to represent an ITI (question 7)

I've never done an infrastructure representation in which I had to give 5 attributes for each component. My models are always higher-level and very informal. If I did need to include 5 attributes for each component that would take me a lot longer.

Question 8: Think about the time you spent to represent an ITI using your current approach. How often that **INVESTMENT WAS WORTH** considering the overall quality in terms of completeness and consistency?

Table 5.9: Q8 results: time spent to represent ITIs

Answer	**%**	**Count**
Never	2.9%	2
Rarely	17.4%	12
Sometimes	43.5%	30
Most times	29.0%	20
Always	7.3%	5

The investment of time to represent ITIs was not always considered a good investment of respondent's time considering the overall quality of the representation. According to our experience most ITI representations created in tools such as *Microsoft Powerpoint* have graphics and icons to represent the ITI with high quality but lack in terms of consistency and in the details of each ITI component.

Question 9: How often the ITI representation (e.g. sketch, diagram, picture) is the **ONLY DOCUMENTATION** of the IT infrastructure?

Table 5.10: Q9 results: ITI documentation

Answer	%	Count
More than 75% of cases	26.1%	18
51% to 75% of cases	37.7%	26
25% to 50% of cases	18.8%	13
Less than 25% of cases	13.0%	9
Never (There is always other documentation)	4.4%	3

The expectations in terms of the ITI documentation was that, it is inexistent or when they exist, they are poor in quality and in the amount of information. The question confirmed our claims with a significant number of respondents answering that the ITI representations are the only documentation about the ITI.

Question 10: How often does the ITI representation (e.g. sketch, diagram, picture) contain enough information to support *per se* the whole ITI **IMPLEMENTATION**?

Table 5.11: Q10 results: implementation support

Answer	%	Count
Never contains enough information to support Implementation	29.4%	20
Less than 25% of cases	36.7%	25
25% to 50% of cases	23.5%	16
51% to 75% of cases	8.8%	6
More than 75% of cases	1.5%	1

According to our field experience most of ITI representations do not have enough information to support the ITI implementation activities. As can be seen in table 5.11 the results corroborate our field experience with most of the respondents confirming that the representations rarely have enough information.

Question 11: If any, what **NOTATION** do you use more frequently to represent IT infrastructures?

As was mentioned in section 2.3.1, most respondents answered that no notation is used to design ITIs. Some of the users that mentioned, on question 6, that most ITI representations are formal, or somewhat formal, answered that no notation was used in this question which, according to our definition of formal representation, is informal.

Table 5.12: Q11 results: notation used to represent ITIs

Answer	%	Count
NONE (No Notation)	44.1%	30
UML Deployment Diagrams	35.3%	24
ArchiMate	1.5%	1
Architecture Content Framework (TOGAF)	7.4%	5
Other (Please Specify)	11.8%	8

Notations used to represent ITIs (question 11)

No notation, but I choose this option to let you know that in my experience most people do come up with consistent approaches and develop their own "icon set" and minimum structured information layout, that they fine tune with experience and continued work.

5.5.5.4 Section VI - ITI Patterns (Answers)

This section presents the questionnaire answers to the questions related with ITI knowledge such as how to obtain, store and share information about ITIs. There are questions related also with reusing best practices and complexity, which are challenges that normally patterns can address.

Question 12: How do you classify the existing practices in ITI representations as a way to **CAPTURE KNOWLEDGE**?

Table 5.13: Q12 results: existing ITI practices to represent ITIs

Answer	%	Count
Completely Useless	6.2%	4
Useless	1.5%	1
Somewhat Useless	26.2%	17
Neutral	27.7%	18
Somewhat Useful	21.5%	14
Useful	10.8%	7
Very Useful	6.2%	4

Based on our field experience and since the state of the art of most ITI representations are based on pictures and sketches (with low quality) the task of recording in this format technical knowledge, insights, experiences or lessons learned using these representations are very limited. Most of the respondents classified the effectiveness of existing practices as somewhat useless or neutral.

Question 13: How do you classify the existing practices in ITI representations as a way to **STORE KNOWLEDGE** of a specific ITI component?

Table 5.14: Q13 results: existing ITI practices for storing knowledge

Answer	%	Count
Completely Useless	6.2%	4
Useless	4.6%	3
Somewhat Useless	24.6%	16
Neutral	20.0%	13
Somewhat Useful	24.6%	16
Useful	13.9%	9
Very Useful	6.2%	4

As can be seen in table 5.14, most respondents mentioned that existing ITI practices are not useful to store knowledge. According to our field experience, most of the knowledge is held by individuals, based on their accumulated experience, but not readily available elsewhere.

Question 14: How do you classify the existing practices in ITI representations as a way to **SHARE KNOWLEDGE**?

Table 5.15: Q14 results: existing ITI practices for sharing knowledge

Answer	%	Count
Very Good	4.6%	3
Good	16.7%	11
Somewhat Good	30.3%	20
Neutral	27.3%	18
Somewhat Bad	13.6%	9
Bad	3.0%	2
Very Bad	4.6%	3

Most respondents answered that existing ITI representations are good or somewhat good for sharing knowledge among individuals. One of the reasons that led some respondents to classify existing ITI practices as good or very good practices for sharing knowledge is that ITI representations are high level and easy to understand.

Question 15: How do you classify the ability of existing ITI representations to **REUSE IT** Infrastructure **BEST PRACTICES**?

As can be seen in table 5.16 almost half of respondents considered useless the existing forms of ITI representation to reuse ITI best practices, with an expected (from our

Table 5.16: Q15 results: reuse ITI best practices

Answer	%	Count
Completely Useless	4.6%	3
Useless	7.6%	5
Somewhat Useless	19.7%	13
Neutral	16.7%	11
Somewhat Useful	19.7%	13
Useful	21.2%	14
Very Useful	10.6%	7

perspective) value of respondents considering useful the existing practices. Based on our experience, there is a high degree of sharing diagrams, pictures and sketches about ITIs. However, apart of what is depicted in these diagrams, there is no way to represent and reuse these best practices in the ITI design. The following question about the importance of having a repository to promote the reuse of ITI best practices is related with this question.

Question 16: How useful a **REPOSITORY** with ITI best practices can be?

Table 5.17: Q16 results: ITI repository

Answer	%	Count
Completely Useless	0.0%	0
Useless	1.5%	1
Somewhat Useless	1.5%	1
Neutral	13.9%	9
Somewhat Useful	10.8%	7
Useful	24.6%	16
Very Useful	47.7%	31

An ITI repository with best practices was considered by the vast majority of respondents as a useful resource. The answers to this question confirmed the importance of having this information in a way that can be easily shared and reused.

Question 17: Which strategies do you use most to deal with **COMPLEX ITI** designs?

Most respondents confirmed that when dealing with complex ITI design issues they consult colleagues experienced with ITI design, which is completely aligned with our expectations.

Table 5.18: Q17 results: complex ITI designs

Answer	%	Count
Ask Questions on Online Forums	30.8%	20
Ask Questions on Distribution Lists	21.5%	14
Ask for Advisory Services	38.5%	25
Search for Product Blueprints	64.6%	42
Contact Manufacturers	56.9%	37
Ask Colleagues	69.2%	45
Other (Please specify)	6.2%	4

Other strategies (question 17)

- *Peer Review.*
- *Consult with SMEs.*
- *Design ourselves using TOGAF certified architects on our staff.*
- *Subject matter experts' blogs are usually a good source also, as they tend to share and write on detailed practical knowledge.*

Question 18: How do you share your **ITI KNOWLEDGE** with others?

Table 5.19: Q18 results: ITI knowledge sharing

Answer	%	Count
Classroom Training	29.2%	19
Online Training	24.6%	16
On-the-job Training	64.6%	42
Coaching	53.9%	35
Recommend Literature	41.5%	27
Other (Please specify)	10.8%	7

As can be seen in table 5.19 ITI knowledge is shared using on-the-job training and coaching. These results are very aligned with our expectations. As stated before the ITI knowledge is held by ITI professionals based on their accumulated experience and not available for reuse.

Other forms of sharing ITI knowledge (question 18)

- *Knowledge base articles.*
- *Peer-reviewed technical papers.*
- *Via my consulting role for the last 17 years.*
- *Good documentation and knowledge transfers.*
- *Informal sharing with peers/colleagues, DLs, blogging, mailing lists.*
- *Other informal written forms, like field notes and not-so-structured documentation (OneNote notebooks, for example).*

5.5.5.5 Section V - Tools (Answers)

This section presents the questionnaire answers to the questions related with the use of software tools to represent ITIs.

Question 19: What **SOFTWARE TOOLS** do you use to represent ITIs?

As mentioned in section 2.4, according to the respondents there is not a single software tool to represent ITIs. The vast majority of respondents mention that they use *Microsoft Office Tools* such as *Visio* and *PowerPoint* to create visual representations and *Word* and *Excel* for descriptive representation and details. Table 2.8 present the list of 34 tools provided by all survey respondents.

Question 20: How do you **CLASSIFY** these tools in terms of their capacity to represent ITIs?

Table 5.20: Q20 results: tools capacity

Answer	%	Count
Completely Useless	1.6%	1
Useless	6.4%	4
Somewhat Useless	1.6%	1
Neutral	15.9%	10
Somewhat Useful	30.2%	19
Useful	36.5%	23
Very Useful	7.9%	5

Most respondents classified as useful the capabilities to represent ITIs with their existing tools. Visio was the top used tool, which offers limited capacity to represent ITI concepts.

Question 21: Which is the most recurrent **FORMAT** used to represent IT infrastructures that you had ever access to?

Table 5.21: Q21 results: how ITIs are represented

Answer	**%**	**Count**
Diagramming Software (e.g. Microsoft Visio)	67.7%	44
Presentation (e.g. Microsoft PowerPoint)	20.0%	13
Drawing Software	4.6%	3
CMDB	3.0%	2
Hand-written Sketches	3.0%	2
Other. Please Specify?	1.5%	1

This question confirmed that diagramming software such as Microsoft Visio followed by Presentation software are the most used when representing ITIs.

Other (question 21)

- *I have seen all of the above being used.*
- *Varies widely depending on the scope and size of the project. Except for the CMDB (which in my experience is uncommon, except as a component of the Infrastructure itself).*

Question 22: Please evaluate the ITI profile based on the ITI profile video.

Table 5.22: Q22 results: ITI profile questions

Question	**Str. Dis-agree**	**Dis-agree**	**Neither Agree nor Disagree**	**Agree**	**Str. Agree**
Better documented ITIs	0.0%	1.6%	12.9%	43.6%	41.9%
Can be used by non ITI experts	1.6%	19.4%	22.6%	40.3%	16.1%
Reduces the time to create an ITI	0.0%	4.8%	25.8%	40.3%	29.0%
Support implementation	0.0%	3.2%	12.9%	51.6%	32.2%
Knowledge sharing simpler	0.0%	4.8%	22.6%	38.7%	33.9%

After watching the video (appendix B), most respondents considered that the approach can contribute to have better documented ITIs, make knowledge sharing simple, reduce the time to build ITI representations and provide more information to support ITI implementation activities.

Question 23: Please evaluate the ITI patterns approach based on the ITI patterns video.

A video demonstrating the process of representing ITI with ITI patterns was shown before asking these questions. As can be seen in the results presented in table 5.23, the respondents considered that ITI patterns can decrease design effort, ITI patterns

Table 5.23: Q23 results: ITI patterns questions

Question	Str. Disagree	Disagree	Neither Agree nor Disagree	Agree	Str. Agree
Decreases design effort	0.0%	3.2%	12.9%	51.6%	32.3%
Importance of a repository	0.0%	0.0%	11.3%	43.6%	45.1%
Knowledge sharing simpler	0.0%	4.8%	22.6%	38.7%	33.9%

repository is very important to simplify ITI knowledge reuse and the knowledge transfer activities are simple.

5.5.5.6 Section VI - About You (Answers)

The last section of the questionnaire was dedicated to understanding more of the respondents.

Question 24: YEARS working with ITIs?

Table 5.24: Q24 results: years working with ITIs

Experience	Respondents	Experience	Respondents
1 Year	1	14 Years	2
2 Years	1	15 Years	3
4 Years	1	16 Years	5
5 Years	4	17 Years	2
6 Years	2	18 Years	4
8 Years	4	20 Years	12
9 Years	2	22 Years	1
10 Years	5	25 Years	3
11 Years	2	29 Years	1
12 Years	1	30 Years	4

As was mentioned in section 2.3 we were able to verify that the respondents had experience with ITIs. The results showed that the majority of respondents had 20 years of experience and only 3 respondents had less than 5 years' experience. This question was optional.

Question 25: Highest level of **EDUCATION** or degree completed?

Regarding the level of education, which is also an optional question we had 64 answers with 19 having a graduate degree and with 17 respondents with a master and 17 with a doctoral degree as can be seen in table 5.25.

Table 5.25: Q25 results: level of education

Education	**%**	**Count**
Doctoral	26.6%	17
Master	26.6%	17
Graduate	29.7%	19
High School	14.1%	9
Other	3.1%	2

Question 26: Please provide additional information regarding ITI representations that you want to share.

To allow respondents to comment details about their experiences with ITI representation not covered in the questionnaire we allowed them to provide comments.

Comment I (question 26)

Very interesting approach, I'm glad I got to know about it. Just one crucial absence in the presented toolbox and approach: data flows (not the rules, just the flows and perhaps main constraints like protocols and other 1st tier information). It has become increasingly critical to represent data flows in a IT infrastructure diagram, as to clearly identify from where and to where and how is data being transported, stored and accessed. This is specially important as it pertains to knowledge sharing - as an IT architect, I need to have the flows in the diagram so I can easily assess the fit and compliance with business needs and requisites. Maybe it's there and it just didn't happen to be shown or given relevance :-) after all, it has a direct link with practices and patterns. I'll be glad to discuss this if you find it interesting too, so just let me know.

Comment II (question 26)

Agree with the lack of a tool to better represent IT infrastructures and its relevant details!

Comment III (question 26)

Promising and interesting approach to a problem all IT architects face. The extensive nature of UML seems to be a good fit for this non-standard use. On its own, as a way to formalize (and especially, to normalize) the representation and collection of an IT infrastructure design. I believe this approach is worthwhile investing on. However, I am not so convinced this approach is enough to transition directly from design to deployment without further data which will probably come from other 3rd party repositories. It would be important to highlight how to bridge the gap to the configuration as code, which seems to be a positive trend in the industry. I agree the focus must be placed on the functional aspects of the approach; that is where there is currently less standardization. It appears to be more of a creation toolset for architects and consultants (and maybe some IT managers), and more of a library or CMDB type repository for all other technical roles involved in the IT lifecycle. However, it is important to be able to easily transition to beautiful and colorful visual representations, which are critical for "selling the concept" to the business decision makers in the early stages of the project life; if after this we still have to go and make separate drawings in Visio and PowerPoint to include in the scoping presentations, some of the effort is lost and we end up going back to the original problem.

Comment IV (question 26)

I see as very useful the possibility to relate or connect the ITI diagrams with CMDB, so the information, like attributes could be synchronized.

Comment V (question 26)

If that's what the tool is intended to support, I think a much more difficult part of the task is obtaining the data to go into the representation, rather than drawing the representation - your tool seems to be focused on the latter. It would be great if your tool could automatically populate the representation with some attribute data. The reason why infrastructure representations are often inaccurate is that the difficulty of obtaining the data (especially with Bring Your Own Device), not the difficulty of drawing infrastructure diagrams. The repository of patterns could be handy for drawing the diagram if the company's architecture matches one of the patterns, but it might not - architectures vary widely. Like most employees, I do every so often need to make an informal, very high-level architecture sketch without much details.

Comment VI (question 26)

The major issue that I found with the IT infrastructure representations is that they are not easy to update, and after a short period of time they become outdated. The presented tool would make it easier to update, although this task must be included in the change process.

Comment VII (question 26)

I'm increasingly realizing that a good enough approach for documenting an infrastructure is the mix of a simple, 2d diagram that shows the overall design and a configuration matrix that documents all aspect of the architecture. usually config matrix with the right level of detail (complemented with a graphic representation) is enough to document the infrastructure.

Question 27: Additional information regarding the questionnaire that you want to share?

This question was intended to allow respondents to provide more feedback related with the questionnaire in general.

Comment I (question 27)

Well done, with short videos and relevant information!

Comment II (question 27)

Although effective, video quality could be improved and provide a somewhat better experience - not a main issue, just an improvement opportunity :)

Comment III (question 27)

In the spirit of an open discussion, I think the Profiles of the professionals that are responsible, or can be responsible to elaborate a Datacenter Representation, needs to be debated. What I mean is, for example, I disagree that a less experienced IT professional, should be responsible for the creation of a Datacenter representation, if we are to go deep on the level of detailed information. It can be however a team effort. This of course, taking into account the type of design we are talking about... Concept?, Physical or Logical Again as an example, for concept, we need someone with strategic and technical knowledge, but not necessarily deep technical expertise over infrastructures.... does this make sense, or did I miss the point?

5.6 Controlled Experiment

5.6.1 Introduction

To improve our understanding on the impact of our proposals for ITIs designs, it is essential to develop and test their efficiency and effectiveness. To validate our proposals, we used empirical research in ITIs, which implies creating experiences and checking if our understanding is correct through testing hypotheses against collected data.

As will be detailed in the following sections, we setup an experiment with professional subjects. The experiment consists of three stages: *Demo*, *Modeling* and *Wrap-up*. In the *Demo* phase, the goal is to introduce concepts to the audience and demonstrate, with an example, what is expected. In the *Modeling* stage, there are three different phases, each with 20 minutes duration. The last phase, *Wrap-up*, consists in a questionnaire designed to gather information from the participants after the *Modeling* stage. It contains questions concerning the previous stages and the usefulness of the ITI profile and patterns during design. The data collected in this phase will be useful for the interpretation and validation of results.

We followed Kitchenham et al. [KP02c] guidelines to design experiments, primarily in the planning phase of the experiment and the Jedlitschka and Pfahl [Jed+08] guidelines were used to report the experimental results.

5.6.1.1 Independent variables

As stated before, the *modeling stage* of the experiment encompasses three phases, one for each experiment. Based on the desired goal, three different types of assignments were defined for the same subjects[3]. We had just one group of participants and we applied three treatments (within-groups design). Table 5.26, shows the independent variables that can influence the dependent variables.

5.6.1.2 Dependent variables

During the planning phase, three dependent variables were identified for the experiment, as shown below in Table 5.27.

The *attcoverage* (Attributes coverage) variable measures the percentage of properties or attributes filled in the diagram by the participant in the assignment. This variable is based on *attfilled* (Addressed attributes) variable or the number of attributes that the participant represented in the experiment and *totalatt* (Total attributes) variable which is the maximum number of attributes that can be represented by the participant. This measures the understandability of a diagram as expressed in the following ratio:

[3]Since the experiment was performed with human subjects, the term "participants" will be used in the remainder of this document.

Table 5.26: Independent variables

Variable	Description	Scale Type
Personal Information		
participant	Participant number	Scale
education	Level of education	Ordinal
yearsindustry	Number of years working in the IT industry	Scale
yearsiti	Number of years working with ITIs	Scale
yearsitidesign	Number of years designing IT infrastructures	Scale
Design of IT infrastructures		
itidpermonth	Number of ITI designs produced per month	Scale
itivalidated	The ITI design is reviewed by peers	Dichotomous
iticreationtime	Time to produce an ITI design	Ordinal
formality	Level of formality used when designing ITIs	Ordinal
itiformat	Common format to represent ITIs	Nominal
itidesignmain	The ITI design is the main outcome	Dichotomous
itiimplement	The ITI design supports implementation	Dichotomous
timevsbenefits	The time invested in ITI design was worthy	Dichotomous
Experimental assignment		
designtechnique	Methodology used (ad-hoc, profile, patterns)	Nominal
assignment	Assignment used in the experiment	Nominal
totalatt	Total attributes in the assignment	Nominal
totalreq	Total number of mandatory requirements	Scale
time	Time to complete the assignment	Scale

$$Attributes\ coverage = \left(\frac{attfilled}{totalatt}\right) \tag{5.1}$$

Each assignment in the experiment contains a set of requirements that the participant should include in the ITI design. The *reqcoverage* (Requirements coverage) variable is used to express the correctness based on how many of these requirements were correctly addressed *reqfilled* (Addressed requirements) during the design activity and the *totalreq* (Total requirements), as expressed in the following ratio:

$$Requirements\ coverage = \left(\frac{reqfilled}{totalreq}\right) \tag{5.2}$$

The *efficiency* (Design efficiency) is a ratio that measures the results obtained by the participant in terms of attributes addressed *attfilled*variable in the assignment face to the *time* (Execution time) variable needed to address these requirements.

$$Design\ efficiency = \left(\frac{attfilled}{time}\right) \tag{5.3}$$

Table 5.27: Dependent variables

Variable	Description	Scale Type
attcoverage	% of attributes correctly addressed by the participant	Ratio
reqcoverage	% of requirements addressed by the participant	Ratio
efficiency	Design efficiency in the experiment	Ratio
attfilled	Attributes addressed by the participant	Scale
reqfilled	Requirements addressed by the participant	Scale
difficulty	Perceived assignment difficulty	Ordinal

5.6.2 Goals, Hypotheses, Parameters, and Variables

The goal of the experiment was to find out if the infrastructure design based on ITI patterns leads to higher quality in terms of completeness, requirements coverage and design efficiency, when compared to infrastructure designs produced with *ad-hoc* methods. As defined in section 5.4, the null hypotheses under evaluation are:

H01: The usage of the proposed concrete representation during the ITI design process has no influence on *Attributes coverage*, when compared with *ad-hoc* methods.

H02: The usage of the proposed concrete representation during the ITI design process has no influence on *Requirements coverage*, when compared with *ad-hoc* methods.

H03: The usage of ITI patterns during the ITI design process has no influence on the *Attributes coverage*, when compared with *ad-hoc* methods.

H04: The usage of ITI patterns during the ITI design process has no influence on the *Requirements coverage*, when compared with *ad-hoc* methods.

H05: The use of methodological support does not reduce the *Design efficiency*, when compared with *ad-hoc* methods.

5.6.3 Experimental Design

5.6.3.1 Participants

To test the hypotheses, five participants participated in a controlled experiment. The relative low number of participants is related with our requirement of having experienced participants with a strong professional background, instead of students without experience.

5.6.3.2 Treatments

In the first phase of the *modeling stage* (hereinafter referred as *ad-hoc phase*"), a modeling exercise was assigned to each participant a modeling assignment that should be solved using *ad-hoc* methods. Then, in the second phase (hereinafter referred as *profile phase*), a different modeling assignment was assigned to each participant, but the ITI profile was made available to participants. In the third and last phase (hereinafter referred as *pattern phase*) participants had access to ITI patterns and received a different assignment. Each of these three phases had a maximum allowed time of 20 minutes.

5.6.3.3 Instruments

The basis for the experiment were five different assignments, each with a brief description of the organization and a set of ITI design requirements to guide the participants in the experiment. The requirements were presented as a bullet list in an informal/natural language, thereby simplifying the evaluation of *requirements coverage*.

5.6.3.4 Validation procedure

As stated before, the experiment took place in a controlled environment with five participants. Table 5.28 shows the allocation of modeling assignments to participants.

Table 5.28: Modeling assignment allocation

Phase	Participant1	Participant2	Participant3	Participant4	Participant5
ad-hoc	Exercise A	Exercise B	Exercise C	Exercise D	Exercise E
profile	Exercise C	Exercise D	Exercise E	Exercise A	Exercise B
pattern	Exercise E	Exercise A	Exercise B	Exercise C	Exercise D

5.6.3.5 Validity

We are aware of the limitation of the experiment in terms of external validity, and conducting a more broad experiment is future work. Ideally, the study should have been conducted in a field experiment with many real projects, but that was not feasible due to time, budget and non-disclosure constraints.

5.6.3.6 Blocking

To make sure that the results from both experiments were not influenced by factors such as learning experience, five different assignments were created, and each participant had to solve three different assignments during the experiment.

The generalization to the entire population is not sustainable since we did not use a random sample. Instead we used a convenience sample of experienced participants with different experiences and backgrounds.

5.6.3.7 Instrumentation

As can be seen in Table 5.29, each stage had a different set of instruments.

Table 5.29: List of phases and instruments

Stages	Instrument	Objective
Demo	Demo assignment	Explain the purpose of experiments with a demo
Modeling	Exercises (A .. E)	Requirements for ITI modeling assignments
	Virtual Machine	Virtual machine with all software installed.
	ITI profile tutorial	Instructions on how to use the ITI profile
	ITI patterns tutorial	Instructions on how to use the ITI patterns
Wrap-up	Questionnaire	Gather participants feedback

The virtual machine (VM) had all software installed and configured (e.g. Visio, Sparx EA and the ITI profile and patterns configured to allow each participant to perform the experiment) and each VM had three different accounts (*ad-hoc*, *profile* and *pattern*). Each account had access to a different set of instruments, according to each experimental phase. In the ad-hoc phase participants were also allowed to use Visio or image editing software to do the assignment. The same VM was made available for each participant. There was also a tutorial explaining how to use the profile and patterns.

5.6.3.8 Blinding

To prevent bias, the participants were not allowed to talk to each other about the modeling assignments, until the experiment was concluded. The results were collected and analyzed by three independent experts (myself and my two advisors), that did not participate as experimental subjects.

5.6.3.9 Data collection

After 15 minutes of introduction and answering questions for clarification purposes in the *demo* stage, the participants started the *modeling* stage. As stated before, a maximum of 20 minutes was defined for each assignment. In the end (*wrap-up stage*), a questionnaire was distributed and filled by each participant and collected along with the files resulting from the experiments. The time required to complete the experiment was also recorded (*Execution Time*) for each phase. The whole experiment took 80 minutes to conclude.

5.6.4 Execution

This section provides more details regarding the execution of the experimental design plan described earlier.

5.6.4.1 Sample and preparation

As described in the design section, during each phase of the *modeling stage*, different instruments were provided to participants.

In the *ad-hoc phase* with the exception of the VM, no other instruments were provided. In the *profile phase* each participant had access to:

- **ITI profile VM.** There was a user account called *profile* and after logging on the user had access to Sparx EA loaded with the ITI profile to perform the experiment, plus a palette with modeling elements.
- **ITI profile tutorial.** Consisted in a document similar to a user guide, with detailed instructions to complete the tasks related with the ITI profile.
- **ITI profile demo.** Five minutes video, to demonstrate how the ITI profile can be used with an example.

In the *pattern phase* the following instruments were provided to each participant:

- **ITI patterns VM.** There was a user account called *pattern* and, after logging on, the user had access to Sparx EA loaded with the ITI profile to perform the experiment, plus a palette of ITI patterns.
- **ITI patterns tutorial.** Consists in a document similar to a user guide with detailed instructions to complete the tasks related with ITI patterns.
- **ITI patterns demo.** Ten minutes video, to demonstrate how the ITI patterns can be used, with an example.

5.6.4.2 Collected results

The results were collected for each participant. Table 5.28 summarizes the results. More details of the individual results and assignment allocation are available in appendixes and summarized in the following tables:

- Exercise A - Scientists (Table E.2)
- Exercise B - OilSemaphore (Table E.3)
- Exercise C - New Remote Offices (Table E.4)
- Exercise D - Internet Access (Table E.5)
- Exercise E - Payroll Application (Table E.6)

As was mentioned before and can be seen in the table 5.30, each participant solved different exercises using the ad-hoc, profile or pattern approach.

5.6.5 Experimental Results

As mentioned before the same participants participated in the application of different design techniques (different treatments), so we will use a within-subjects design technique. Since we want to analyze the difference between the results of three design techniques (one factor with three levels: ad-hoc, profile, patterns), the first step is to verify the assumptions for the application of the parametric one-way Repeated-Measures Analysis Of Variance (RM-ANOVA).

Table 5.30: Participant's results

participants	designtechnique	attfilled	totalatt	attcoverage	reqfilled	totalreq	reqcoverage	time	efficiency
1	ad-hoc	42	211	0,20	42	50	0,84	12,6	3,3
1	profile	64	357	0,18	49	49	1,00	13,3	3,7
1	patterns	144	183	0,79	47	47	1,00	9,8	4,8
2	ad-hoc	48	247	0,19	48	50	0,96	9,6	5,0
2	profile	66	242	0,27	50	54	0,93	7,8	6,4
2	patterns	153	211	0,73	50	50	1,00	9,6	5,2
3	ad-hoc	36	357	0,10	36	49	0,73	16	2,3
3	profile	60	183	0,33	45	47	0,96	10,4	4,3
3	patterns	138	247	0,56	50	50	1,00	16	3,1
4	ad-hoc	51	242	0,21	51	54	0,94	12,6	4,0
4	profile	88	211	0,42	47	50	0,94	15,3	3,1
4	patterns	204	357	0,57	49	49	1,00	16,2	3,0
5	ad-hoc	41	183	0,22	41	47	0,87	12	3,4
5	profile	63	247	0,26	44	50	0,88	18	2,4
5	patterns	143	242	0,59	54	54	1,00	12	4,5

Repeated Measures:

The objective of this test would be to assess if there is an impact of the *designtechnique* (independent variable) on the *attfilled* (outcome or dependent variable), representing the number of properties or attributes identified by the participant.

The use of RM-ANOVA has the following five assumptions:

- **Assumption 1.** The dependent variable should be measured at the continuous level or absolute scale. The *addressedattributes* is the number of attributes identified in the diagram.
- **Assumption 2.** The independent variables define at least two categorical "related groups", that corresponds to the phases in this experiment, which indicates that the same subjects are present in all groups. In our example, the same subjects in each phase were measured in three occasions on the same dependent variable.
- **Assumption 3.** There should be no significant outliers in the groups since they can have a negative effect on the RM-ANOVA, distorting the differences among related groups (whether increasing or decreasing the scores on the dependent variable) and can reduce the accuracy of results.
- **Assumption 4.** The distribution of the dependent variable in the two or more related groups should be approximately normally distributed. The normality can be tested using the *Shapiro-Wilk test of normality*.

- **Assumption 5.** The variances of the differences between all combinations of related groups must be equal, which is known as *sphericity*. Violating the assumption of *sphericity*, leads to an increase in the Type I error rate; that is, the likelihood of detecting a statistically significant result when there is not one. The *sphericity* can be evaluated with *Mauchly's test*.

It is trivial to verify that the three first assumptions to use RM-ANOVA were met. Regarding the forth assumption, we tested the normality using two well-known tests of normality, namely the *Kolmogorov-Smirnov* Test and the *Shapiro-Wilk* test. We will only consider the results of the *Shapiro-Wilk* test to assess normality since the latter is more appropriate for small sample sizes (< 50 samples).

Table 5.31: Shapiro-Wilk test on *attcoverage*

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	designtechnique	Statistic	df	Sig.	Statistic	df	Sig.
attcoverage	adhoc	,349	5	,045	,768	5	,043
	patterns	,310	5	,132	,832	5	,144
	profile	,197	5	,200*	,976	5	,913

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Table 5.32: Shapiro-Wilk test on *reqcoverage*

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	designtechnique	Statistic	df	Sig.	Statistic	df	Sig.
reqcoverage	adhoc	,184	5	,200*	,936	5	,636
	profile	,192	5	,200*	,985	5	,962

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

c. reqcoverage is constant when designtechnique = patterns. It has been omitted.

Table 5.33: Shapiro-Wilk test on *efficiency*

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	designtechnique	Statistic	df	Sig.	Statistic	df	Sig.
efficiency	adhoc	,181	5	,200*	,978	5	,926
	patterns	,247	5	,200*	,862	5	,235
	profile	,217	5	,200*	,934	5	,623

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Based on a confidence level of 95% which implies a test significance $\alpha = 0.05$ (probability of type I error of 5%) we can conclude, based on the significance level greater than 0.05, that we failed to reject the null hypothesis, so the assumption four is met since the data does not deviate from a normal distribution.

To verify the last assumption, the data will be analyzed using Mauchly's *sphericity* tests to test that variations or standard deviations are going to be the same among ad-hoc, profile and patterns.

Table 5.34: Mauchly's test of sphericity

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b]		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
DesignTechnique	,060	8,443	2	,015	,515	,531	,500

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: DesignTechnique

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Based on the results of table 5.34 and considering an approx. Chi Square of 8,443, a degree of freedom of 2 and based on a confidence level of 95% which implies a test significance $\alpha = 0.05$ we can conclude that we must reject the null hypothesis, and we cannot assume the homogeneity of variance and covariances, since we get a significance $\rho < \alpha$. So the Mauchly's Test of Sphericity indicated that the assumption of *sphericity* has been violated, $x^2(2) = 8.443, p = .015$.

Since the RM-ANOVA cannot be used due to violation of assumption five, we will use the non-parametric equivalent: The Friedman test.

Friedman test:

The Friedman test is an extension of the Wilcoxon test and are the equivalent non parametric tests to RM-ANOVA tested earlier, that is also applicable when we have subjects that participate in two or more treatments. The test will determine if the designs produced varied significantly across the application of each of the three design techniques (treatments). With repeated-measures designs, each participant in an experiment has scores on K variables obtained in each treatment. In the case of the Friedman test the observations come in sets of K observations. The use of Friedman test has the following four assumptions:

- **Assumption 1.** One group that is measured on three or more different occasions.
- **Assumption 2.** Group is a random sample from the population.
- **Assumption 3.** Your dependent variable should be measured at the ordinal or continuous level.

- **Assumption 4.** Samples do NOT need to be normally distributed.

All assumptions of the Friedman test are met since in our case, we have three sets of data corresponding each to the design techniques: ad-hoc, profiles and patterns. Within each set the observations are dependent, but between sets the observations are independent. To evaluate if there is difference between the treatments, the following null and alternative hypothesis will be evaluated:

H0: There is no difference among the application of the three design techniques.
H1: There is difference among the application of the three design techniques.

The null hypothesis states that the population medians are equal for the K levels of a factor. The evaluation of both hypotheses will be performed for the following variables:

- *Attributes coverage*
- *Requirements coverage*
- *Efficiency*

We will consider an $\alpha = 0.05$ (probability of type I error of 5%). The degree of freedom is $k-1$ where k is the number of treatments. Since we have 3 design techniques $df = 2$. Looking at the chi square table for an $\alpha = 0.05$ with $df = 2$, the critical value is 5.99147. So, if X^2 is greater than 5.99 the null hypothesis will be rejected.

The formula for Friedman test is as follows:

$$\chi_r^2 = \frac{12}{nk(k+1)} \sum R^2 - 3n(k+1) \tag{5.4}$$

where (n) is the number of participants, (k) is the number of treatments and (R) is the sum of ranks.

5.6.5.1 Analysis of *Attributes coverage*:

The descriptive statistics for *Attributes coverage* is presented in table 5.35.

Table 5.35: *Attribute coverage* descriptive statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles 25th	Percentiles 50th (Median)	Percentiles 75th
adhocattcoverage	5	,18580	,048803	,101	,224	,14750	,19900	,21750
profileattcoverage	5	,29040	,088610	,179	,417	,21700	,27300	,37250
patternsattcoverage	5	,64660	,102882	,559	,787	,56500	,59100	,75600

Ranking the results obtained for each participant (based on data collected and available in table 5.30), will result in an $R = 6$ for ad-hoc design technique, $R = 9$ for profile and $R = 15$ for patterns as can be seen in table 5.36.

Table 5.36: *Attribute coverage* ranking

participants	**ad-hoc**	**profile**	**patterns**
1	2	1	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
R	6	9	15

Table 5.37 shows the test statistics results of the Friedman test, and whether there was an overall statistically significant difference between the mean ranks among design techniques for *Attributes coverage*.

Table 5.37: Friedman test on *Attribute coverage*

	Mean Rank
adhocattcoverage	1,20
profileattcoverage	1,80
patternsattcoverage	3,00

N	5
Chi-Square	8,400
df	2
Asymp. Sig.	,015

$$\chi_r^2 = \frac{12}{(5)(3)(3+1)}(6^2 + 9^2 + 15^2) - 3(5)(3+1) \tag{5.5}$$

$$\chi_r^2 = 8,4$$

Based on the test statistic χ_r^2 value ("Chi-square"), degrees of freedom ("df") and the significance level ("Asymp. Sig."), there was a statistically significant difference depending on which design technique was used, χ_r^2 = 8.400, p = 0.015.

Considering $(2, n = 5), p > 0.05$ **we will reject the null hypothesis and conclude that there is a difference among the three groups.**

Based on this the following hypotheses can also be rejected:

H01: The usage of the proposed concrete representation during the ITI design process has no influence on *Attributes coverage*, when compared with *ad-hoc* methods.

H03: The usage of ITI patterns during the ITI design process has no influence on the *Attributes coverage*, when compared with *ad-hoc* methods.

There was a statistically significant difference of the usage of the proposed concrete representation during the ITI design process on *Attributes coverage*, when compared with *ad-hoc* methods and there was a statistically significant difference of the usage of ITI patterns during the ITI design process on *Attributes coverage*, when compared with *ad-hoc* methods, χ_r^2 = 8.400, p = 0.015.

5.6.5.2 Analysis of *Requirements coverage*:

The descriptive statistics for *Requirements coverage* is presented in table 5.38.

Table 5.38: *Requirements coverage* descriptive statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
adhocreqcoverage	5	,87020	,090422	,735	,960	,78750	,87200	,95200
profilereqcoverage	5	,94060	,043827	,880	1,000	,90300	,94000	,97850
patternsreqcoverage	5	1,00000	,000000	1,000	1,000	1,00000	1,00000	1,00000

Ranking the results will result in an $R = 8$ for ad-hoc design technique, $R = 7$ for profile and $R = 15$ for patterns as can be seen in table 5.39.

Table 5.39: *Requirements coverage* ranking

participants	ad-hoc	profile	patterns
1	1	2	3
2	2	1	3
3	1	2	3
4	2	1	3
5	2	1	3
R	8	7	15

Table 5.40 show the test statistics result of the Friedman test, and whether there was an overall statistically significant difference between the mean ranks among design techniques for *Requirements coverage*.

Table 5.40: Friedman test on *Requirements coverage*

	Mean Rank
adhocreqcoverage	1,40
profilereqcoverage	1,70
patternsreqcoverage	2,90

N	5
Chi-Square	6,632
df	2
Asymp. Sig.	,036

$$\chi_r^2 = \frac{12}{(5)(3)(3+1)}(8^2 + 7^2 + 15^2) - 3(5)(3+1) \tag{5.6}$$

$$\chi_r^2 = 6,6$$

Based on the test statistic χ_r^2 value, degrees of freedom and the significance level, there was a statistically significant difference depending on which design technique was used, $\chi_r^2 = 6.632$, p = 0.036.

Considering $(2, n = 5), p > 0.05$ **we will reject the null hypothesis and conclude that there is a difference among the three groups.**

Based on this the following hypotheses can also be rejected:

H02: The usage of the proposed concrete representation during the ITI design process has no influence on *Requirements coverage*, when compared with *ad-hoc* methods.

H04: The usage of ITI patterns during the ITI design process has no influence on the *Requirements coverage*, when compared with *ad-hoc* methods.

There was a statistically significant difference of the usage of the proposed concrete representation during the ITI design process on *requirements coverage*, when compared with *ad-hoc* methods and there was a statistically significant difference of the usage of ITI patterns during the ITI design process on *Requirements coverage*, when compared with *ad-hoc* methods, χ_r^2 = 6.632, p = 0.036.

5.6.5.3 Analysis of *Efficiency*:

The descriptive statistics for *Efficiency* is presented in table 5.41.

Table 5.41: *Efficiency* descriptive statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
adhocefficiency	5	3,60960	1,010881	2,250	5,000	2,79150	3,41700	4,52400
profileefficiency	5	5,65900	1,820457	3,500	8,462	4,15600	5,75200	7,11550
patternsefficiency	5	12,75340	2,814098	8,625	15,938	10,27100	12,59300	15,31600

Ranking the results will result in an $R = 5$ for ad-hoc experiment, $R = 10$ for Profile and $R = 15$ for patterns as can be seen in table 5.36.

Table 5.42: *Efficiency* ranking

participants	ad-hoc	profile	patterns
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
R	5	10	15

Table 5.43 show the test statistics result of the Friedman test, and whether there was an overall statistically significant difference between the mean ranks among design techniques for *Efficiency*.

Table 5.43: Friedman test on *Efficiency*

	Mean Rank
adhocefficiency	1,00
profileefficiency	2,00
patternsefficiency	3,00

N	5
Chi-Square	10,000
df	2
Asymp. Sig.	,007

$$\chi_r^2 = \frac{12}{(5)(3)(3+1)}(5^2 + 10^2 + 15^2) - 3(5)(3+1) \tag{5.7}$$

$$\chi_r^2 = 10$$

Based on the test statistic χ_r^2 value, degrees of freedom and the significance level, there was a statistically significant difference depending on which design technique was used, χ_r^2 = 10.000, p = 0.007.

Considering $(2, n = 5), p > 0.05$ **we will reject the null hypothesis and conclude that there is a difference among the three groups.**

Based on this the following hypothesis can also be rejected:

H05: The use of methodological support does not reduce the *Design efficiency*, when compared with *ad-hoc* methods.

There was a statistically significant difference of the usage of the use of methodological support on the *Design efficiency*, when compared with *ad-hoc* methods, χ_r^2 = 10.000, p = 0.007.

5.7 Summary

This chapter showed how our proposals were validated using a controlled experiment and the online survey. Sections 5.2, 5.3, and 5.4 include the research questions and hypotheses to address the problems: *(i)* informal notation used to represent ITIs, the *(ii)* inability to express best practices in the ITI design and *(iii)* the nonexistence of a methodological support for a pattern based approach for ITI design.

The online survey is detailed in section 5.5 and includes how the survey was designed and distributed. The results are available in section 5.5.5. Based on the survey results we validate that our methodological support with ITI profile created, contribute to have better documented ITIs, reduces the time to create ITIs, it can be used to support implementation activities and makes simple the knowledge transfer about ITIs. The survey also confirmed that the use of our methodological support with ITI patterns contribute to decrease the ITI design effort.

In the controlled experiment, beside the definition of goals, hypotheses, parameters and independent and dependent variables, the experimental design was described. Based on the non-parametric *Friedman* tests the null hypotheses were rejected meaning that the proposed concrete representation influences and increases the number of requirements and the number of attributes addressed during the ITI design process when compared with *ad-hoc* methods.

The same approach was used to validate the methodological supported created during this research work and we were able to validate the increase in design efficiency when compared with *ad-hoc* methods.

[This page has been intentionally left blank]

Chapter 6

Conclusions and Future Work

Contents

6.1 Introduction

A well planned and designed ITI creates value to the business. Many organizations have experienced the impact of poorly designed ITIs, such as high latency times, difficulties in interoperability, poor business alignment, among other symptoms. In the last decade ITIs evolved and increased dramatically, being more and more responsible for business critical solutions, whose failure causes a huge impact in the business.

ITIs have to be designed faster and based on optimal solutions that not only proved that they worked, but also represent a reliable, flexible, secure solution and are economically viable.

The promise of transforming ITIs to be more dynamic has been caught by large organizations as a business opportunity to sell their services and solutions. In most cases, the products are presented in extensive *blueprints*, containing global ITI solutions, as a strategy to position as much products and services as possible from their portfolio.

These *blueprints* are most times ad-hoc diagrams, extensive, hard to read and vendor-dependent, which limits the interoperability among solutions from different vendors and most times do not cope with the required ITI innovation pace.

The main issues observed in more than 18 years working and designing ITIs and confirmed by the systematic literature review, an online survey and a controlled experiment are:

- Informal notations and *blueprints* are used to support the ITI design.
- Lack of technology agnostic solutions that can be applied to ITIs.
- Time required to design complex ITIs is high and the business cannot wait.
- The topic is complex. Designing ITIs encompasses expertise in software, hardware, networking, storage that rarely is mastered by a single person.
- ITI models rarely exist.
- An ITI can be very complex, particularly in large organizations such as banks and telecommunications companies.
- ITI modeling knowledge is not reused.
- Difficulty of communication among heterogeneous ITI teams or stakeholders;

In summary, the state of the art in the design of ITIs is poor, which motivated this work and its contributions.

To address the main issues, and in particular the identified research problems, we took an agnostic standpoint regarding the applicability of existing solutions and we assumed that organizations should not have to adopt solutions from a single manufacturer.

We introduced semi-formal specifications based on models to the design of ITIs which represents a foundation for designing complex ITIs, providing mechanisms for eliminating problems, which are difficult to overcome using current ad-hoc design processes. We propose an ITI language whose syntax and semantics are formally defined. This language comprises a syntax that defines a concrete notation used for modeling and the capability of describing an entire ITI with the relations among components.

The concept of ITI patterns was introduced to mitigate some of the challenges in the design of ITIs such as providing a vendor-independent approach to ITI design.

We also added the possibility of checking models and evaluating existing ITIs though the use of well-formedness rules and best-practices rules defined in OCL. With these proposals, we were able to reuse knowledge and create more robust ITIs based on proven solutions.

We realized that as important as the creation of ideas to overcome the current state of the art in the design of ITIs is the feasibility of that idea. Therefore, to allow infrastructure architects and other professionals responsible for designing ITIs to use our proposals, we extended a modeling tool to operationalize our proposals. The next section will provide more details regarding the contributions of each chapter.

6.2 Thesis Synthesis

As presented in chapter 1, grounded in the research problems and research questions, the purpose of the dissertation is to address the informal notation used to represent ITIs, overcome the inability to express ITI best practices and provide computational support to operationalize the previous contributions.

In chapter 2 we performed a literature review to characterize the current state of the art of ITIs in terms of the use of formal ITI design methods, how can they be assessed in terms of their size, level of complexity and whether certain policies have been applied, among other assessments.

To validate and complement the literature review, we performed also a systematic literature review in chapter 3. The protocol allowed us to retrieve 706 publications about ITIs from a specific set of databases. We concluded that none has a modeling approach defined for ITIs that can be used simultaneously to evaluate several dimensions of the infrastructure such as complexity and the definition of WFR.

In chapter 4, we compared candidate metamodels to fulfill our needs. We found out that the best approach was to extend UML and therefore create an ITI metamodel that we called itiPM. We decided to extend the Sparx EA modeling tool to incorporate our proposals and we described the creation of an ITI profile with an abstract syntax (UML metamodel extension) and a concrete syntax (notation). We then focused our attention in having the itiPM in a tool with better support for OCL called USE, so we can enrich the model with rules, constraints and operations to guarantee metamodel consistency, and to allow several types of ITI evaluations such as detecting any violations of ITI best practices, evaluating complexity metrics such as detecting the ITI size, among other possible analysis. We demonstrate the assessment with a real ITI from a large organization.

The chapter 5, was dedicated to patterns where the presented the pattern catalog and some examples of ITI patterns successfully published. We finalized the chapter with a detailed description on how to create ITI patterns and how to use them and an instantiation of an ITI pattern for an organization with components from several manufacturers.

In chapter 6, we describe the application of an online survey to validate the problems addressed by this thesis as well as its contributions. We had 123 infrastructure experts around the world confirming the problems and confirming the importance of our contributions to decrease design effort, facilitate knowledge sharing, by means of a repository of ITI patterns. The majority of the respondents agreed that *(i)* using our proposal would allow obtaining better documented ITIs, *(ii)* the approach can be used by non ITI experts and *(iii)* the importance of these diagrams to support implementation activities. We described also a controlled experiment performed with our proposals and ITI experts, where we were able to confirm the results of the online survey and the importance of our research proposals.

6.3 Main Contributions

This section presents a summary of the three main contributions for the design of ITIs proposed in this dissertation:

- **An ITI modeling language.** Using UML extensibility capabilities, we created an ITI profile (abstract syntax) using mechanisms such as stereotypes, tagged values and constraints. We also proposed a concrete representation (ITI notation) to directly address the problems of ambiguity, traceability and documentation among others, created by the use of *ad-hoc* approaches in the ITI design (research problem 1). The objective of the ITI profile is the definition of a UML-compliant extension, which supports ITI concepts.

- **ITI patterns.** The use of patterns brings to the domain of ITIs a well-known formal approach to capture, store and sharing domain knowledge, using provider independent solutions. These contributions address directly the problems of expressing and using best practices in ITI design, which corresponds to our research problem 2 (as expressed in section 5.2).

- **ITI design operationalization.** To enable ITI architects, consultants, and administrators to use the previously mentioned approaches and contributions, a modeling tool was extended to include the ITI profile and ITI patterns to simplify the ITI design process. This contribution addresses the research problem 3 (identified in section 5.2).

A systematic literature review (SLR) performed in the realm of ITIs based upon the research questions and a set of inclusion and exclusion criteria. This SLR is important because we want to provide evidence that this research work is relevant and not conducted before. The research databases, publications among other materials used in the SLR is in appendix A. The result of the latter is expected to provide us a clear view of the current state of the art in the ITI modeling and, as such discuss the relevance of the aforementioned main contributions. The systematic literature review itself when concluded and published is expected to be a relevant contribution to the community since we are not aware of any related secondary study in the field of ITI modeling.

6.4 Research Opportunities

In terms of future research work we intend to submit the ITI profile to OMG. According to the feedback received, it is crucial to find a sponsor from the industry to submit the profile. This initiative implies changing the ITI profile, to cope with OMG standards requirements.

We also envisage the development of a new ITI modeling tool (ideally using open source technologies) that includes the ITI profile and patterns built-in and also an extended toolbox with high quality images associated to each stereotype defined in the metamodel. The later, was intended to address the high number of users (according to the online survey and our experience) that use Microsoft Office tools such as PowerPoint and Visio to design ITIs due to their graphical capacities.

We have identified other research opportunities that we think deserve to be further explorer. Each of the opportunities is detailed next.

6.4.1 ITI Total Cost of Ownership

The concept of Total Cost of Ownership (TCO) was popularized by Gartner in 1986 [KM05], with the goal of clearly and reasonably addressing the real costs attributed to owning and managing ITIs. TCO identifies costs as being made of two main groups, the direct costs and the indirect costs (aka soft costs [Kir03a] because they often occur outside the budget). The direct costs are normally the capital, fees and labor costs. Indirect costs are more difficult to measure and include the costs associated with training IT professionals and users, costs associated with failure or outage (planned and unplanned), development and testing, costs associated with distributed computing, datacenters, storage and telecommunications, electricity and much more [Kir03a]. The nature of indirect costs leads some organizations to underestimate their impact on ITIs. However, TCO analysis often shows that the acquisition or purchase price of an asset represents only a small fraction of its total cost and indirect costs can typically represent as much as 60% of the total cost of managing and owning an ITI [KM05]. The TCO allows the alignment of IT operational efficiency goals with business performance requirements [Kir03b] and should not be used with the purpose of justifying IT investments, validating initiatives or, increasing or decreasing ITIs spending. There are some frequent misunderstandings, such as that TCO is only a way of cutting costs or that the IT platform with the lowest TCO is the best choice and indirect costs do not count [Kir03a]. The TCO has proven to be a vital and popular framework for IT and business management decision-making and has been applied to different technology areas. The idea of using TCO as a way to gauge IT performance has taken shape and may be used as a proxy for Activity-based costing (ABC), a technique in which all costs associated with a specific IT function are measured and compared to industry averages. This comparison is sometimes more efficient than looking at IT costs at the macro level. For instance, a TCO analysis of the costs associated with managing Enterprise Resource Planning (ERP) applications could reveal cost disparities that would not show up if a company only considered its overall IT costs [Kir03b].

There has been an increasing interest in recent years in calculating and reducing the costs of ownership with the aim of helping Chief Information Officers (CIOs) to make better decisions as they purchase, upgrade and/or replace their ITIs [Eng06; Mac03]. Effectively managing the ITI TCO is an open research opportunity. Effectively, means not only being able to identify, calculate or reduce ITI TCO, but also find the right balance between TCO and agility of the ITI.

We believe, that the approach for modeling ITIs proposed in this dissertation is a goal starting point to address this research opportunity.

6.4.2 Mining ITI Patterns

Besides being general solutions to commonly problems, the ITI patterns can be seen as documentation for a subset of the ITI. As was confirmed by the online survey, documentation is very important and most times, the only documentation available is an ad-hoc picture or diagram. According to our experience, sometimes when everything else fails, the documentation may be the last resort. There are exceptions, but in many cases the documentation is the last deliverable of projects and when available is incomplete, have low quality and is badly organized. Many projects rely on existent documentation and if well written can decrease the probability of human errors. As an example [HA08], rely on existing documentation to manually study architecture documentation of 47 systems with the only purpose of learn and identify the most common patterns deployed. Without documentation, this study cannot be performed.

The research opportunity, consists in mitigating the aforementioned problem of poor or inexistent documentation, by crawling ITIs and discovering ITI patterns deployed without the help of an ITI architect. The information gathered can contribute to a better documentation of ITIs and can also increase the number of patterns in the repository.

Notice that the vast majority of survey respondents, confirmed the importance of having a knowledge repository with well-known solutions to recurring problems. This research implies creating, adapting or using existing mining algorithms adapted to the domain of ITIs with the objective of implementing a methodological approach for mining ITI patterns in models of large and complex ITIs. We believe that the ITI profile and patterns developed in this dissertation, can be used to support this research. Another contribution that can arise from this research, is the ITI profile enrichment with the data obtained in ITIs. As an example, we were able to find in the CMDB of a large organization, 380 different types of routers.

6.4.3 Refactoring to ITI patterns

To accommodate new requirements the ITI needs to be modified and adapted constantly. To avoid accidental requirements changes, a systematic and safe approach to modifying ITI is needed. ITI patterns provide a high degree of design flexibility for such accidental requirements changes.

The research opportunity, consists in the proposal of an automated approach to refactoring based on ITI design patterns. A possible research approach could be to obtain ITI configurations and then propose (based on ITI patterns repository) the ITI areas in need of pattern-directed refactoring and the candidate ITI patterns.

The organizations that manage complexity are normally in a better position when compared with their competitors [AK17]. To be able to address new business needs and predicting infrastructure complexity, organizations, should measure their agility with focus on ITIs and then making the required IT investments [Plu05]. Agility is the ability of ITIs to adapt to business condition. Agility is typically expressed by the required time to implement a new capability or to achieve an IT goal such as increase the IT capacity by 15 percent to support a new business application [Mer+08].

6.4.4 ITI Complexity Evaluation

The rapid change in ITIs had a drastic impact on their complexity since moving from different eras, such as mainframes to cloud computing, is not straightforward and, as mention before, several organizations are still dependent of services and processes supported by old technologies. Due to the innovation gap between technologies, strategies to reduce complexity such as migrations and consolidations are sometimes difficult to achieve. Most of these services are critical to organizations and are completely disconnected from the rest of ITI, which results in unnecessary expenses, hidden costs, and an increase in complexity.

The complexity of any system has several drivers of which the most important are *(i)* size, *(ii)* the diversity and *(iii)* the mutation of its parts and of their interconnections. Often, we have to drill down complexity analysis, since each component of a system may itself be considered a (sub) system. We stop drilling down when a component can be considered as a blackbox. An ITI is a special kind of a system. Its parts are software and hardware components. Software components range from applications down to firmware (embedded software). However, components can be computer devices (e.g. desktop, handled or mobile devices), servers, switching and communication equipment (e.g. hubs, routers, access points, repeaters) and other devices (e.g. printers, plotters, scanners). While the size driver of ITI complexity is self-explanatory, it may not be so obvious for the diversity driver. The diversity driver of ITI components can manifest itself in different installation operations and maintenance procedures. Consider, for instance, two ITIs, with the same number of servers, equipment and topology. The complexity of those two ITIs, will be much different if, in one case, there is no technology diversity and, on the other case, each component requires specific customization or operation. Just imagine that you have 10 different printers each requiring a different kind of maintenance intervention. The mutation driver of ITI complexity has to do with its modifications throughout time. The observation period may vary depending on the characteristic being observed. For instance, when computing the percentage of PCs replaced, a yearly basis would make sense, while we may need to observe the maximum number of transactions per hour for balancing online versus offline services.

Complex ITIs are not easy to manage and due to the multitude of services supported by ITIs they are more complex to manage today than ever. Today employees bring their personal devices inside organizations[1] and use them with organizational data which forces IT departments to manage all these different devices. Managing all these different devices and providing IT services to be delivered and consumed in a number of different ways represent a huge increase in the complexity and is one of the main concerns of CIOs. An independent market research company, Vanson Bourne, surveyed 200 UK CIOs and senior IT decision makers from large enterprises with over 1,000 employees and found that the majority of CIOs (86%) feel ITI has become more complex over the last five years [Tru16]. Understanding the ITIs complexity is achievable and must be easily determined (static perspective) and kept under control (dynamic or evolutive perspective). In both cases we need to express that complexity quantitatively.

A step forward in this research topic was already performed, but considered an ITI as a network of graphs and the resulting complexity was only based on the number and size of graphs and relations among them. The approach, used network analysis, graph theory and the following three well known complexity metrics:

- **Coefficient of Network Complexity (CNC)** [Pas66] which is a widely used metric for evaluating network complexity in the field of network analysis.
- **Cyclomatic Complexity Metric (CCM)** [McC76] proposed by Tom McCabe, which was one of the first metrics to evaluate the complexity of flowcharts (directed graphs) representing the internal implementation of individual software modules.
- **Henry and Kafura Metric (HKM)** [HK81] used to evaluate the complexity of software modules (nodes) of the so-called "call graph" (a directed graph representation of the call relationships among those modules).

Extend this work by proposing new complexity metrics or adapting existing ones, considering other ITI dimensions besides the network is an important research topic. Other ITI dimensions to evaluate ITI complexity can be hardware, software, number of different manufacturers, level of standardization are just some examples. The organizations that manage complexity are normally in a better position when compared with their competitors [AK17]. To be able to address new business needs and predicting infrastructure complexity, organizations, should measure their agility with focus on ITIs and then making the required IT investments [Plu05]. Agility is the ability of ITIs to adapt to business condition. Agility is typically expressed by the required time to implement a new capability or to achieve an IT goal such as increase the IT capacity by 15 percent to support a new business application [Mer+08].

[1] often known as Bring Your Own Device (BYOD) approach

Bibliography

[Abr97] F. B. Abreu. "Pedagogical Patterns: Picking up the design patterns approach." In: *Object Expert* 2.3 (1997), pp. 37–41.

[Abr98] F. B. Abreu. "Padrões Pedagógicos - Um contributo das tecnologias de informação para o ensino." In: *Proceedings of the VIII Encontro das Universidades de Língua Portuguesa*. Ed. by A. das Universidades de Língua Portuguesa (AULP). Macau, Apr. 1998.

[AC94] F. B. Abreu and R. Carapuça. "Object-Oriented Software Engineering: Measuring and Controlling the Development Process." In: *Proc. Int'l Conf. Software Quality*. Ed. by A. S. for Quality (ASQ). Oct. 1994.

[Abr+99] F. B. Abreu, L. Ochoa, and M. Goulão. "The GOODLY design language for MOOD2 metrics collection." In: *Proceedings of the Workshop on Object-Oriented Technology*. Ed. by S. Demeyer and J. Bosch. London, UK: Springer-Verlag, 1999, pp. 328–329. ISBN: 3-540-66954-X. URL: http://dl.acm.org/citation.cfm?id=646779.705122.

[Ado+06] G. Adomavicius, J. Bockstedt, A. Gupta, and R. J. Kauffman. "Understanding patterns of technology evolution: An ecosystem perspective." In: *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*. Ed. by R. Sprague. Vol. 8. Jan. 2006, 189a–189a. DOI: 10.1109/HICSS.2006.515.

[AK17] D. Alamoudi and A. Kumar. "Information System Complexity and Business Value." In: *Int Journal Econ Manag Sci* 6.2 (2017). DOI: 10.4172/2162-6359.1000400.

[Ald+03] O. Aldawud, T. Elrad, and A. Bader. "UML profile for aspect-oriented software development." In: *Proceedings of the Third International Workshop on Aspect Oriented Modeling*. Ed. by N. Nunes, B. Selic, A. Silva, and A. Alvarez. Mar. 2003.

[Ale65] C. Alexander. *A city is not a tree*. Vol. 122. 1. Sustasis Foundation, 1965, p. 241. ISBN: 978-0-98-934697-9.

[Ale79] C. Alexander. *The timeless way of building*. Oxford University Press, 1979, pp. 1–551. ISBN: 978-0-19-502402-9.

[Ale+77] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel. *A Pattern Language - Towns, Buildings, Construction*. Oxford University Press, 1977, pp. 1–1171. ISBN: 978-0-19-501919-3.

[Ale+75] C. Alexander, M. Silverstein, S. Angel, S. Ishikawa, and D.Abrams. *The Oregon Experiment*. Oxford University Press, 1975, pp. 1–202. ISBN: 9780195 018240.

[Ale64] C. Alexander. *Notes on the Synthesis of Form*. Harvard Univ. Press, Jan. 1, 1964, p. 224. ISBN: 9780674627512.

[Anc+11] P. Anchuri, M. J. Zaki, O. Barkol, R. Bergman, Y. Felder, S. Golan, and A. Sityon. "Infrastructure pattern discovery in configuration management databases via large sparse graph mining." In: *2011 IEEE 11th International Conference on Data Mining*. Ed. by R. Stahlbock and S. Crone. Dec. 2011, pp. 11–20. DOI: 10.1109/ICDM.2011.81.

[And+03] D. Andrews, B. Nonnecke, and J. Preece. "Electronic survey methodology: A case study in reaching hard-to-involve internet users." In: *International Journal of Human-Computer Interaction* (2003), pp. 185–210.

[Arp+11] T. Arpinen, T. D. Hämäläinen, and M. Hännikäinen. "Metamodel and UML profile for requirements management of software and embedded systems." In: *EURASIP Journal on Embedded Systems* (Mar. 2011), pp. 1–14. DOI: 10.1155/2011/592168.

[Bal+07] A. Baldwin, Y. Beres, and S. Shiu. "Using assurance models to aid the risk and governance life cycle." In: *BT Technology Journal* 25.1 (2007), pp. 128–140. ISSN: 1573-1995. DOI: 10.1007/s10550-007-0015-7.

[BA02] A. L. Baroni and F. B. Abreu. "Formalizing object-oriented design metrics based upon the UML meta-model." In: *Actas do XVI Simpósio Brasileiro de Engenharia de Software*. Ed. by R. Price. Sociedade Brasileira de Computação, 2002, pp. 130–145.

[Bar+06] A. L. Baroni, C. Calero, F. B. Abreu, and M. Piattini. "Object-relational database metrics formalization." In: *Proceedings of the Sixth International Conference on Quality Software (QSIC'2006)*. Ed. by H. Mei. IEEE Computer Society, 2006, pp. 30–37.

[Bha+10] G. Bhatt, A. Emdada, N. Roberts, and V. Grover. "Building and leveraging information in dynamic environments: The role of IT infrastructure flexibility as enabler of organizational responsiveness and competitive advantage." In: *Information & Management* (2010). DOI: 10.1016/j.im.2010.08.001.

[Bin99] R. V. Binder. *Testing object-oriented systems: models, patterns, and tools*. Object Technology Series. Boston, MA, USA: Addison-Wesley, 1999. ISBN: 0-201-80938-9.

[Bio+07] J. C. A. Biolchini, P. G. Mian, A. C. C. Natali, T. U. Conte, and G. H. Travassos. "Scientific research ontology to support systematic review in software engineering." In: *Advanced Engineering Informatics* 21.2 (2007), pp. 133–151. ISSN: 1474-0346. DOI: 10.1016/j.aei.2006.11.006.

[Boa98] B. H. Boar. *Constructing blueprints for enterprise IT architectures*. Wiley computer publishing. Wiley, 1998. ISBN: 9780471296201.

[Bos+05] M. Boskovic, D. Gasevic, and V. Devedzic. "PatternGuru: An educational system for software patterns." In: *International Conference on Advanced Learning Technologies*. Ed. by PatternGuru: 2005, pp. 650–654. DOI: 10.1109/ICALT.2005.218.

[BF95] L. Bourque and E. Fielder. *How to Conduct Self-Administered and Mail Surveys*. SAGE Publications, 1995. ISBN: 9780761925620. DOI: 10.4135/9781412984430.

[Bri+02] N. Britten, R. Campbell, C. Pope, J. Donovan, M. Morgan, and R. Pill. "Using meta ethnography to synthesise qualitative research: a worked example." In: *Journal of Health Services Research & Policy* 7 (4 2002), pp. 209–215. DOI: 10.1258/135581902320432732.

[Bro+99] M. Broadbent, P. Weill, and B. S. Neo. "Strategic context and patterns of IT infrastructure capability." In: *The Journal of Strategic Information Systems* 8.2 (1999), p. 20. DOI: 10.1016/S0963-8687(99)00022-0.

[Bus+96] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-oriented software architecture: a system of patterns*. New York, NY, USA: John Wiley & Sons, Inc., 1996. ISBN: 0-471-95869-7.

[Cal+11] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, and R. Buyya. "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." In: *Softw. Pract. Exper.* 41.1 (2011), pp. 1–28. DOI: 10.1002/spe.995.

[Can11] D. Cannon. *IT Infrastructure Library (ITIL) - service strategy (version 3)*. London: The Stationery Office, 2011. ISBN: 0113310455.

[Coc98] A. Cockburn. "Surviving your OO project." In: *TOOLS*. Ed. by I. C. Society. Santa Barbara, CA, USA, 1998, p. 462. ISBN: 0-8186-8482-8.

[Cop97] J. Coplien. "Idioms and patterns as architectural literature." In: *IEEE Software* 14.1 (1997), pp. 36–42. DOI: 10.1109/52.566426.

[CS95] J. O. Coplien and D. C. Schmidt. *Pattern languages of program design*. Addison-Wesley, 1995. ISBN: 978-0-201-60734-5.

[CB15] A. Correia and F. Brito e Abreu. "Enhancing the Correctness of BPMN Models." In: *Improving Organizational Effectiveness with Enterprise Information Systems*. Ed. by J. Varajão. 1st ed. Vol. 1. Advances in Business Information Systems and Analytics (ABISA). Hershey, PA, USA: IGI-Global, 2015, pp. 241–241. URL: http://www.igi-global.com/chapter/enhancing-the-correctness-of-bpmn-models/133100.

[CP99] S. Cranefield and M. K. Purvis. "UML as an ontology modelling language." In: *Intelligent Information Integration*. Ed. by S. Cranefield and M. K. Purvis. Vol. 23. CEUR Workshop Proceedings. 1999. URL: http://ceur-ws.org/Vol-23/cranefield-ijcai99-iii.pdf.

[CRD09] CRD. *Systematic Reviews: CRDs guidance for undertaking reviews in health care*. York, England: Centre for Reviews and Dissemination, University of York, 2009. ISBN: 978-1-900640-47-3.

[CB86] W. Cunningham and K. Beck. "A diagram for object-oriented programs." In: *SIGPLAN Not.* 21.11 (June 1986), pp. 361–367. ISSN: 0362-1340. DOI: 10.1145/960112.28734.

[Dat09] R. Datta. *Cloud Networking: Design Patterns for "Cloud-Centric"Application Environments*. Tech. rep. Arista Networks, June 2009, pp. 1–10. (Visited on 02/25/2017).

[DMT99] DMTF. *Common Information Model (CIM) specification*. Distributed Management Task Force. June 1999. URL: http://www.dmtf.org/standards/cim (visited on 02/25/2017).

[DD08] T. Dyba and T. Dingsoyr. "Strength of evidence in systematic reviews in software engineering." In: *Empirical Software Engineering and Measurement*. Ed. by S. Elbaum. 2008, pp. 178–187. DOI: 10.1145/1414004.1414034.

[Dyb+07] T. Dyba, T. Dingsoyr, and G. K. Hanssen. "Applying systematic reviews to diverse study types: An experience report." In: *Empirical Software Engineering and Measurement*. Ed. by S. Elbaum. 2007, pp. 225–234. DOI: 10.1109/ESEM.2007.21.

[Dyb+08] T. Dyba, T. Dingsoyr, and G. K. Hanssen. "Empirical studies of agile software development: A systematic review." In: *Information and Software Technology* 50 (9-10 2008), pp. 833–859. DOI: 10.1016/j.infsof.2008.01.006.

[Ear89] M. J. Earl. *Management strategies for Information Technology*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. ISBN: 0-13-551664-1.

[EB09] A. F. Emdad and G. D. Bhatt. "An empirical examination of the relationship between information technology (IT) infrastructure, customer focus, and business advantages." In: *Journal of Systems and Information Technology* 12.1 (2009), pp. 4–16. DOI: 10.1108/13287261011032625.

[ES14] R. J. Engel and R. K. Schutt. *Fundamentals of social work research - second edition*. SAGE Publications, Inc., 2014. ISBN: 9781506352039.

[Eng06] M. Engels. "Reducing the total cost of ownership." In: *Broadband Fixed Wireless Access*. Springer, 2006, pp. 163–183. DOI: 10.1007/978-0-387-34593-2_8.

[Eva01] N. D. Evans. *Business agility: Strategies for gaining competitive advantage through mobile business solutions*. Financial Times/Prentice Hall, 2001. ISBN: 0130668370.

[Ewi86] J. J. Ewing. "An object-oriented operating system interface." In: *SIGPLAN Not.* 21.11 (June 1986). Ed. by ACM, pp. 46–56. ISSN: 0362-1340. DOI: 10.1145/960112.28703.

[Fin03] A. Fink. *The survey handbook*. Ed. by A. Fink. 0th ed. SAGE Publications, Inc., 2003. DOI: 10.4135/9781412986328.

[Fow97] M. Fowler. *Analysis patterns - reusable object models*. Addison-Wesley series in object-oriented software engineering. Addison-Wesley, 1997. ISBN: 978-0-201-89542-1.

[Fre+15] F. J. Frey, R. Bijvank, and M. Pöttker. "Weaving in Patterns into It Infrastructure Models: Industry Case and Exemplary Approaches." In: *Proceedings of the 20th European Conference on Pattern Languages of Programs*. EuroPLoP '15. Kaufbeuren, Germany: ACM, 2015, 28:1–28:25. ISBN: 978-1-4503-3847-9. DOI: 10.1145/2855321.2855350. URL: http://doi.acm.org/10.1145/2855321.2855350.

[Fue+03] J. M. Fuentes, V. Quintana, J. Llorens, G. Génova, and R. Prieto-Diaz. "Errors in the UML metamodel?" In: *ACM SIGSOFT Software Engineering Notes* 28.6 (6 2003), pp. 3–3. DOI: 10.1145/966221.966236.

[Gal08a] D. Galorath. *Towards a standard definition of IT Infrastruture*. Accessed on: August 2016. Sept. 2008. URL: http://galorath.com/wp/towards-a-standard-definition-of-it-infrastruture/.

[Gal08b] D. Galorath. *What is IT Infrastructure: A definition and a taxonomy*. Accessed on: July 2016. Sept. 2008. URL: http://galorath.com/wp/what-is-it-infrastructure-a-definition-and-a-taxonomy/.

[Gam+94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of reusable object-oriented software*. 1st ed. Addison-Wesley Professional, Nov. 1994. ISBN: 0201633612.

[Gar13] Gartner. *IT Infrastructure Definition | Gartner*. Feb. 2, 2013. URL: http://www.gartner.com/it-glossary/it-infrastructure (visited on 08/01/2016).

[Ger+10] A. Gerber, P. Kotzé, and A. V. Merwe. "Towards the formalisation of the TOGAF content metamodel using ontologies." In: *Proceedings of International Conference on Enterprise Information Systems*. Ed. by J. Filipe and J. Cordeiro. 2010, pp. 54–64. DOI: 10.5220/0002903200540064.

[Ghi+11] A. Ghiran, G. C. Silaghi, and N. Tomai. "Deploying an agent platform to automate the IT infrastructure auditing process." In: *Business Information Systems: 14th International Conference, BIS 2011, Poland, June 15-17, 2011. Proceedings*. Ed. by W. Abramowicz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 266–277. ISBN: 978-3-642-21863-7. DOI: 10.1007/978-3-642-21863-7_23.

[Gog+07] M. Gogolla, F. Büttner, and M. Richters. "USE: A UML-based specification environment for validating UML and OCL." In: *Science of Computer Programming* 69 (2007), pp. 27–34. DOI: 10.1016/j.scico.2007.01.013.

[Gog+02] M. Gogolla, A. Lindow, M. Richters, and P. Ziemann. "Metamodel transformation of data models." In: *Proceedings of the Workshop in Software Model Engineering*. Ed. by J. Bezivin and R. France. 2002.

[GA05] M. Goulão and F. B. e. Abreu. "Formal Definition of Metrics Upon the CORBA Component Model." In: *Quality of Software Architectures and Software Quality: First International Conference on the Quality of Software Architectures, QoSA 2005, and Second International Workshop on Software Quality, SOQUA 2005, Erfurt, Germany, September 20-22, 2005. Proceedings*. Ed. by R. Reussner, J. Mayer, J. A. Stafford, S. Overhage, S. Becker, and P. J. Schroeder. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 88–105. ISBN: 978-3-540-32056-2. DOI: 10.1007/11558569_8.

[GA07] M. Goulão and F. B. e Abreu. "An overview of metrics-based approaches to support software components reusability assessment." In: *Software Quality Measurements: Concepts and Approaches*. Vol. abs/1109.6802. 2007. URL: http://arxiv.org/abs/1109.6802.

[GR08] N. Gronau and M. Rohloff. "Information Systems Implementation: The Big Picture." In: *Proceedings of the 2008 ACM Symposium on Applied Computing*. Ed. by N. Gronau and M. Rohloff. SAC '08. Fortaleza, Ceara, Brazil: ACM, 2008, pp. 1077–1078. ISBN: 978-1-59593-753-7. DOI: 10.1145/1363686.1363937.

[Ham98] M. Hammersley. *Reading ethnographic research: a critical guide*. Longman social research series. Longman, 1998. ISBN: 9780582311046.

[HP12a] V. Haren and V. H. Publishing. *ArchiMate 2.0 specification*. The Open Group. Van Haren Publishing, 2012. ISBN: 9789087536923.

[Har11] V. Haren. *TOGAF Version 9.1*. 10th. Van Haren Publishing, 2011. ISBN: 9087536798.

[HA08] N. Harrison and P. Avgeriou. "Analysis of architecture pattern usage in legacy system architecture documentation." In: *Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*. WICSA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 147–156. ISBN: 978-0-7695-3092-5. DOI: 10.1109/WICSA.2008.18. URL: https://doi.org/10.1109/WICSA.2008.18.

[HV93] J. C. Henderson and n. Venkatraman. "Strategic Alignment: Leveraging Information Technology for Transforming Organizations." In: *IBM Syst. J.* 32.1 (1993), pp. 472–484.

[HK81] S. Henry and D. Kafura. "Software structure metrics based on information flow." In: *Transactions on Software Engineering* 7.5 (1981), pp. 510–518. DOI: 10.1109/TSE.1981.231113.

[HG11] J. Higgins and S. Green. *Cochrane Handbook for Systematic Reviews of Interventions*. Version 5.1.0. Oxford: The Cochrane Collaboration, 2011. URL: http://handbook-5-1.cochrane.org/ (visited on 02/25/2017).

[Hol+14] H. Holm, M. Buschle, R. Lagerström, and M. Ekstedt. "Automatic Data collection for enterprise architecture models." In: *Softw. Syst. Model.* 13.2 (2014), pp. 825–841. ISSN: 1619-1366. DOI: 10.1007/s10270-012-0252-1.

[HP12b] HP. *HP Converged Infrastructure Reference Architecture Design Guide*. Tech. rep. 4AA3-4935ENW. HP, Aug. 2012. URL: http://www.datacenterknowledge.com/archives/2013/04/15/hp-converged-infrastructure-reference-architecture-design-guide (visited on 02/25/2017).

[HB04] J. M. Hudson and A. Bruckman. "Go Away: Participant objections to being studied and the ethics of chatroom research." In: *The Information Society* 20.2 (2004), pp. 127–139. ISSN: 01972243. DOI: 10.1080/01972240490423030.

[Hun11] L. Hunnebeck. *IT Infrastructure Library (ITIL) - service design (version 3)*. London: The Stationery Office, 2011. ISBN: 0113313055.

[Jed+08] A. Jedlitschka, M. Ciolkowski, and D. Pfahl. "Reporting experiments in software engineering." English. In: *Guide to Advanced Empirical Software Engineering*. Ed. by F. Shull, J. Singer, and D. Sjoberg. Springer London, 2008, pp. 201–228. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5_8.

[Kee91] P. G. W. Keen. *Shaping the future: Business design through information technology*. Boston, MA, USA: Harvard Business School Press, 1991. ISBN: 0-87584-237-2.

[Ken07] M. M. Kennedy. "Defining a literature." In: *Educational Researcher* 36 (3 2007), pp. 139–147. DOI: 10.3102/0013189X07299197.

[Kha11] Y. Khalidi. "Building a cloud computing platform for new possibilities." In: *IEEE Computer* 44.3 (2011), pp. 29–34. doi: 10.1109/MC.2011.63.

[Kir03a] B. Kirwin. *CIO update: To control TCO, it must be measured and managed.* InSide Gartner. Gartner, Apr. 16, 2003. url: http://www.bus.umich.edu/KresgePublic/Journals/Gartner/research/114300/114326/114326.pdf (visited on 02/25/2017).

[Kir03b] B. Kirwin. *Management update: Total cost of ownership analysis provides many benefits.* Report IGG-08272003-01. Gartner, 2003.

[KM05] B. Kirwin and L. Mieritz. *Control distributed computing TCO by keeping an eye on complexity creep.* Report G00129518. Gartner, Aug. 2005. url: https://www.gartner.com/doc/484357/control-distributed-computing-tco-keeping (visited on 02/25/2017).

[Kit+02] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg. "Preliminary guidelines for empirical research in software engineering." In: *IEEE Transactions on Software Engineering* 28.8 (Aug. 2002), pp. 721–734. doi: 10.1109/TSE.2002.1027796.

[KC07] B. Kitchenham and S. Charters. "Guidelines for performing systematic literature reviews in software engineering." In: *Technical report, Ver. 2.3 EBSE Technical Report. EBSE.* Ed. by D. Budgen, P. Brereton, M. Turner, and S. Linkman. Keele University and Durham University Joint Report, 2007. url: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf.

[KP02a] B. Kitchenham and S. L. Pfleeger. "Principles of survey research - Part 2: Designing a survey." In: *SIGSOFT Softw. Eng. Notes* 27.1 (Jan. 2002), pp. 18–20. issn: 0163-5948. doi: 10.1145/566493.566495.

[KP02b] B. Kitchenham and S. L. Pfleeger. "Principles of survey research - Part 3: Constructing a survey instrument." In: *SIGSOFT Softw. Eng. Notes* 27.2 (Mar. 2002), pp. 20–24. issn: 0163-5948. doi: 10.1145/511152.511155.

[KP02c] B. Kitchenham and S. L. Pfleeger. "Principles of survey research - Part 4: Questionnaire evaluation." In: *SIGSOFT Softw. Eng. Notes* 27.3 (May 2002), pp. 20–23. doi: 10.1145/638574.638580.

[KP02d] B. Kitchenham and S. L. Pfleeger. "Principles of survey research - Part 5: Populations and samples." In: *SIGSOFT Softw. Eng. Notes* 7.5 (Sept. 2002), pp. 17–20. doi: 10.1145/571681.571686.

[Knu84] D. Knuth. *Literate Programming.* Centre for the Study of Language & Information, 1984. isbn: 0937073806.

[Krü12] P. Krüger. "Feature modeling: An extended perspective to design IT infrastructure." In: *Proceedings of the Conf-IRM* (2012). URL: http://aisel.aisnet.org/confirm2012 (visited on 02/25/2017).

[Krü+12] P. Krüger, T. Urban, A. Siegling, R. Zimmermann, and H. Arndt. "Conceptual methods to design sustainable IT Infrastructures-standardization, consolidation, and virtualization." In: *EnviroInfo* (2012).

[Lea94] D. Lea. "Christopher Alexander: An Introduction for Object-oriented Designers." In: *SIGSOFT Softw. Eng. Notes* 19.1 (Jan. 1994), pp. 39–46. ISSN: 0163-5948. DOI: 10.1145/181610.181617. URL: http://doi.acm.org/10.1145/181610.181617.

[Lin+02] J. Lin, M. Thomsen, and J. A. Landay. "A visual language for sketching large and complex interactive designs." In: *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems: Changing our World, Changing ourselves*. Ed. by D. Wixon. Minneapolis, Minnesota, USA: ACM, 2002, pp. 307–314. ISBN: 1-58113-453-3. URL: http://doi.acm.org/10.1145/503376.503431.

[LK05] B. List and B. Korherr. "A UML 2 profile for business process modelling." In: *Proceedings of Perspectives in Conceptual Modeling*. Ed. by J. Akoka, S. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J. Heuvel, M. Kolp, J. Trujillo, C. Kop, and H. Mayr. Vol. 3770. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 85–96. ISBN: 978-3-540-29395-8. DOI: 10.1007/11568346_10.

[Llo11] V. Lloyd. *IT Infrastructure Library (ITIL) - continual service improvement (version 3)*. London: The Stationery Office, 2011. ISBN: 0113310498.

[Mac03] A. MacCormack. "Evaluating total cost of ownership for software platforms: Comparing apples, oranges and cucumbers." In: *AEI-Brookings Joint Center for Regulatory Studies Related Publication* 9 (2003), pp. 1–11.

[McC76] T. J. McCabe. "A complexity measure." In: *IEEE Transactions on Software Engineering* 4 (1976), pp. 308–320. DOI: 10.1109/TSE.1976.233837.

[MH04] D. L. McGuinness and F. V. Harmelen. *OWL web ontology language overview*. Tech. rep. W3C, Feb. 10, 2004. URL: http://www.w3.org/TR/owl-features/ (visited on 02/25/2017).

[MB89] D. T. McKay and D. W. Brockway. "Building IT Infrastructure for the 1990s." In: *Stage by stage* 9.3 (1989), pp. 1–11.

[Mer+08] K. Mertins, R. Ruggaber, K. Popplewell, and X. Xu. "An approach for the evaluation of the agility in the context of enterprise interoperability." In: *Enterprise Interoperability III*. Springer London, 2008, pp. 3–14. DOI: 10.1007/978-1-84800-221-0.

[MB64] D. M. Miller and W. R. Borg. "Educational research: An introduction." In: *American Educational Research Journal* (1964). DOI: 10.2307/1161989.

[Mon13] M. Montgomery. *IT Infrastructure assessment and staffing analysis*. Accessed on: August 2013. Mar. 2013. URL: http://wa-sequim.civicplus.com/DocumentCenter/View/3576.

[NW05] J. Noble and C. Weir. *Small Memory Software: Patterns for Systems with Limited Memory*. Addison-Wesley-Longman, 2005, pp. 1–333. ISBN: 978-0-201-59607-6.

[Nyr06] M. Nyrhinen. *IT Infrastructure: structure, properties and processes*. Working Paper 403. Helsinki School of Economics, June 2006. URL: http://aisel.aisnet.org/sprouts_all/140 (visited on 02/25/2017).

[Oat06] B. J. Oates. *Researching information systems and computing*. SAGE, 2006. ISBN: 9781412902243. URL: http://hdl.handle.net/10149/123247.

[OGC07] OGC. *Core publications for IT Infrastructure Library (ITIL) (version 3)*. Accessed on: August 2013. 2007. URL: http://www.itil-officialsite.com/Publications/Core.asp.

[OMG07] OMG. *MOF 2.0/XMI mapping, version 2.1.1*. Tech. rep. OMG, Dec. 2007. URL: http://www.omg.org/spec/XMI/2.1.1/ (visited on 02/25/2017).

[OMG08] OMG. *UML Infrastructure specification (version 2.4.1)*. Accessed on: August 2013. Object Management Group, Apr. 2008. URL: http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF.

[OMG09] OMG. *A UML profile for MARTE: Modeling and analysis of real-time embedded systems*. Accessed on: August 2013. Object Management Group, Apr. 2009. URL: http://www.omg.org/omgmarte/.

[Pal+12] G. K. Palshikar, A. L. Ahuja, and H. M. Vin. "Utilization analysis of servers in a data centre." In: *Proceedings of the Second International Conference on Data Engineering and Management*. Ed. by G. K. Palshikar, A. L. Ahuja, and H. M. Vin. ICDEM'10. Tiruchirappalli, India: Springer-Verlag, 2012, pp. 173–180. ISBN: 978-3-642-27871-6. DOI: 10.1007/978-3-642-27872-3_26.

[Pas66] T. L. Pascoe. "Allocation of resources-CPM." In: *Revue Française de Recherche Opérationelle* 38 (1966), pp. 31–38.

[PW95] W. W. Peng and D. R. Wallace. *Software Error Analysis*. Summit, NJ, USA: Silicon Press, Feb. 1, 1995. ISBN: 092930618X.

[PI13] T. Perroud and R. Inversini. *Enterprise architecture patterns : practical solutions for recurring IT-architecture problems*. Springer, 2013. URL: http://ci.nii.ac.jp/ncid/BB19156673.

[PG07] R. Perry and A. Gillen. *Demonstrating business value: Selling to your c-level executives*. Tech. rep. IDC, Apr. 2007. URL: https://aka.ms/vvduuh.

[Pet13] M. Petre. “UML in practice.” In: *Proceedings of the 2013 International Conference on Software Engineering*. Ed. by I. Press. ICSE ’13. 2013, pp. 722–731. ISBN: 978-1-4673-3076-3. DOI: 10.1109/ICSE.2013.6606618.

[PK01] S. L. Pfleeger and B. A. Kitchenham. “Principles of survey research - Part 1: Turning lemons into lemonade.” In: *SIGSOFT Softw. Eng. Notes* 26.6 (Nov. 2001), pp. 16–18. ISSN: 0163-5948. DOI: 10.1145/505532.505535.

[Plu05] D. Plummer. *Findings from the 'agility' research meeting: IT role in agility can be measured and improved*. Report. Gartner, Sept. 2005. URL: https://aka.ms/kgzxnb (visited on 02/25/2017).

[Qi+08] X. Qi, B. Lan, and Z. Guo. “Conceptual model of IT Infrastructure capability and its empirical justification.” In: *Tsinghua Science & Technology* 13.3 (2008), pp. 390–394. ISSN: 1007-0214. DOI: 10.1016/S1007-0214(08)70062-2.

[Ran11] S. Rance. *IT Infrastructure Library (ITIL) - service transition (version 3)*. London: The Stationery Office, 2011. ISBN: 978-0113313068.

[Roh08] M. Rohloff. “An Integrated View on Business- and IT-architecture.” In: *Proceedings of the 2008 ACM Symposium on Applied Computing*. Ed. by L. Liebrock. SAC ’08. Fortaleza, Ceará, Brazil: ACM, 2008, pp. 561–565. ISBN: 978-1-59593-753-7. DOI: 10.1145/1363686.1363822.

[Sau+09] J. P. Sauvé, A. Moura, and F. T. Marques. “Business-driven design of infrastructures for IT services.” In: *Journal of Network and Systems Management* 17.4 (2009), pp. 422–456. ISSN: 1573-7705. DOI: 10.1007/s10922-009-9140-6.

[Ses08] R. Sessions. *Simple architectures for complex enterprises*. Best practices. Microsoft Press, 2008. ISBN: 9780735625785.

[Sil08] L. F. Silva. “Assessment of IT Infrastructures: A model driven approach.” MSc thesis. Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Dec. 2008. URL: http://citi.di.fct.unl.pt/publication/dissertation.php?id=53.

[SA10a] L. F. Silva and F. B. Abreu. “An IT Infrastructure Patterns approach to improve IT service management quality.” In: *Proceedings of the 7th International Conference on the Quality of Information and Communications Technology*. Ed. by F. B. Abreu, J. Faria, and R. Machado. Nov. 2010, pp. 171–176. DOI: 10.1109/QUATIC.2010.34.

[SA10b] L. F. Silva and F. B. Abreu. “Patterns for IT Infrastructure Design.” In: *Proceedings of the 15th European Conference on Pattern Languages of Programs (EuroPLoP 2010)*. Ed. by P. A. M. Weiss. July 2010. DOI: 10.1145/2328909.2328934.

[SA10c] L. F. Silva and F. B. Abreu. "Reengineering IT Infrastructures: A method for topology discovery." In: *Proceedings of the 7th International Conference on the Quality of Information and Communications Technology*. Ed. by F. B. Abreu, J. Faria, and R. Machado. IEEE Computer Society, Sept. 2010, pp. 331–336. DOI: 10.1109/QUATIC.2010.62.

[SA11] L. F. Silva and F. B. Abreu. "Moving to the cloud: Estimating the internet connection bandwidth." In: *Proceedings of the 11 Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI 2011)*. Ed. by H. O. Neil. Associação Portuguesa de Sistemas de Informação, Oct. 2011. URL: http://nova-lincs.di.fct.unl.pt/publication/1138.

[Sil+12a] L. F. Silva, F. B. Abreu, and V. Moreira. "A UML Profile for Modeling IT Infrastructures." In: *Proceedings of the INFORUM 2012*. Ed. by J. Cachopo. INFORUM, Nov. 2012. URL: http://nova-lincs.di.fct.unl.pt/publication/1234.

[Sil+12b] L. F. Silva, F. B. Abreu, and V. Moreira. "Improving IT Infrastructures Representation: A UML Profile." In: *Proceedings of the Seventh International Conference on Software Engineering Advances (ICSEA 2012)*. Ed. by H. M. et al. Nov. 2012, pp. 459–464. URL: http://nova-lincs.di.fct.unl.pt/publication/1233.

[Smi03] B. Smith. *The blackwell guide to the philosophy of computing and information*. Ed. by L. Floridi. Blackwell Philosofy Guides. Blackwell Publishing, 2003. Chap. Ontology. ISBN: 0631229191.

[Sne12] B. Snell. *McAfee deep defender technology blueprint - fight rootkits*. Tech. rep. Intel, 2012. URL: http://www.mcafee.com/us/resources/technology-blueprints/tb-fighting-rootkits.pdf (visited on 02/25/2017).

[Sol+07] B. Solomon, D. Ionescu, M. Litoiu, and M. Mihaescu. "Towards a real-time reference architecture for autonomic systems." In: *Proceedings of the 2007 International Workshop on Software Engineering for Adaptive and Self-Managing Systems*. Ed. by B. Cheng, R. Lemos, S. Fickas, D. Garlan, M. Litoiu, J. Magee, and R. Taylor. SEAMS '07. IEEE Computer Society, 2007, pp. 10–14. ISBN: 0-7695-2973-9. DOI: 10.1109/SEAMS.2007.20.

[SN07] M. Staples and M. Niazi. "Experiences using systematic review guidelines." In: *Journal of Systems and Software* 80 (9 2007), pp. 1425–1437. ISSN: 0164-1212. DOI: 10.1016/j.jss.2006.09.046.

[Ste11] R. A. Steinberg. *IT Infrastructure Library (ITIL) - service operation (version 3)*. London: The Stationery Office, 2011. ISBN: 978-0113313075.

[Sul+12] B. Suleiman, S. Sakr, R. Jeffery, and A. Liu. "On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure." In: *Journal of Internet Services and Applications* 3.2 (2012), pp. 173–193. ISSN: 1869-0238. DOI: 10.1007/s13174-011-0050-y.

[Tre+13] R. Tretau, J. Ben-Bassat, and M. Chalogany. *Designing a reference architecture for virtualized environments using IBM system storage N series*. Tech. rep. IBM, Feb. 2013. URL: http://www.redbooks.ibm.com/technotes/tips0944.pdf (visited on 02/25/2017).

[Tro+03] D. Trowbridge, D. Mancini, D. Quick, G. Hohpe, and J. Newkirk. *Enterprise Solution Patterns using Microsoft .NET: version 2.0: Patterns and Practices*. Redmond, WA, USA: Microsoft Press, 2003. ISBN: 0735618399.

[Tru16] Trustmarque. *The CIO challenge: Simplifying IT support*. Trustmarque. May 2016. URL: https://www.trustmarque.com/it-support-report (visited on 02/25/2017).

[Tru10] F. Truyen. *Model driven architecture with enterprise architect: Measuring EA features to MDA concepts*. Accessed on: August 2013. Jan. 2010. URL: http://enterprisemodelingsolutions.com/wp-content/uploads/2017/05/EA4MDA_Features__WP.pdf.

[Tur91] P. D. Turnbull. "Effective investment in information infrastructures." In: *Information and Software Technology* 33.3 (1991), pp. 191–199. ISSN: 0950-5849. DOI: 10.1016/0950-5849(91)90133-V.

[Völ01] M. Völter. "Server-side components - A pattern language." In: *Proceedings of the 6th European Conference on Pattern Languages of Programms (EuroPLoP '2001)*. Ed. by A. Rüping, J. Eckstein, and C. Schwanninger. Irsee, Germany, July 2001, pp. 87–112. ISBN: 978-3-87940-780-4. URL: https://aka.ms/pfjs6h.

[WB98] P. Weill and M. Broadbent. *Leveraging the new Infrastructure: How market leaders capitalize on Information Technology*. Boston, MA, USA: Harvard Business School Press, 1998. ISBN: 0-87584-830-3.

[Wei+02] P. Weill, M. Subramaniand, and M. Broadbent. "Building IT Infrastructure for Strategic Agility." In: *MIT Sloan Management Review* Vol. 44.1 (2002), pp. 57–65.

[WF12] T. Wellhausen and A. Fiesser. "How to write a pattern? A rough guide for first-time pattern authors." In: *Proceedings of PLoP*. Ed. by P. Avgeriou. 2012, p. 9. DOI: 10.1145/2396716.2396721.

[Woh14] C. Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering." In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. Ed. by M. Shepperd. ACM. 2014, p. 38. DOI: 10.1145/2601248.2601268.

[XK04] W. Xia and W. R. King. *Antecedents of organizational IT Infrastructure capabilities*. Research rep. University of Minnesota, Mar. 12, 2004. URL: https://www.misrc.umn.edu/workshops/2004/fall/weidong.pdf (visited on 02/25/2017).

[Yan10] G. Yann-Gaël. "Design Patterns: Empirical studies on the impact of design patterns on quality." In: *Encyclopedia of Software Engineering*. Taylor and Francis, Sept. 2010. Chap. 19, pp. 204–219. DOI: 10.1081/E-ESE-120044217.

[Zha11] M. Zhang. *iModel - Infrastructure Modeling in Sparx EA*. Accessed on: August 2015. Jan. 2011. URL: http://www.catchlimited.com/uploads/sparxsystems/iModelpresentation-eaug.pdf (visited on 08/22/2013).

Appendix A

Systematic Literature Review Materials

This appendix presents the *(i)* content types and databases available in b-on Knowledge Library, *(ii)* the data collection form used in SLR and *(iii)* list of papers selected related with ITI and this research work.

A.1 b-on Knowledge Library

The *b-on Knowledge Library* is a service that provides access to different types of scientific content as summarized in table A.1. The *b-on* service provides access to thousands of journals and e-books from some of the leading providers of international scientific content. The list of *b-on* databases with content related with this research work is available in table A.2.

Table A.1: Content types provided by b-on knowledge library

Content Type	Description
Periodic	Scientific magazines with articles reviewed by experts.
Base References	Only abstracts, table of contents and citations are available.
Integral Base References	Access to magazines abstract, table of contents and the whole text.
Library Catalogs	Library databases that provide document references and location in one or multiple libraries.
Reference Work	General or specific folders, dictionaries, glossaries and guides.
Portal	Internet pages with access to online services such as news, folders with information and links to other pages.
Indexes	Alphabetic lists with people names, titles, subjects.

Table A.2: Databases available in the b-on portal

Database Name	URL	Content Type
Web of Science (ISI)	`http://webofknowledge.com/`	Base References
Journal Citation Reports (ISI)	`http://admin-router.webofknowledge.com/?DestApp=JCR`	Base References
Theses & Dissertations Catalog	`http://www.ndltd.org`	Indexes
Academic Search Complete (EBSCO)	`http://search.ebscohost.com`	Integral Base References
British Library Public Catalog	`http://catalogue.bl.uk`	Library Catalog
Library of Congress	`http://catalog.loc.gov/`	Library Catalog
DOAJ	`http://www.doaj.org/`	Periodic
Highwire Press	`http://highwire.stanford.edu/`	Periodic
Oaister	`http://www.oclc.org/oaister`	Periodic
SCIRUS	`http://www.scirus.com`	Portal
ACM - Digital Library	`http://portal.acm.org/dl.cfm`	Publisher
Annual Reviews	`http://www.annualreviews.org/`	Publisher
Elsevier - Science Direct	`http://www.sciencedirect.com`	Publisher
IEEE Xplore	`http://ieeexplore.ieee.org/Xplore/dynhome.jsp`	Publisher
IOP Journals	`http://iopscience.iop.org/`	Publisher
SpringerLink(Springer/Kluwer)	`http://www.springerlink.com`	Publisher
Taylor & Francis	`http://www.tandfonline.com/page/looking-for-something`	Publisher
Wiley Online Library (Wiley)	`http://onlinelibrary.wiley.com/advanced/search`	Publisher

A.2 SLR Data Collection Form

Table A.3: Systematic review data collection form

#	Item	Description
Publication description		
1	Publication identifier	unique id for the publication, year
2	Publication author	
3	Publication title	Title of the publication
4	Type of publication	Journal article, conference paper, workshop paper, book section, dissertation, technical report
5	Publication aims	What were the aims of the publication?
6	Design of publication:	Qualitative, quantitative (experiment, survey, case study, action research)
7	Research hypothesis:	Statement of hypotheses, if any
8	Definition of ITI in the publication	Verbatim from the publication
9	Setting of publication	Industry, in-house/supplier, products and processes used
10	Control group	Yes, No (#groups, sample size)
11	Data collection	How was the data obtained? (questionnaires, interviews, etc.)
12	Data analysis	How was the data analyzed? (qualitative, quantitative)
Findings		
1	Findings and conclusions	What were the findings and conclusions?
2	Validity	Limitations, threats to validity, identification
3	Relevance	Research, practice

A.3 Eligible Publications

After applying the exclusion criteria, the list on table A.4 contain the remaining 72 publications to be full analyzed. These studies were obtained from the following sources:

- **26** from Research Databases
- **46** from Google Scholar.

Table A.4: Eligible publications

#	Publication	Source	Year
1	Impact of IT Infrastructure on Innovation Performance: An Empirical Study on Private Universities In Iraq	ScienceDirect	2016
2	ACM Weaving in Patterns into It Infrastructure Models: Industry Case and Exemplary Approaches	ACM	2015
3	A preliminar Study on Learning Orientation, IT Infrastructure Flexibility and Agility	Scholar	2015
4	Understanding the ways organizational learning drives IT infrastructure	IEEE	2014
5	Design Criteria and Design Concepts for an Integrated Management Platform of IT Infrastructure Metrics	Scholar	2014
6	IT Infrastructure Flexibility as Determinant of Strategic Utilization of Information Systems - A Conceptual Framework	Scholar	2014
7	Living IT infrastructures - An ontology-based approach to aligning IT infrastructure capacity and business needs	Scholar	2014
8	Next-generation IT infrastructure	Scholar	2014
9	A model to support IT infrastructure planning and the allocation of IT governance authority	ScienceDirect	2014
10	An IT Infrastructure Capability Model	ACM	2013
11	Exploring organizational learning orientation as antecedent of IT infrastructure capability to achieve organizational agility	IEEE	2013
12	Designing network infrastructure for an e-learning cloud	Scholar	2013
13	IT Infrastructure and Customer Service Delivery	Scholar	2013
14	(MC2)2: criteria, requirements and a software prototype for Cloud infrastructure decisions	Wiley	2013
15	An ontology-based system for Cloud infrastructure services' discovery	IEEE	2012
16	An Enterprise Cloud Model for Optimizing IT Infrastructure	Scholar	2012
17	Capturing value from IT infrastructure innovation	Scholar	2012

Continued on next page

Table A.4 – continued from previous page

#	Publication	Source	Year
18	Feature modeling: An extended perspective to design IT infrastructure	Scholar	2012
19	Leveraging IT infrastructure to facilitate a firm's customer agility and competitive activity: An empirical investigation	Scholar	2012
20	The role of IT infrastructure capability (ITIC) in management	Scholar	2012
21	Impact of IT Infrastructure on Procurement Process Performance of SME's (An Empirical Study of SME's in Pakistan)	Scholar	2011
22	On Technological IT Infrastructure Flexibility	Scholar	2011
23	A Conceptual Study on IT Infrastructure Flexibility Evaluation Model for IT/IS Projects: Construction Industry Perspective	Scholar	2010
24	Designing a concept for an IT Infrastructure for an integrated research and treatment center	Scholar	2010
25	Innovation Managed and IT Infrastructure Capability	Scholar	2010
26	Modelling cloud computing infrastructure	Scholar	2010
27	The role of IT infrastructure flexibility as enabler of organizational responsiveness and competitive advantage	ScienceDirect	2010
28	IT infrastructure capabilities and IT project success: a development team perspective	SpringerLink	2010
29	An optimal infrastructure design method of cloud computing services from the BDIM perspective	Scholar	2009
30	Resource or capability? A dissection of SMEs' IT infrastructure flexibility and its relationship with IT responsiveness	Scholar	2009
31	The effect of IT infrastructure flexibility on intranet effectiveness	Scholar	2009
32	Exploring the perceived business value of the flexibility enabled by IT infrastructure	ScienceDirect	2009
33	A Process Oriented Assessment of the IT Infrastructure Value: A Proposal of an Ontology Based Approach	SpringerLink	2009
34	Conceptual model of IT infrastructure capability and its empirical justification	IEEE	2008

Continued on next page

Table A.4 – continued from previous page

#	Publication	Source	Year
35	Modeling dependencies of IT Infrastructure elements	IEEE	2008
36	The development of global IT infrastructure	ScienceDirect	2008
37	Examination of a Path Model relating IT Infrastructure with firm performance	Wiley	2008
38	Gaining agility through IT personnel capabilities: The mediating role of IT infrastructure capabilities	Scholar	2007
39	IT infrastructure: structure, properties and processes	Scholar	2006
40	An Empirical Study of the Relationships Between IT Infrastructure Flexibility, Mass Customization, and Business Performance	ACM	2005
41	Strategic alignment revisited: connecting organizational architecture and IT infrastructure	IEEE	2004
42	Antecedents of organizational IT infrastructure capabilities	Scholar	2004
43	Assessing the Organizational Impact of IT Infrastructure Capabilities	Scholar	2004
44	Bridging the gap: Linking IT Infrastructure and business processes	Scholar	2004
45	IT infrastructure capability and firm performance: An empirical analysis	Scholar	2004
46	Managing next-generation IT infrastructure	Scholar	2004
47	The complementarity of IT infrastructure and e-commerce capability: A resource-based assessment of their business value	Scholar	2004
48	Aligning organizations and their IT infrastructure: how to make IT support business	Scholar	2003
49	The impact of IT infrastructure flexibility on strategic alignment and application implementations	Scholar	2003
50	Where have all the flowers gone? A modular systems perspective of IT infrastructure design and productivity	Scholar	2003
51	A practical framework for discussing IT infrastructure	IEEE	2002
52	Building IT infrastructure for strategic agility	Scholar	2002
53	Capturing the technological dimensions of IT infrastructure change: A model and empirical evidence	Scholar	2002

Continued on next page

Table A.4 – continued from previous page

#	Publication	Source	Year
54	Determinants of Organizational IT Infrastructure Capabilities: An Empirical Study	Scholar	2002
55	IT infrastructure for strategic agility	Scholar	2002
56	The shareholder-wealth and trading-volume effects of information-technology infrastructure investments	Scholar	2002
57	What IT infrastructure capabilities are needed to implement e-business models?	Scholar	2002
58	Managing IT infrastructure standardisation in the networked manufacturing firm	ScienceDirect	2002
59	Funding the"infostructure": A guide to financing technology infrastructure in higher education	Scholar	2001
60	An exploratory examination of the relationship between flexible IT infrastructure and competitive advantage	ScienceDirect	2001
61	Hunting for the Treasure at the End of the Rainbow: Standardizing corporate IT Infrastructure	SpringerLink	2001
62	An Exploratory Analysis of the Value of the Skills of IT Personnel: Their Relationship to IS Infrastructure and Competitive Advantage	Wiley	2001
63	IT infrastructure capability's impact on firm financial performance: an exploratory study	Scholar	2000
64	Measuring the flexibility of IT infrastructure: Exploratory analysis of a construct	Scholar	2000
65	The implications of IT infrastructure for business process redesign	Scholar	1999
66	The transparent evolution of IT infrastructure components	Scholar	1999
67	Strategic context and patterns of IT infrastructure capability	ScienceDirect	1999
68	Building a re-use infrastructure: Optronics Reference Architecture and Layered Reference Model	Wiley	1999
69	IT infrastructure: standards and flexibility	Scholar	1998
70	Multi-stage real options: The cases of IT infrastructure and international bank expansion	ScienceDirect	1998
71	Firm Context and Patterns of IT Infrastructure Capability (Best Paper Award)	Scholar	1996

Continued on next page

Table A.4 – continued from previous page

#	Publication	Source	Year
72	An exploration of firm-wide IT Infrastructure investment and services	Scholar	1995

Appendix B

Online Survey

This appendix presents the *(i)* Online survey used to evaluate the state of the art in the design of ITIs and to validate our proposals, *(ii)* the *call for participation* which was personalized messages sent to authors of papers about ITI, indicating the importance of answering the survey.

B.1 Survey

This section, presents the online survey, organized in six sections (Introduction, Background, Modeling IT Infrastructures, IT Infrastructure Patterns, Tools and About). The introduction section consists of an *(i)* invitation mail, *(ii)* cover letter and a *(iii)* consent form. The online survey included the following three videos:

- **Introduction:** Duration: 1:30s Description: This work is focused on Information Technology Infrastructures (ITIs), more specifically in the context of their design or representation. The term ITI is not new but surprisingly there is not a generally accepted definition for it. According to ITIL, infrastructure includes all the hardware, software, networks, facilities, etc., that are required to develop, test, deliver, monitor, control or support IT Services, but not the associated people, processes and documentation. The video defines was in our context we considered IT Infrastructure. URL: `https://youtu.be/pVrATLAS87k`
- **ITI Profile.** Duration: 2:54s Description: This video gives an overview of the ITI profile. The ITI profile was based on the Unified Modeling Language (UML) that provides a generic extension mechanism for customizing UML models for particular domains such as IT infrastructures. Extension mechanisms allow refining standard semantics in strictly additive manner, preventing them from contradicting standard

semantics. Profiles are defined using stereotypes, tag definitions, and constraints which are applied to specific model elements, like Classes, Attributes, Operations, and Activities. The ITI Profile is a collection of such extensions that collectively customize UML for the domain of IT infrastructures. URL: `https://youtu.be/UwU37YG3WWg`

- **ITI Patterns:** Duration: 1:19s Description: The concept of design patterns is not new and was created in the field of architecture by the architect Christopher Alexander. He designed more than 200 buildings around the world, and the idea behind design patterns was to empower anyone to design and build at any scale. Design patterns can be described has proven solutions or best practices that can be used to solve common problems. The concept was applied in multiple different domains and gained popularity in computer science after the book "Elements of Reusable Object-Oriented Software" in 1994 by the so-called "Gang of Four". However, in the domain of IT infrastructures, the concept of design patterns did not become popular and is not widely used. To simplify the design of IT Infrastructures, promote knowledge reuse, contribute to have better documented IT infrastructures among other benefits, we created the concept of IT infrastructure Patterns. To be able to use these patterns during the IT Infrastructure design, we only have to drag and drop the pattern from the toolbox and adapt to our reality. URL: `https://youtu.be/qABGjxujpII`

B.1.1 Section I - Introduction - Invitation Email

Subject: Infrastructure Architects: LinkedIn Request for Expertise

Dear firstname lastname,

Thank you for being a member of the infrastructure architects LinkedIn group!

We are doing research regarding the current state of the art to represent IT infrastructures (ITIs) , and the effectiveness of these designs in terms of knowledge transfer, reuse of best practices, support implementation, notation and formality, among other aspects. We are also studying the impact of using an ITI profile and ITI design patterns.

To better understand the use of the ITI profile and ITI design patterns, we would like to kindly ask you to watch the following video.

`https://youtu.be/pVrATLAS87k`

We would like to invite you to participate in this research by answering a few questions regarding your personal experiences with ITI representations. Your contribution will help us analyze how useful an ITI profile and ITI design patterns can be in practice. The survey is located at the link below and may take about 5-10 minutes to complete.

`https://iscteiul.co1.qualtrics.com/SE/?SID=SV_54saTHd97SsOw0R`

If you have any question about our research or are interested in receiving the survey results, please let us know and we will answer as soon as possible. We look forward to hearing from you.

Please do also copy this request to anyone else who you think might be interested in participating in this survey. We really appreciate your time. This invitation will expire within 5 business days or until we have reached the required number of answers.

Best regards,
Luís Ferreira da Silva

B.1.2 Section I - Introduction - Research Study Cover Letter

PhD Dissertation Survey Research

Exploring the current practices in the IT infrastructure representation and the impact of using an approach based on patterns on the infrastructure design process.

Dear Respondent:

I am a PhD candidate at Universidade Nova de Lisboa and my research is on exploring the current practices in the IT infrastructure representation and the impact of using an approach based on patterns on the infrastructure design process.

This study will examine the impact of ITI patterns on the current state of the art for designing infrastructures. I will appreciate your assistance in taking about 10 minutes to take this on-line survey. The survey that follows is a research tool for my doctoral dissertation and your participation in completing the on-line survey will be highly appreciated.

There are no wrong or right questions in this questionnaire and we are looking for honest answers. For anonymity, no names or identification of respondents will appear in the dissertation or anywhere else. Should you have any questions, please kindly feel free to contact the researcher at luis.alexandre@campus.fct.unl.pt or the research supervisors:
Prof. Doutor Fernando Brito e Abreu at `fba@iscte.pt`
Prof. Doutor Miguel Afonso Goulão at `mgoul@fct.unl.pt`

We truly appreciate the help and I thank you for your time and assistance in this matter.

Sincerely,
Luís Ferreira da Silva

B.1.3 Section I - Introduction - Personalized Message (Example)

Figure B.1: Personalized message sent to ITI authors (example)

Dear Researcher,

We found your paper with the name **"Machine learning approach for exploring rock arts through the cloud infrastructure"** published in 2015 and we are analyzing it in a systematic literature review about IT Infrastructures.

In the scope of my PhD research work at the Universidade Nova de Lisboa we believe to have produced some advancements to mitigate the problem of representing IT infrastructures, and we would like to invite you to provide feedback on those advancements answering the following online survey.

Anonymous Survey Link: https://iscteiul.co1.qualtrics.com/SE/?SID=SV_54saTHd97SsOw0R

The survey contains three short videos that demonstrate the core of our proposals in practice. We will sincerely appreciate if you accept participating in this online survey, since your opinion is of utmost importance to check the practical feasibility and validity of our proposals and fine-tune our research strategy accordingly. **In recognition for your effort (it will take around 20m) we will provide free first-hand access to our tools to all respondents that fully complete the survey.**

There are no wrong or right answers. We are just looking for honest answers that match your perception of reality as close and fairly as possible. For anonymity sake, no names or identification of respondents will appear in the PhD dissertation or anywhere else.

If you have any questions, please feel free to contact me or to contact the Research Supervisors:

Research Supervisors:

- Professor Fernando Brito e Abreu | fba@iscte.pt
- Professor Miguel Afonso Goulão | mgoul@fct.unl.pt

We truly appreciate your cooperation and personally thank you for your time and assistance in this matter.

Yours sincerely,

Luis Ferreira da Silva
Principal Researcher | Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia, Caparica, 2829-516 Portugal
+351 96 16 24 092 Luis.Alexandre@campus.fct.unl.pt LinkedIn

B.1.4 Section I - Introduction - Informed Consent Form

PhD Dissertation Survey Research

Exploring the current practices in the IT infrastructure representation and the impact of using an approach based on patterns on the infrastructure design process.

Purpose. I invite you to take part in my PhD survey study. The study investigates the current practices in the design of ITIs, and assesses the impact of using the ITI profile and ITI patterns created in the process of designing ITIs. The expected outcome of this study will provide guidance and tools on different approaches to design IT infrastructures.

Participation requirements. These are twenty six (26) questionnaire questions involved and it will take you approximately 20 minutes to complete.

People Involved in this Study. Please feel free to contact at any time, the individuals involved in this study, whose contacts are available in the cover letter.

Potential Risk/Discomfort. This survey has no risks associated with it. Moreover, you may withdraw at any time. You may also choose not to answer any question that you do not feel comfortable to answer or for which you are not sure of the answer.

Benefits. The benefits of participating in this survey includes the access to the results after the scientific analysis and we will give you access to the Enterprise Architect Add-ins modeling IT infrastructures.

Anonymity/Confidentiality. All data collected in this study are kept confidential. They are not accessible to anyone other than the research team.

Right to Withdraw. Participation is voluntary, and you have the right to withdraw from the study at any time without penalty, and you may also omit questions on the survey if you do not want to answer them.

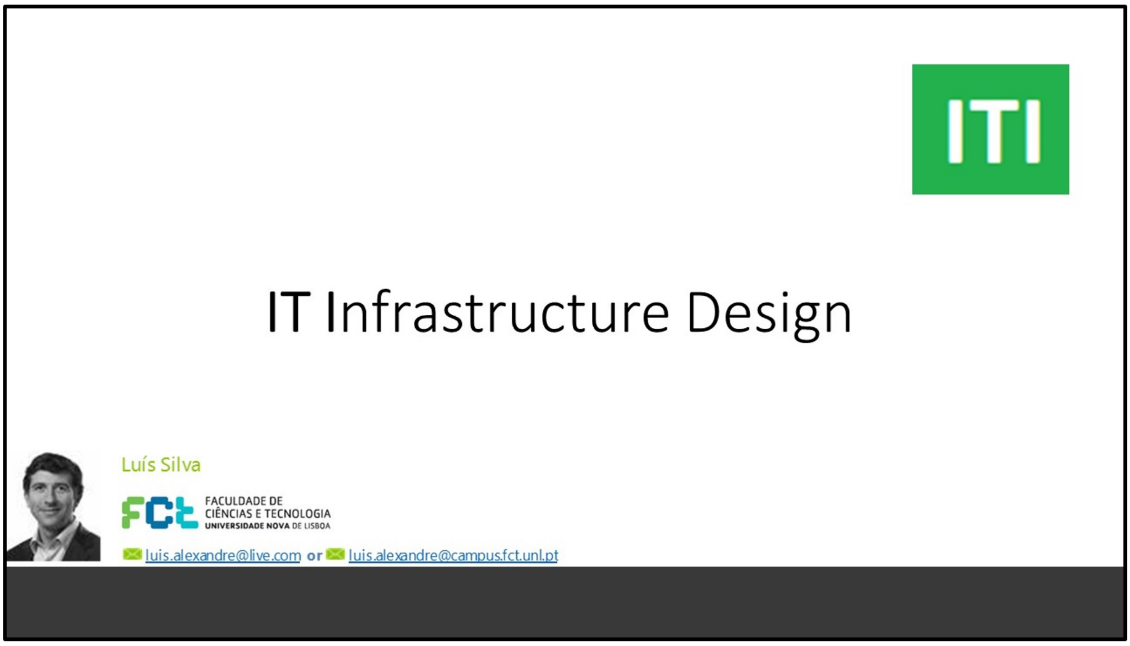
Questions or Complaints. Should you have any questions or complaints about this study, you may contact the researchers, whose name and contact information are provided above.

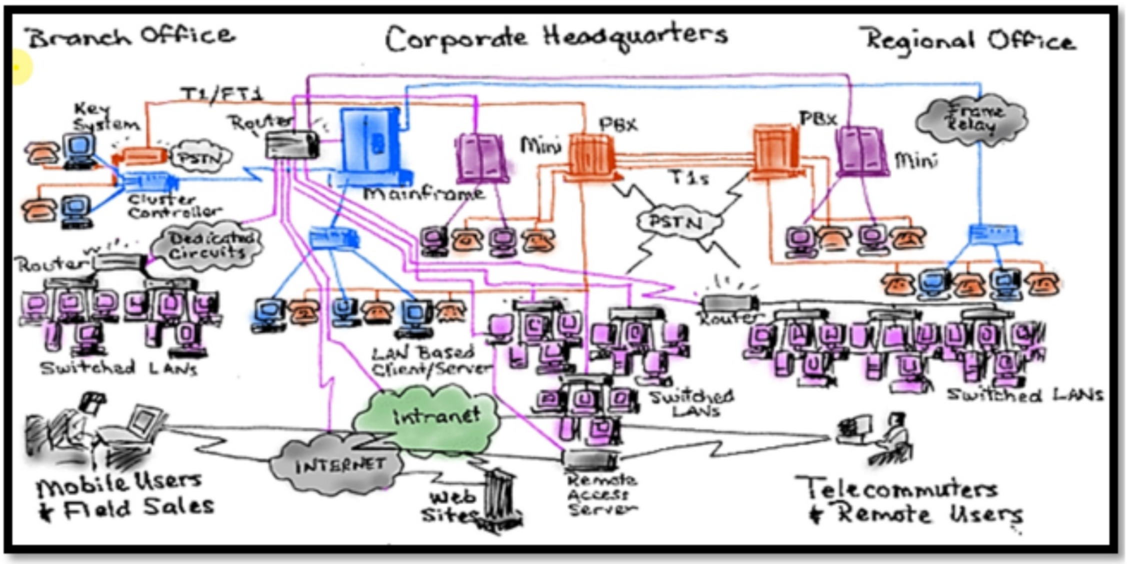
Thank you for participating in this study! We would be happy to answer any questions you might have.

Sincerely,
Luís Ferreira da Silva

B.1.5 Section II - Background

This work is focused on Information Technology Infrastructures (ITIs), more specifically in the context of their design or representation. The term ITI is not new but surprisingly there is not a generally accepted definition for it. According to ITIL, infrastructure includes all of the hardware, software, networks, facilities, etc., that are required to develop, test, deliver, monitor, control or support IT Services, but not the associated people, processes and documentation. The video defines was in our context we considered IT Infrastructure. Video URL: https://www.youtube.com/watch?v=pVrATLAS87k





Question 1. How effective are the current IT infrastructure representations as the example provided above to **STORE** or **CONTAIN** all the required IT infrastructure details/-configurations?

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 2. How frequently do you **NEED ACCESS** to IT Infrastructure representations (e.g. sketches, diagrams, pictures)?

- Daily
- Weekly
- Monthly
- Yearly
- Other (Please specify)

Question 3. According to your experience, **WHICH ROLES** are likely to create or adapt IT infrastructure representations (e.g. sketches, diagrams, pictures) ?

- Architects
- Consultants
- Managers
- Developers
- Business Decision Makers
- IT Decision Makers
- Administrators
- Other (Please Specify)

Question 4. How do you classify the task of creating IT infrastructures representations with IT infrastructure components from **DIFFERENT VENDORS**, each with their own specifications?

- Very easy
- Easy
- Neutral
- Hard
- Very hard

Question 5. How do you classify the **OVERALL QUALITY** (in terms of completeness and consistency) of the IT infrastructures representations (e.g. sketches, diagrams, pictures), that you have come across?

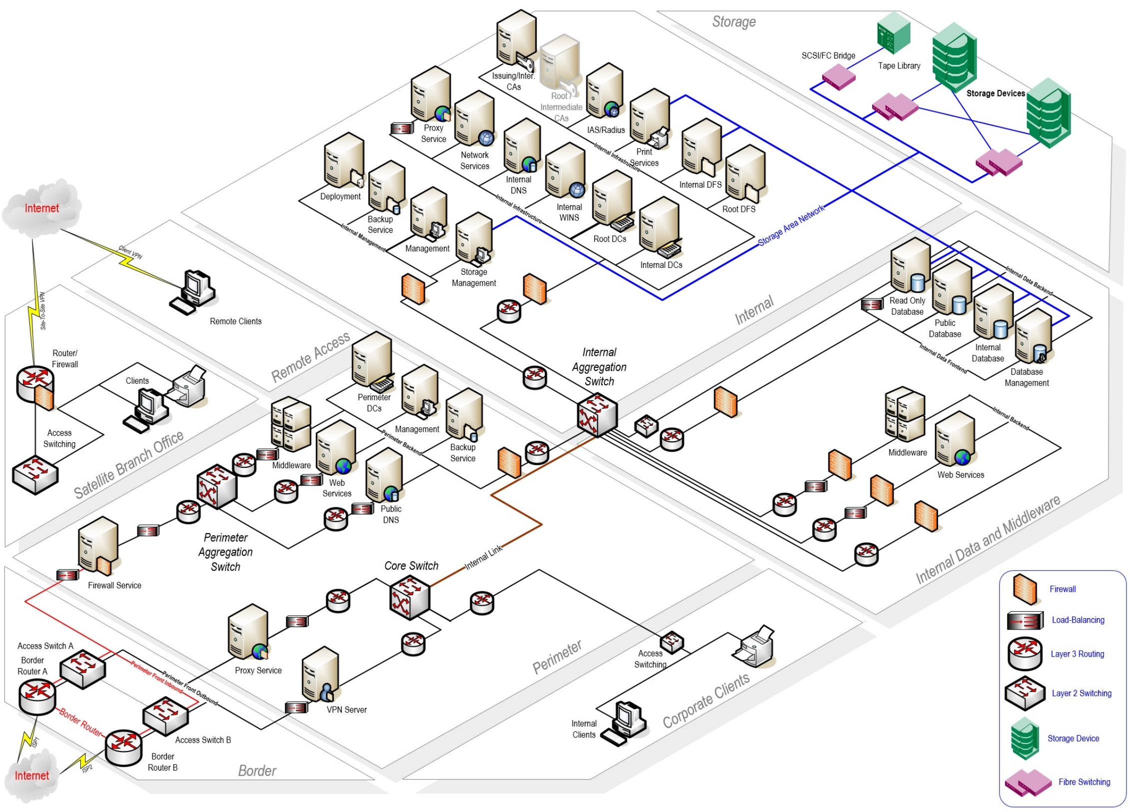
- Poor
- Below average

- Average
- Good
- Excellent

Question 6. What levels of **FORMALITY** are normally applicable to the IT infrastructure representations (e.g. sketches, diagrams, pictures) that you have come across?

- Very informal
- Informal
- Somewhat formal
- Formal
- Very formal

B.1.6 Section III - Modeling IT Infrastructures



Question 7. Using your current approach, how **MUCH EFFORT** do you estimate it would be required to model an IT infrastructure like the one represented in the figure and add to it detailed technical information such as the one represented for one of the modeling elements? Estimate that each infrastructure component has on average 5 attributes that you need to fill.

- Less than 1 day

- 1 to 2 days
- 3 to 5 days
- 6 to 15 days
- More than 15 days

Question 8. Think about the time you spent to represent an IT infrastructure using your current approach. How often that **INVESTMENT WAS WORTH** considering the overall quality in terms of completeness and consistency?

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 9. How often the IT infrastructure representation (e.g. sketch, diagram, picture) is the **ONLY DOCUMENTATION** of the IT infrastructure?

- Never
- Less than 25%
- 26 to 50%
- 51 to 75%
- More than 75%

Question 10. How often does the IT infrastructure representation (e.g. sketch, diagram, picture) contain enough information to support per si the whole IT Infrastructure **IMPLEMENTATION**?

- Never
- Less than 25%
- 26 to 50%
- 51 to 75%
- More than 75%

Question 11. If any, what **NOTATION** do you use more frequently to represent IT infrastructures?

- NONE (No Notation)
- UML Deployment Diagrams
- Archimate
- Architecture Content Framework (TOGAF)
- Other (Please Specify)

B.1.7 Section IV - IT Infrastructure Patterns

Question 12. How do you classify the existing practices in IT infrastructure representations as a way to **CAPTURE KNOWLEDGE**? **Note: CAPTURE KNOWLEDGE** is related

with the capacity of the existing practices to capture individuals technical knowledge such as insights, experiences or lessons learned.

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 13. How do you classify the existing practices in IT infrastructure representations as a way to **STORE KNOWLEDGE** of a specific IT Infrastructure component?

Note: STORE KNOWLEDGE is related with the capacity of the existing practices to represent for instance all the configurations and important attributes of a specific IT Infrastructure component such as a router of switch.

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 14. How do you classify the existing practices in IT infrastructure representations as a way to **SHARE KNOWLEDGE**?

Note: SHARE KNOWLEDGE is an activity through which the IT Infrastructure knowledge (namely, information, skills, or expertise) is exchanged among people with different roles in the organization.

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 15. How do you classify the ability of existing IT infrastructure representations to **REUSE** IT Infrastructure **BEST PRACTICES**?

Note: Examples of best practices in the network design can be for instance the existence of redundant border routers and redundant paths to eliminate a single point of failure.

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 16. How useful a **REPOSITORY** with IT infrastructure best practices can be?

- Not useful
- Somewhat useful
- Neutral
- Very useful
- Extremely useful

Question 17. Which strategies do you use most to deal with **COMPLEX** IT infrastructure designs?

- Ask questions on on-line forums
- Ask questions on distribution lists
- Ask for advisory services
- Search for product blueprints
- Contact manufacturers
- Ask colleagues
- Other (Please specify)

Question 18. How do you share your IT infrastructure **KNOWLEDGE** with others?

- Classroom training
- On-line training
- Training on the job
- Coaching
- Recommend literature
- Other (please specify)

B.1.8 Section V - Tools

Question 19. What **SOFTWARE TOOLS** do you use to represent IT Infrastructures?

Question 20. How do you CLASSIFY these tools in terms of their capacity to represent IT Infrastructures?

- Not useful
- Somewhat useful
- Neutral

- Very useful
- Extremely useful

Question 21. Which is the most recurrent **FORMAT** used to represent IT infrastructures that you have ever access to?

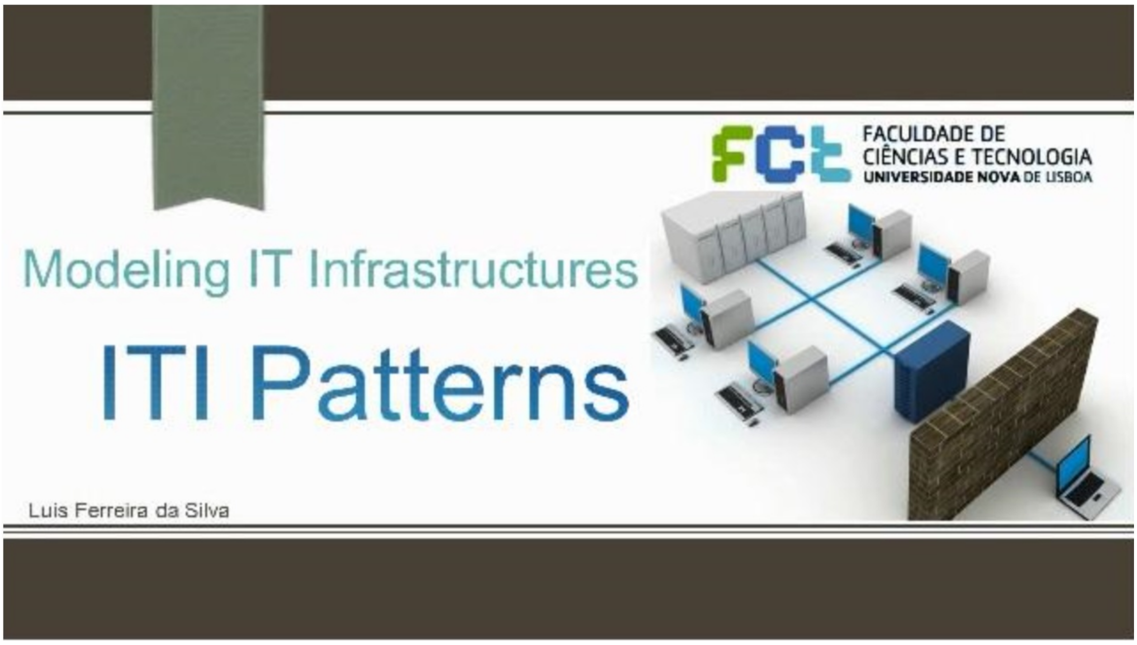
- Diagramming software (e.g. Microsoft Visio)
- Presentation (e.g. Microsoft Powerpoint)
- Imaging software
- CMDB
- Hand-written papers
- Other (please specify)

Video: IT Infrastructure Profile
Duration: 2:54s
URL: https://www.youtube.com/watch?v=UwU37YG3WWg
Description: This video gives an overview of the ITI profile. The ITI profile was based on the Unified Modeling Language (UML) that provides a generic extension mechanism for customizing UML models for particular domains such as IT infrastructures. Extension mechanisms allow refining standard semantics in strictly additive manner, preventing them from contradicting standard semantics. Profiles are defined using stereotypes, tag definitions, and constraints which are applied to specific model elements, like Classes, Attributes, Operations, and Activities. The ITI Profile is a collection of such extensions that collectively customize UML for the domain of IT infrastructures.

Question 22. Please evaluate the ITI Profile based on the IT Infrastructure Profile Video:

- This approach can contribute to have better documented IT Infrastructures.
- This approach can be used by less experienced IT Infrastructure professionals.
- This approach reduces the time to create an IT Infrastructure.
- This approach supports the implementation activities.
- The knowledge sharing with ITI Profile will be simple.

Video: IT Infrastructure Patterns
Duration: 1:19s
URL: https://www.youtube.com/watch?v=qABGjxujpII
The concept of design patterns is not new and was created in the field of architecture by the architect Christopher Alexander. He designed more than 200 buildings around the world, and the idea behind design patterns was to empower anyone to design and build at any scale. Design patterns can be described has proven solutions or best practices that can be used to solve common problems. The concept was applied in multiple different domains and gained popularity in computer science after the book "Elements of Reusable Object-Oriented Software"in 1994 by the so-called "Gang of Four". However, in the domain of IT infrastructures, the concept of design patterns did not became popular and is not widely used.



Question 23. Please evaluate the ITI Profile based on the IT Infrastructure Patterns Video:

- This approach decreases the effort associated with IT infrastructure designs.
- The use a repository with IT infrastructure best practices improves the overall quality of IT infrastructures.

- Based on the ITI Patterns the knowledge sharing among people will be simple.

B.1.9 Section VI - About You and the Questionnaire

Question 24. YEARS working with IT Infrastructures?

Question 25. Highest level of **EDUCATION** or degree completed?

- Doctoral
- Master
- Graduate
- High School
- Other (Please specify)

Question 26. Please provide additional information regarding IT infrastructure representations that you want to share?

Question 27. Additional information regarding the questionnaire that you want to share?

Question 28. Do you want to have access to the IT Infrastructure Modeling Tool and the aggregated results of this survey?

- Yes
- No

Please provide your Email address below.

Note: You will not be identified in any report that is produced using the information you have provided in this questionnaire and your email will not be used for any other purpose, except for sending you feedback on the aggregated answers to this survey and the corresponding interpretation.

B.2 Call for Participation

To increase the participation in the on-line survey we decided to send personalized messages, indicating the importance of answering the survey. The messages were sent only to the authors of papers about IT infrastructures published after 2000. The list of 545 authors e-mail addresses and their papers can be found in table B.1. The personalized message is available in figure B.1.

Table B.1: Articles from authors that answered the survey

Year	Paper Name
2000	Design of a performance technology infrastructure to support the construction of responsive software
2000	Facilitating localized exploitation and enterprise-wide integration in the use of IT infrastructures the role of PC LAN infrastructure standards
2003	Experiences using a dual wireless technology infrastructure to support ad-hoc multiplayer games
2004	A methodological framework for business-oriented modeling of IT infrastructure
2004	Envolving a pervasive IT Infrastructure - A technology Integration approach
2004	Technology infrastructure supporting a medical and bioinformatics masters degree
2005	Evolution of the Intel's e-business data center toward a service-oriented infrastructure
2006	Assessing the Risks of IT Infrastructure - A Personal Network Perspective
2006	IT infrastructure in emerging markets arguing for an end-to-end perspective
2006	Managing the Impact of IT on Firm Success The Link between the Resource-Based View and the IT Infrastructure Library
2006	Reducing barriers for e-business in SMEs through an open service oriented infrastructure
2006	Research and Application on Service Oriented Infrastructure for Networkitized M and S
2006	Single event effects as a reliability issue of IT infrastructure
2006	Technical Infrastructure of a CSIRT
2007	A Communication and IT Infrastructure for Real Time Monitoring and Management of Type 1 Diabetes Patients
2007	Integrated management system for technical infrastructure of telecom sites
2007	Is Transaction Cost Economics Theory Able to Explain Contracts Used for and Success of Firm-Wide IT-Infrastructure Outsourcing
2007	On the integration of Sensor Networks and General purpose IT infrastructure

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2008	A Framework for Investigating the Impact of IT Infrastructure and E-Commerce Capability on Firm Performance
2008	Anticipated IT infrastructure and supply chain integration capabilities for RFID and their associated deployment outcomes
2008	Challenging interoperability and bandwidth issues in national e-Health strategies by a bottom-up approach
2008	Conceptual model of IT infrastructure capability and its empirical justification
2008	Implementing a outsourced technology infrastructure
2008	Realtime-enabled workflow management in service oriented infrastructures
2008	Service Capabilities of IT Infrastructure in Chinese Manufacturing Firms
2008	Service Oriented Infrastructure Framework
2008	Telco goes for IT A Service Oriented IT Infrastructure for Next Generation Networks
2009	A technological infrastructure design for a pediatric oncology network
2009	Impact of recent blackout - return of experiences on utility operation IT infrastructure
2009	IT Challenges for 2009 Fixing the IT Infrastructure
2009	Problem Space and Special Characteristics of Security Testing in Live and Operational Environments of Large Systems Exemplified by a Nationwide IT Infrastructure
2009	Technical infrastructure for a Pan-European federation of testbeds
2009	Technology infrastructure in support of a medical and bioinformatics masters degree
2009	the socio-technological infrastructure of a wikipedia article
2009	Web 2.0, a Boost in IT Infrastructure Flexibility and Team Collaboration
2010	A Multi-Level Technical Infrastructure for Diabetes Chronic Care Management in China
2010	AdWiki Designing and implementing a socio-technical infrastructure for advising freshmen engineering students
2010	An optical network and IT infrastructure virtualisation and provisioning framework
2010	Building a Distributed Block Storage System for Cloud Infrastructure
2010	Complementarities Between Product Design Modularity and IT Infrastructure Flexibility in IT-Enabled Supply Chains
2010	Evaluating the potential of a service oriented infrastructure for the factory of the future
2010	Language technology infrastructures in support to multilingualism

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2010	NoHype Virtualized cloud infrastructure without the virtualization
2010	Representing Eager Evaluation in a Demand Driven Model of Streams on Cloud Infrastructure
2010	Smart grid IT infrastructure selection A T3SD Fuzzy DEA approach
2010	Study on IT Infrastructure Investment Appraisal Based on Real-option Theory
2010	The technical infrastructure of geological survey information grid
2010	Virtual Resources Allocation for Workflow-Based Applications Distribution on a Cloud Infrastructure
2010	VMM-level distributed transparency provisioning using Cloud infrastructure technology
2011	A Cloud Infrastructure for Collaborative Digital Public Services
2011	A Petri Net-T3SD policy driven method for IT infrastructure selection in smart grid
2011	A proposal to provide automated information technology infrastructure with integrated service catalog
2011	Autonomic management of workflows on hybrid grid-cloud infrastructure
2011	Context-Aware Mobile Learning with a Semantic Service-Oriented Infrastructure
2011	Evaluating the impact of planning long-term contracts on the management of a hybrid IT infrastructure
2011	Extensible architecture for high-throughput task processing based on hybrid cloud infrastructure
2011	Information and communication technology infrastructure and management for collaboration with regional universities and colleges
2011	Introducing virtualization management concepts using open source cloud infrastructure managers
2011	myTrustedCloud Trusted Cloud Infrastructure for Security-critical Computation and Data Managment
2011	Notice of Retraction
2011	Security Infrastructure for On-demand Provisioned Cloud Infrastructure Services
2011	Service-Oriented Infrastructure to Support the Deployment of Evolvable Production Systems
2011	Study on digital campus IT infrastructure virtualization
2011	The Design of a Private Cloud Infrastructure Based on XEN
2011	The KOALA cloud management service a modern approach for cloud infrastructure management

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2011	Towards an authorization system for cloud infrastructure providers
2011	Trusted computing architectures for a mobile IT infrastructure
2012	A Cloud Infrastructure for Optimization of a Massive Parallel Sequencing Workflow
2012	A distributed service oriented infrastructure for business process management in Virtual Organizations
2012	A holistic service provisioning solution for federated cloud infrastructures
2012	A Hybrid Cloud Infrastructure for the Optimization of VANET Simulations
2012	A technical infrastructure to support personalized medicine
2012	An Examination of the Relationships among IT Capability Intentions, IT Infrastructure Integration and Quality of Care A Study in U.S. Hospitals
2012	An ontology-based system for Cloud infrastructure services' discovery
2012	An SVM-T3SD policy driven method for IT infrastructure selection in Smart Grid
2012	Analysis of Gromacs MPI Using the Opportunistic Cloud Infrastructure UnaCloud
2012	Application of Artificial Neural Networks in Capacity Planning of Cloud Based IT Infrastructure
2012	Applications of Provenance Data for Cloud Infrastructure
2012	Automated Tagging for the Retrieval of Software Resources in Grid and Cloud Infrastructures
2012	CDM Server A Data Management Framework for Data Intensive Application in Internal Private Cloud Infrastructure
2012	Cloud infrastructure for providing tools as a service quality attributes and potential solutions
2012	Comparison of private cloud infrastructure implementation models
2012	Enabling Scalable Cloud Infrastructure Using Autonomous VM Migration
2012	Implementing Hadoop Platform on Eucalyptus Cloud Infrastructure
2012	Improving higher education with the cloud infrastructure Information system requirements analysis for a modern global university
2012	Incorporating hardware trust mechanisms in Apache Hadoop
2012	iPlant atmosphere a gateway to cloud infrastructure for the plant sciences
2012	Leveraging Cloud Infrastructure for Troubleshooting Edge Computing Systems
2012	Monitoring and detecting abnormal behavior in mobile cloud infrastructure
2012	Optimal Design for Cloud Infrastructure in Multi-layer Service Computing Environment

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2012	Present the Challenges in Eucalyptus Cloud Infrastructure for Implementing Virtual Machine Migration Technique and Provide a Solution for Solve the Challenges
2012	Securing cloud infrastructure against co-resident DoS attacks using game theoretic defense
2012	Security challenges in IT infrastructure for cybersecurity curriculum support
2012	Service-oriented infrastructure at device level to implement agile factories
2012	Smart applications on cloud infrastructure
2012	Streamlining Service Levels for IT Infrastructure Support
2012	Technical infrastructure for monitoring the transportation of oversized and dangerous goods
2012	Towards an Efficient and Secure Educational Platform on cloud infrastructure
2012	Unearthing the Value of Sensor Information Systems for Managing Organizations IT Infrastructure
2012	Unsupervised Neural Predictor to Auto-administrate the Cloud Infrastructure
2012	Validating Cloud Infrastructure Changes by Cloud Audits
2012	Virtualization of IT infrastructure for small and medium businesses
2012	VM provisioning policies to improve the profit of cloud infrastructure service providers
2013	A Framework to Monitor Cloud Infrastructure in Service Oriented Approach
2013	A literature review Exploring organizational learning orientation as antecedent of Information Technology (IT) infrastructure capability to achieve organizational agility
2013	A probabilistic partial order theory approach to IT infrastructure selection for Smart Grid
2013	A Scalable SIEM Correlation Engine and Its Application to the Olympic Games IT Infrastructure
2013	A secured cloud storage technique to improve security in cloud infrastructure
2013	A Virtualization-Based Cloud Infrastructure for IMS Core Network
2013	aCCountS A Service-Oriented Architecture for Flexible Pricing in Cloud Infrastructure
2013	An anti-DoS attack architecture for wireless IT Infrastructure
2013	An integrated framework for optimizing automatic monitoring systems in large IT infrastructures
2013	An IT Infrastructure Capability Model - a11-straube
2013	Automatic Fault Diagnosis in Cloud Infrastructure
2013	Cloud Infrastructure for car service

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2013	Cloud robotics Formation control of a multi robot system utilizing cloud infrastructure
2013	Coexistence of cloud technology and IT infrastructure in higher education
2013	ComSoc's information and communication technology infrastructureComSoc's information and communication technology infrastructure
2013	End-to-end privacy policy enforcement in cloud infrastructure
2013	Hardware-in-the-loop simulation for automated benchmarking of cloud infrastructures
2013	Improving cloud infrastructure utilization through overbooking
2013	Integration of Wireless Sensor and Actuator Nodes With IT Infrastructure Using Service-Oriented Architecture
2013	Monitoring and diagnosing cloud infrastructure
2013	Pattern detection in unstructured data An experience for a virtualized IT infrastructure
2013	Private cloud computing consolidation, virtualization, and service-oriented infrastructure
2013	Resource Allocation Scheme in Cloud Infrastructure
2013	Resource Pool Oriented Trust Management for Cloud Infrastructure
2013	Security mechanisms utilized in a secured cloud infrastructure
2013	Server virtualization in information and communication technology infrastructure in Turkey
2013	Stochastic Model Generation for Cloud Infrastructure Planning
2013	The Impact of IT Infrastructure Flexibility on Strategic Utilization on Information Systems A Conceptual Framework
2013	Towards an OSGi Based Pervasive Cloud Infrastructure
2013	Understanding threats in a cloud infrastructure with no hypervisor
2014	3-D cloud monitoring Enabling effective cloud infrastructure and application management
2014	A Bio-Inspired Prediction Method for Water Quality in a Cyber-Infrastructure Architecture
2014	A cloud infrastructure for scalable and elastic multimedia conferencing applications
2014	A Comparitive Study of Predictive Models for Cloud Infrastructure Management
2014	Acceptance of ICT-intensive socio-technical infrastructure systems Smart metering case in the Netherlands

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2014	Adaptive energy efficient distributed VoIP load balancing in federated cloud infrastructure
2014	Cloud Infrastructure for Higher Education The Sullivan Experience
2014	Computational Offloading or Data BindinD Bridging the Cloud Infrastructure to the Proximity of the Mobile User
2014	Connectivity Re-establishment in the Presence of Dumb Nodes in Sensor-Cloud Infrastructure A Game Theoretic Approach
2014	Construction of Agent-Based Trust in Cloud Infrastructure
2014	Cost minimization scheduling for deadline constrained applications on vehicular cloud infrastructure
2014	Design of Europeana cloud technical infrastructure
2014	Design of wide-area damping systems based on the capabilities of the supporting information communication technology infrastructure
2014	Failure-aware resource provisioning mechanism in cloud infrastructure
2014	Fuzzy Assisted Event Driven Data Collection from Sensor Nodes in Sensor-Cloud Infrastructure
2014	Hybrid cloud infrastructure to handle large scale data for bangladesh people search (BDPS)
2014	In-space transportation infrastructure architecture decisions using a weighted graph approach
2014	IT infrastructure flexibility as determinant of strategic utilization of information systems A conceptual framework
2014	Mobile Traffic Analysis Exploiting a Cloud Infrastructure and Hardware Accelerators
2014	Model-based automation for hardware provisioning in IT infrastructure
2014	Prototyping an Autonomic Cloud Infrastructure to Manage Live Streaming Applications Using a Software Defined Network Performance Analysis and Challenges
2014	Proxy Based Model to Protect Cloud Infrastructure as Service (IaaS) Platforms from DDOS Attacks
2014	Secure Information and Resource Sharing in Cloud Infrastructure as a Service
2014	Securing cloud infrastructure through PKI
2014	Security problems in cloud infrastructure
2014	Special section on data-intensive cloud infrastructure
2014	Synthesis of Adaptation Plans for Cloud Infrastructure with Hybrid Cost Models

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2014	The concept of green Cloud infrastructure based on distributed computing and hardware accelerator within FPGA as a Service
2014	Understanding the ways organizational learning drives information technology (IT) infrastructure
2014	Unified energy aware mapping of multimedia applications on media cloud infrastructure
2014	VM Placement Algorithms for Hierarchical Cloud Infrastructure
2015	A cloud infrastructure for target detection and tracking using audio and video fusion
2015	A decision support platform for IT infrastructure management The university of Trás-os-Montes e Alto Douro services of information and communications case study
2015	A Distributed Cloud Infrastructure for Underresourced higher education communities
2015	A Fairness-Aware Pricing Methodology for Revenue Enhancement in Service Cloud Infrastructure
2015	A Modeling Approach for Cloud Infrastructure Planning Considering Dependability and Cost Requirements
2015	A programmable and virtualized network AND IT infrastructure for the internet of things How can NFV and SDN help for facing the upcoming challenges
2015	A proxy-based cloud infrastructure for home service robots
2015	aDock A Cloud Infrastructure Experimentation Environment Based on Open Stack and Docker
2015	BanglaCloud A cloud infrastructure with abandoned computers
2015	Big data analytics for climate change and biodiversity in the EUBrazilCC federated cloud INFRASTRUCTURE
2015	Enhancing cloud connectivity among NRENs in the SADC region through a novel institution cloud infrastructure framework
2015	Hybrid and Extensible Architecture for Cloud Infrastructure Deployment
2015	Implementation of Cloud Infrastructure Monitor Platform with Power Saving Method
2015	IT Infrastructure-Monitoring Tools
2015	Machine learning approach for exploring rock arts through the cloud infrastructure
2015	Naïve Bayes Classifier Based Traffic Prediction System on Cloud Infrastructure
2015	O-MAP - A per-component on-line anomaly predicting method for Cloud infrastructure

Continued on next page

Table B.1 – continued from previous page

Year	Paper Name
2015	On the Feasibility of an Open-Implementation Cloud Infrastructure - A Game Theoretic Analysis
2015	Power aware resource virtual machine allocation policy for cloud infrastructure
2015	Proposed cloud infrastructure of wearable and ubiquitous medical services
2015	Qualitative evaluation method of IT-infrastructure functioning based on structural optimization of neural network
2015	Reactive Resource Provisioning Heuristics for Dynamic Dataflows on Cloud Infrastructure
2015	Sensing Service Framework for climate alert system using WSN-Cloud infrastructure
2015	Simulation of cloud infrastructure using CloudSim simulator A practical approach for researchers
2015	SSM Secure-Split-Merge data distribution in cloud infrastructure
2015	The migration of the university IT infrastructure toward a secure IaaS Cloud
2015	Towards a Sensor-Cloud Infrastructure with Sensor Virtualization
2015	Tutorial Building Secure and Scalable Private Cloud Infrastructure with Open Stack
2015	User identity and Access Management trends in IT infrastructure- an overview
2015	Virtual Resource Orchestration Constraints in Cloud Infrastructure as a Service
2015	WaSCO A Hybrid Enterprise Desktop Wake-Up System Based on Cloud Infrastructure
2016	Workflow Monitoring of Cloud Infrastructures via Interleaved Logs

Appendix C

ITI Metamodel

This appendix presents the *(i)* created ITI metamodel to support this research work, namely the ITI profile and ITI Patterns, *(ii)* the toolboxes that support the ITI design activities, and *(iii)* an alternative notation that can be used in the ITI profile.

C.1 ITI Metamodel Packages

This section shows how each package is organized in the ITI metamodel, their attributes and relationships. The following packages will be presented:

- ITI Nodes
- ITI Software
- ITI Storage
- ITI Storage Connectors
- ITI Network
- ITI Network Protocols
- ITI Network Zones
- ITI Location
- ITI Network Segment
- Network Cloud
- General Connectors
- ITI Languages
- ITI Team

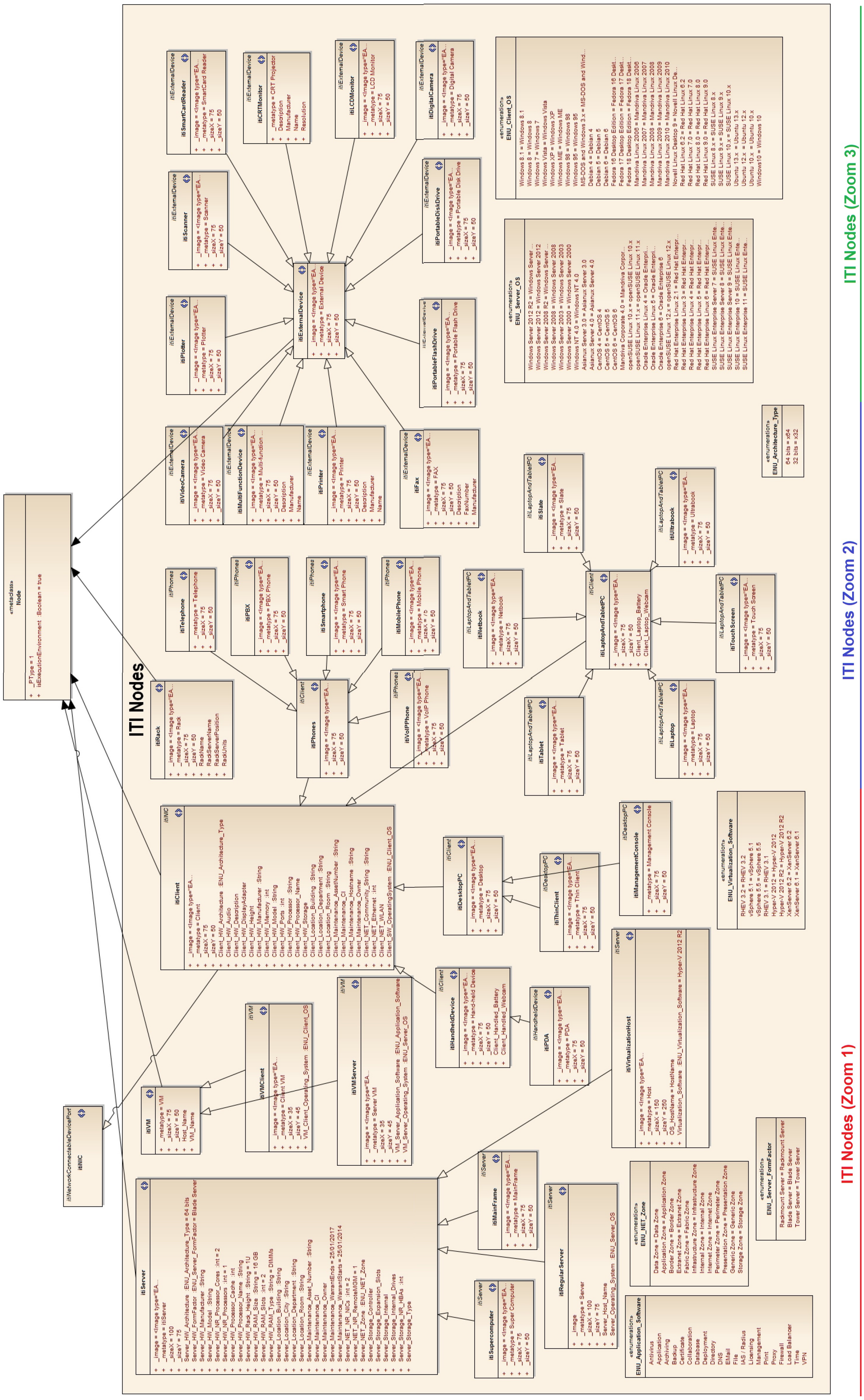
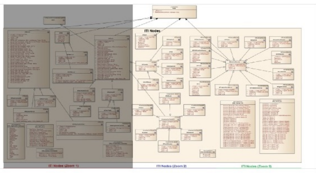


Figure C.1: ITI nodes metaclasses



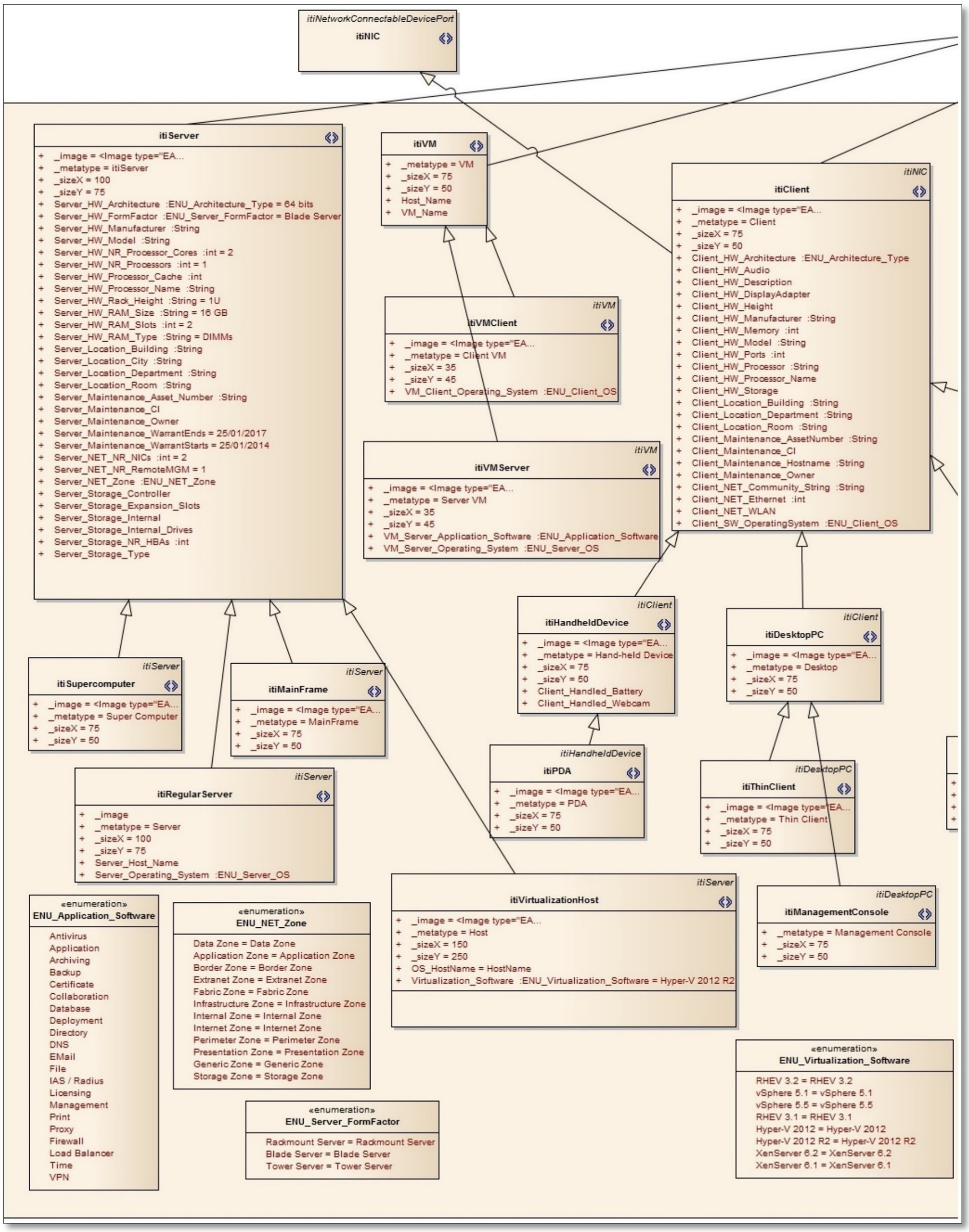


Figure C.2: Subset of ITI nodes metaclasses (zoom 1)

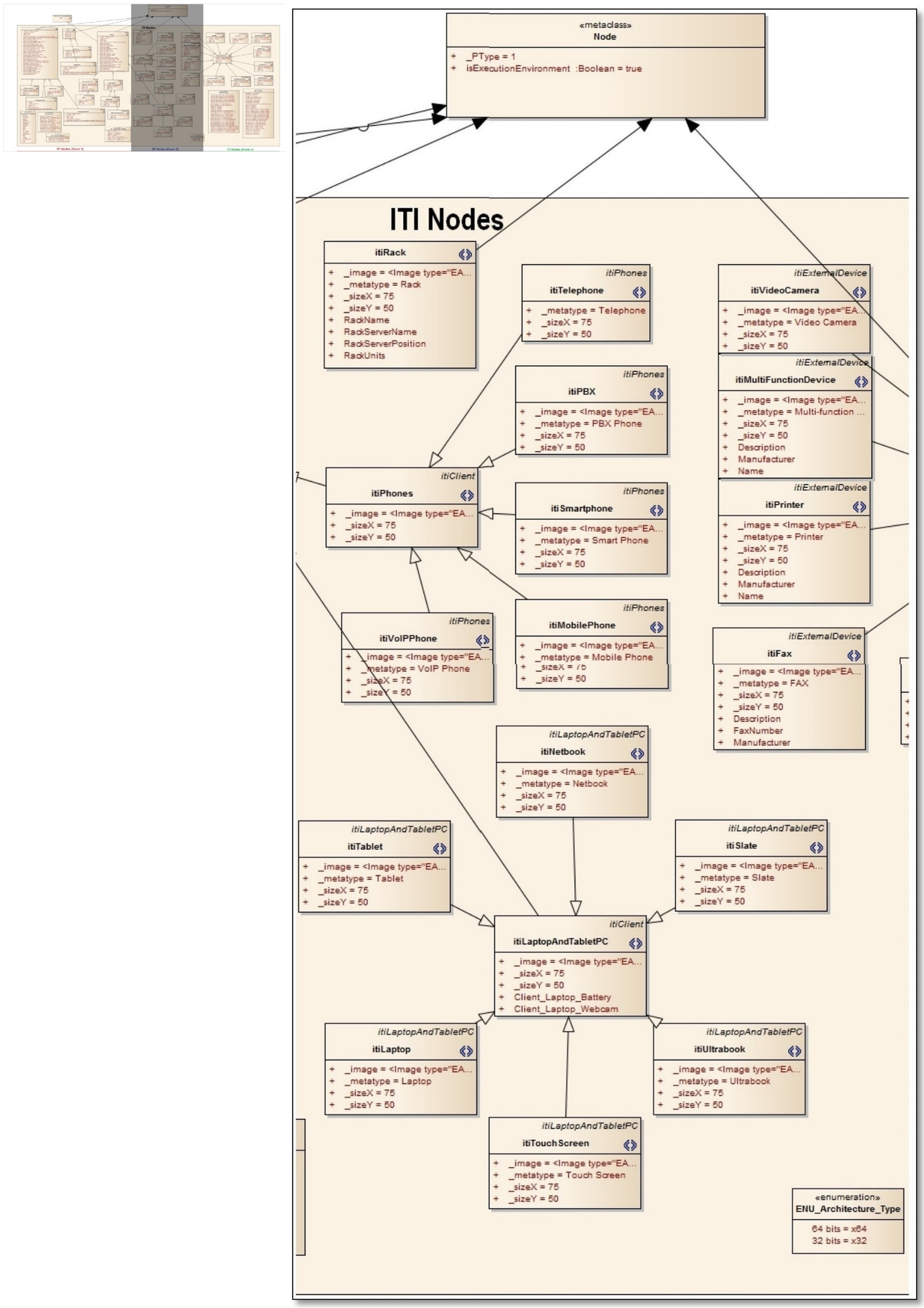
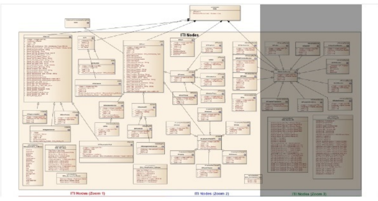


Figure C.3: Subset of ITI nodes metaclasses (zoom 2)



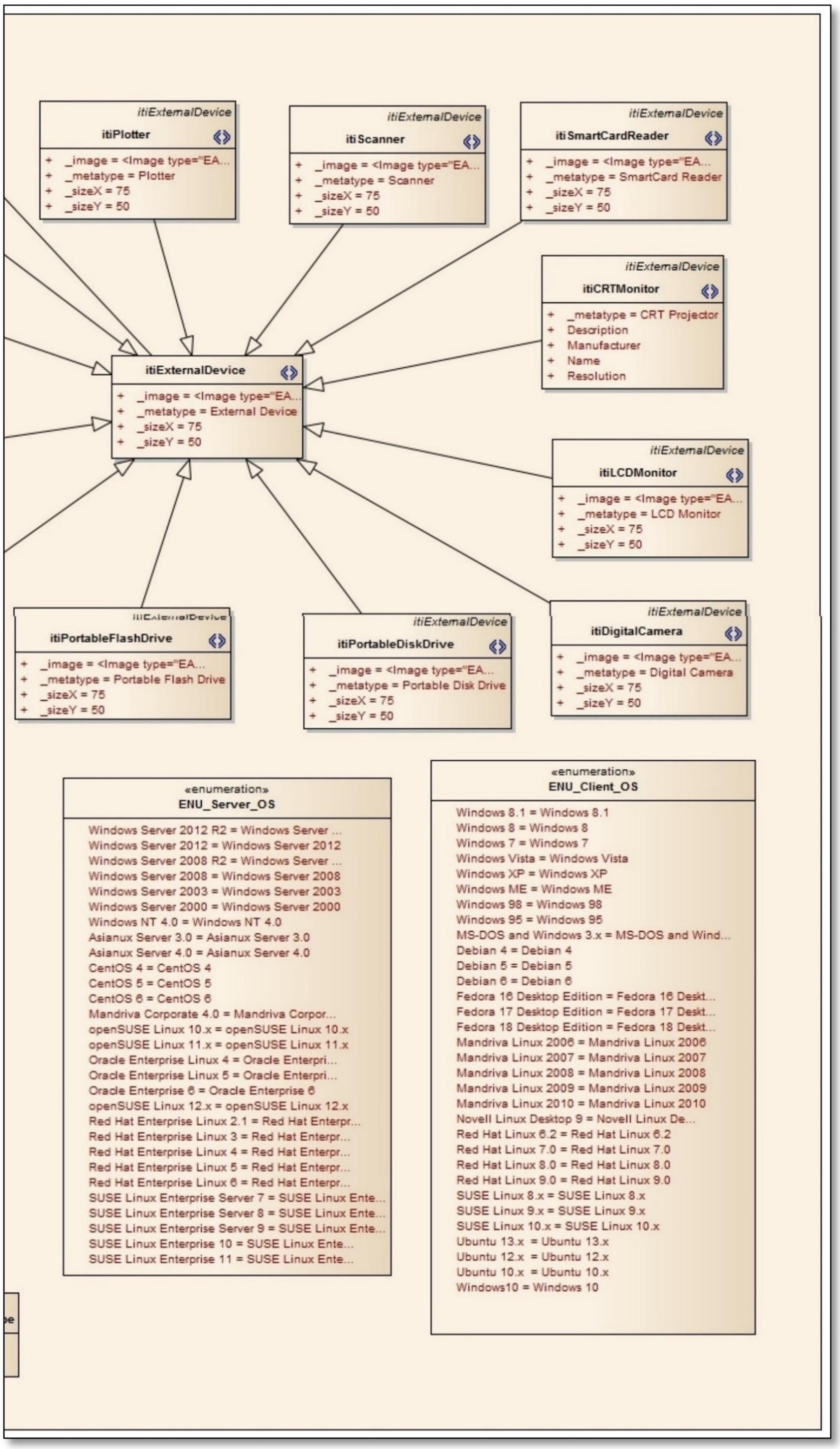


Figure C.4: Subset of ITI nodes metaclasses (zoom 3)

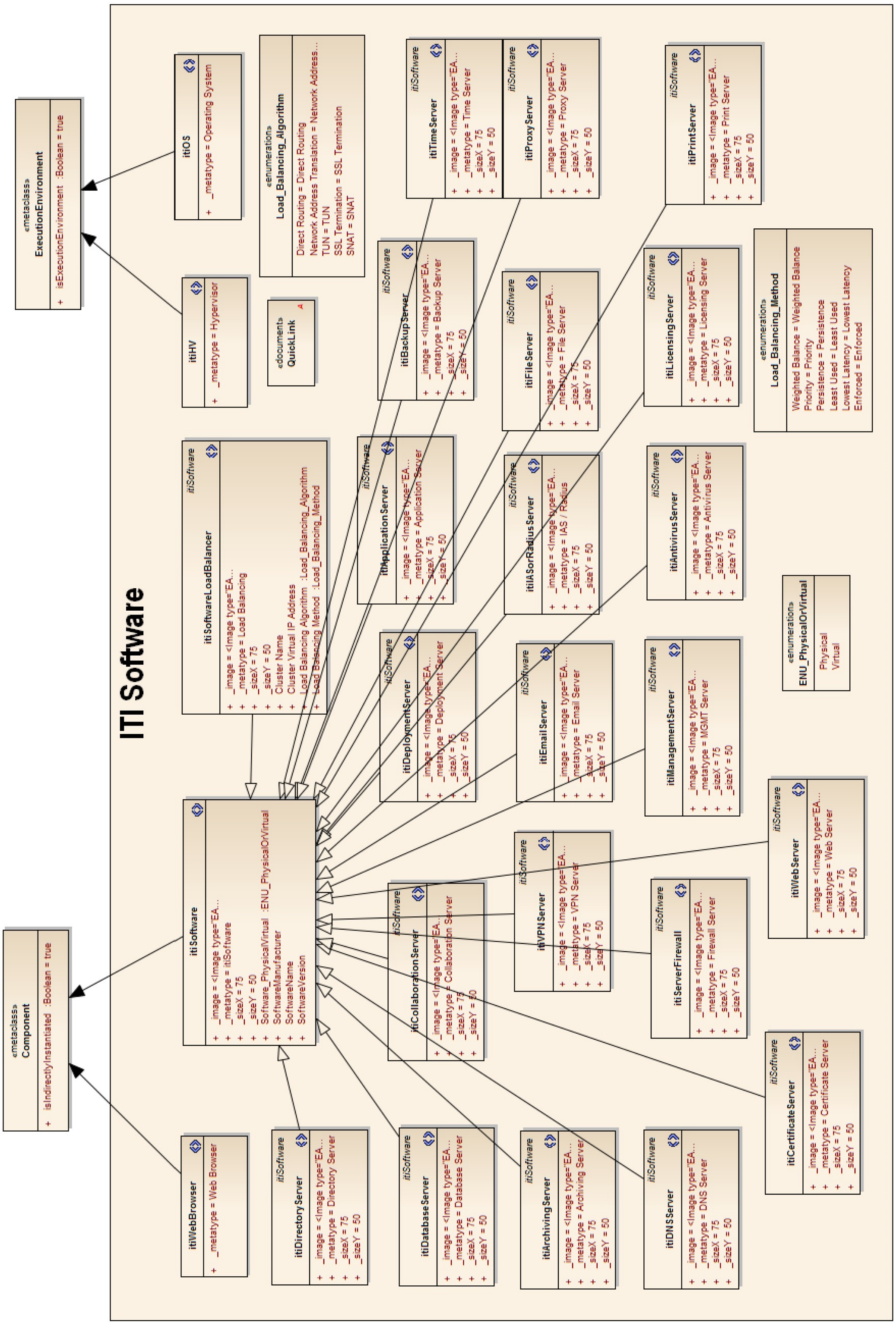


Figure C.5: ITI software metaclasses

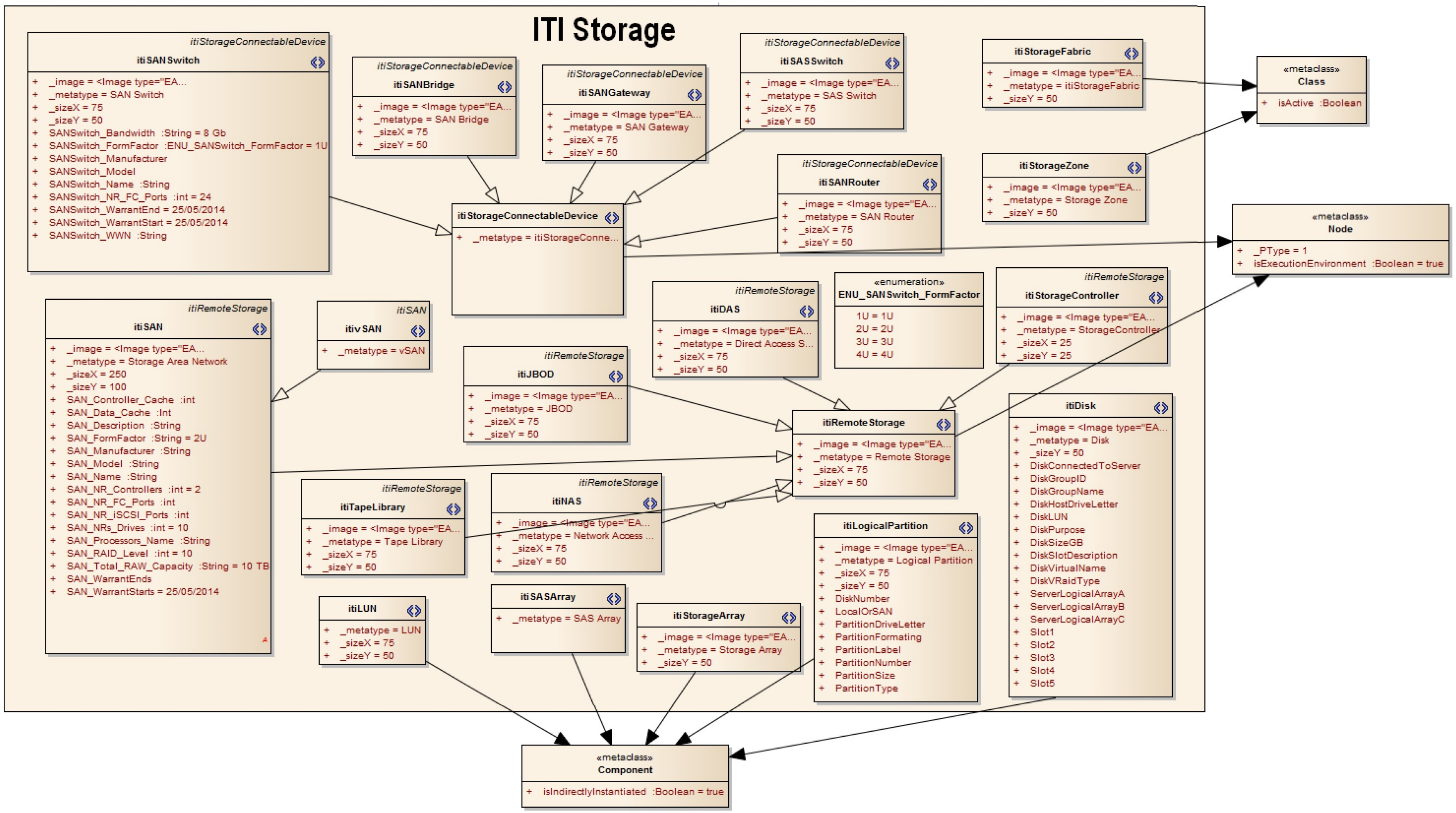


Figure C.6: ITI storage metaclasses

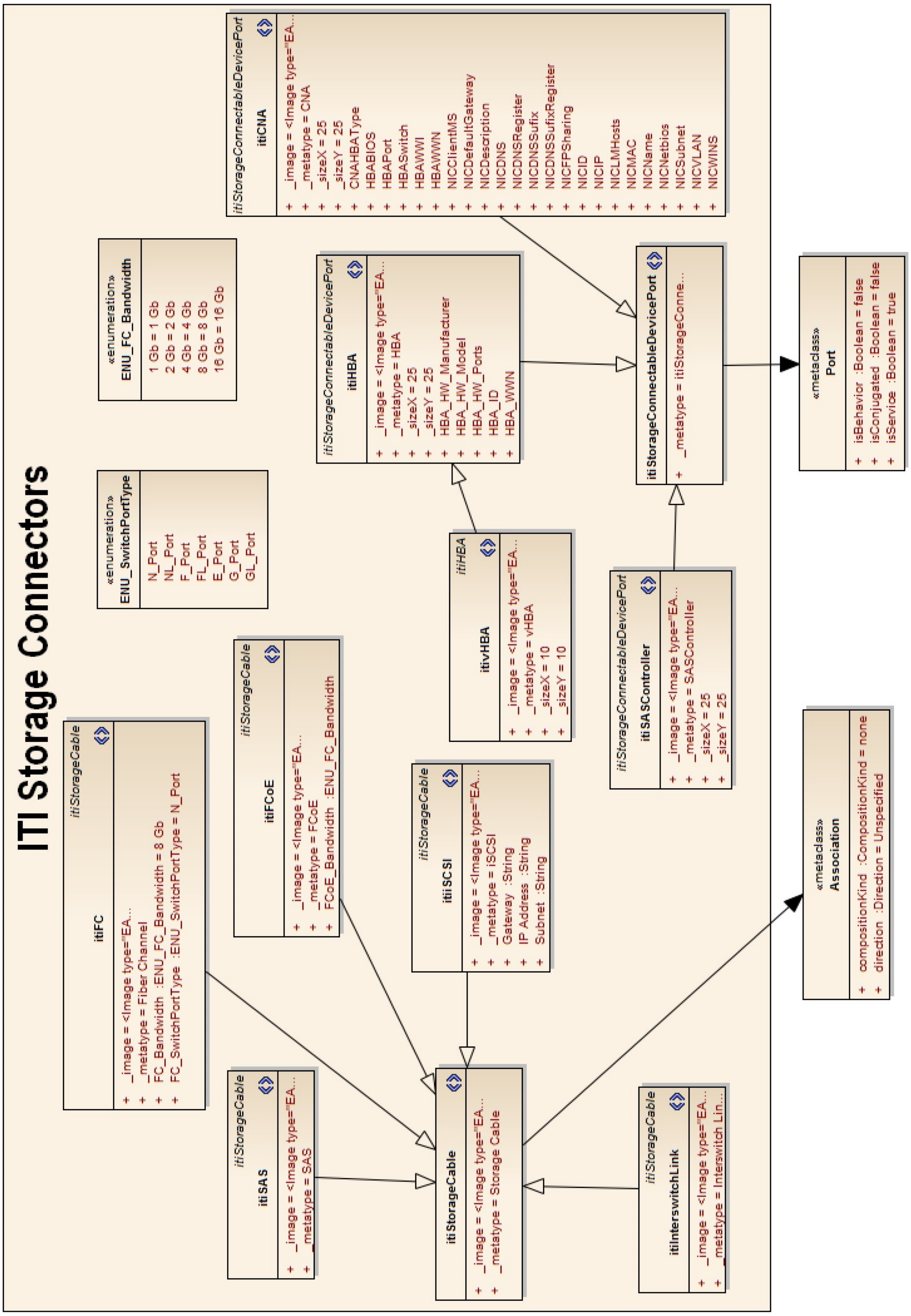


Figure C.7: ITI storage connectors metaclasses

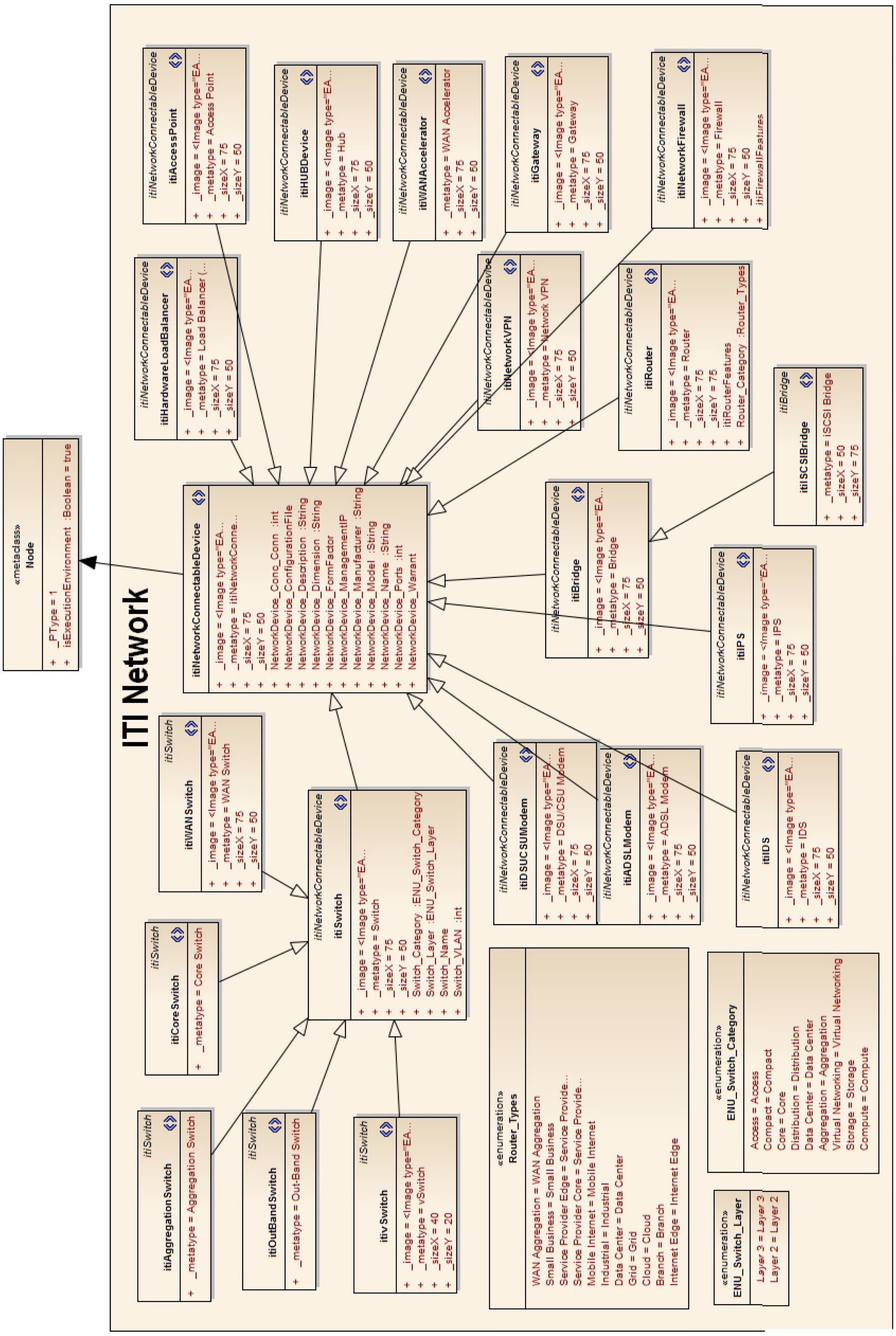


Figure C.8: ITI network metaclasses

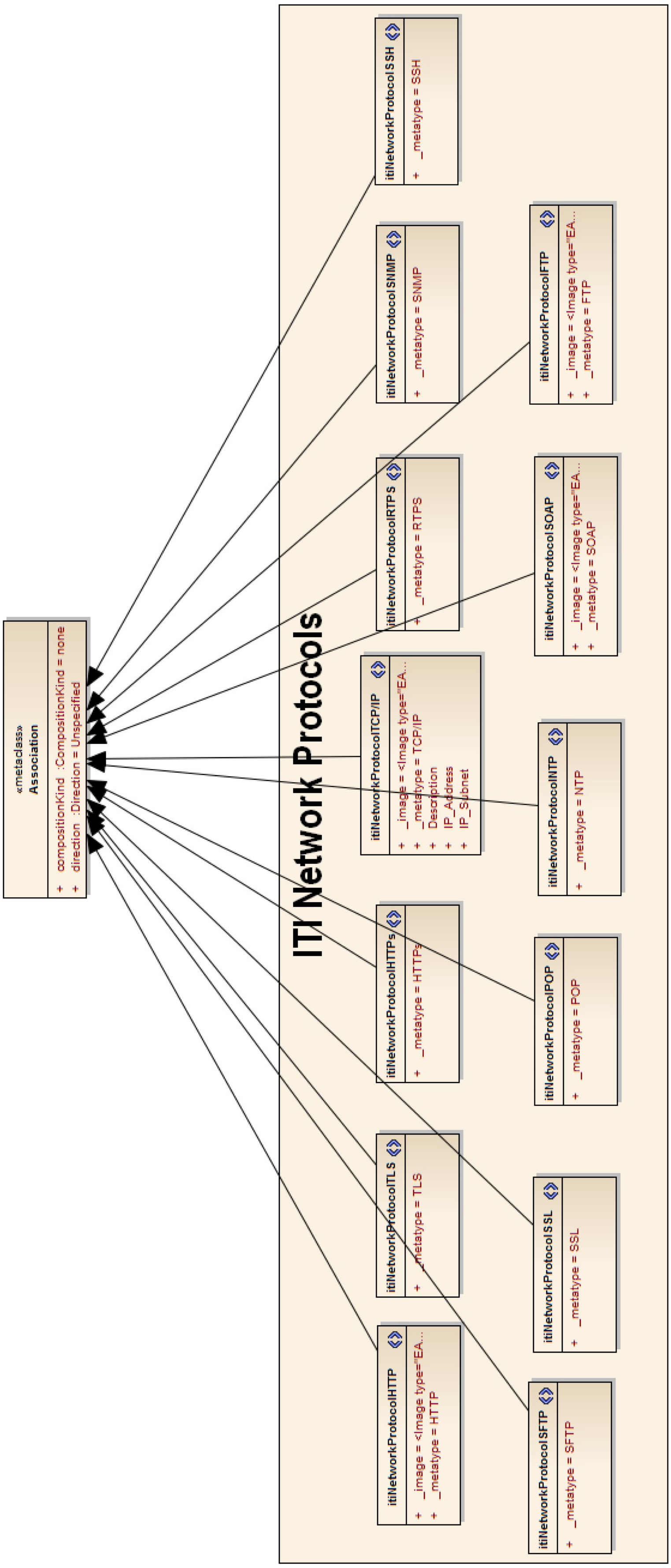


Figure C.9: ITI network protocol metaclasses

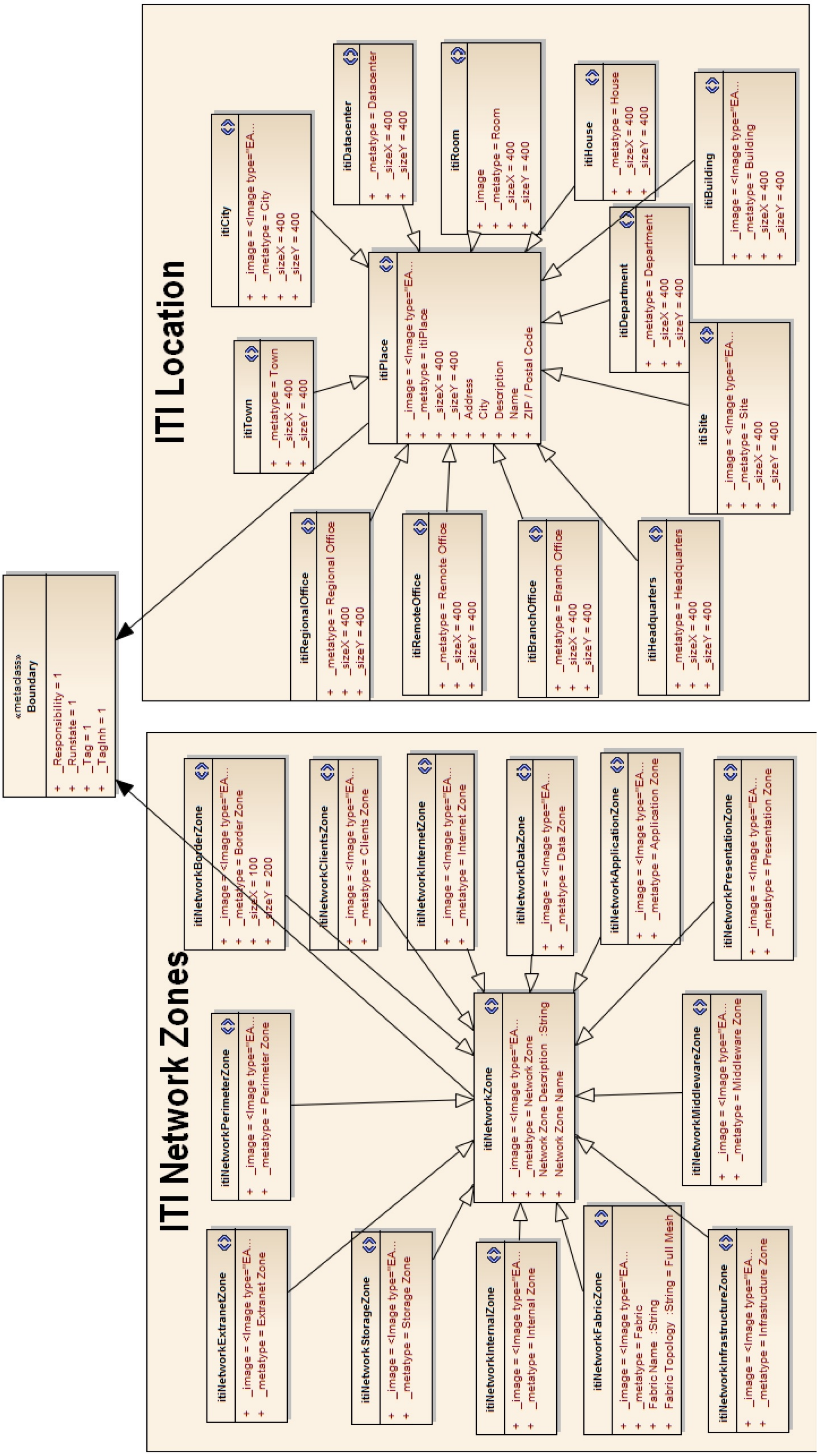


Figure C.10: ITI network zones and ITI location metaclasses

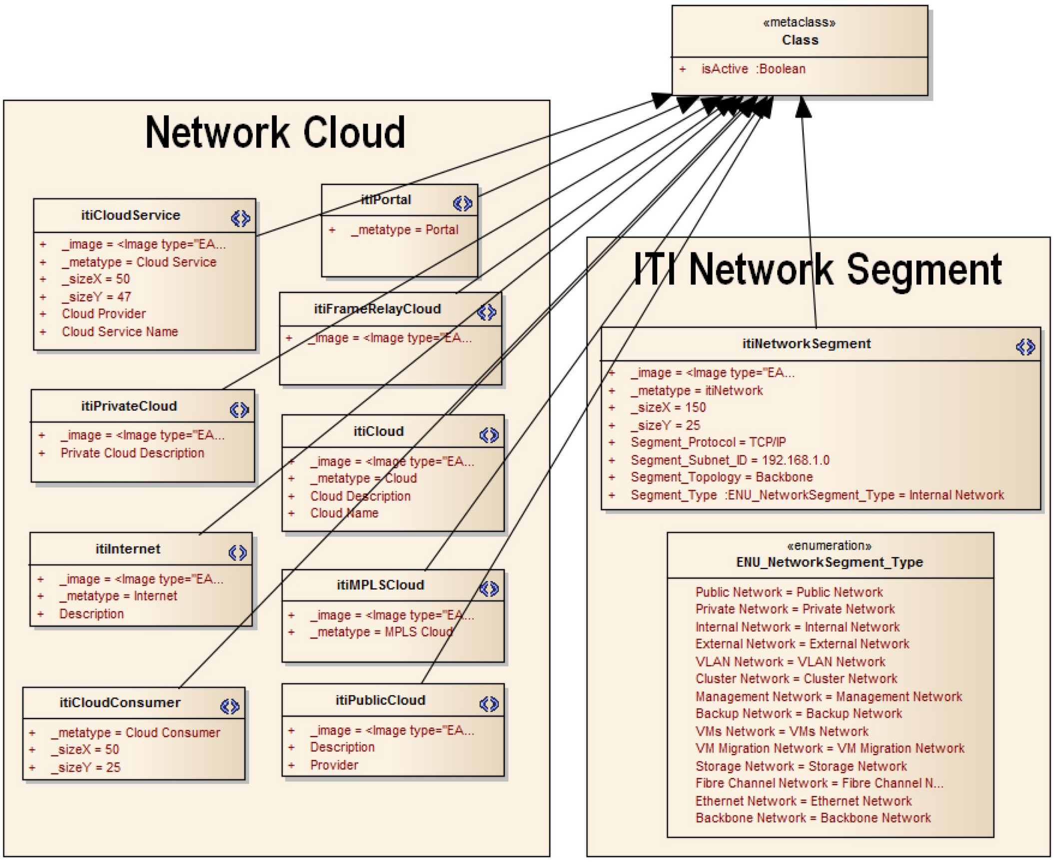


Figure C.11: ITI network segment metaclasses

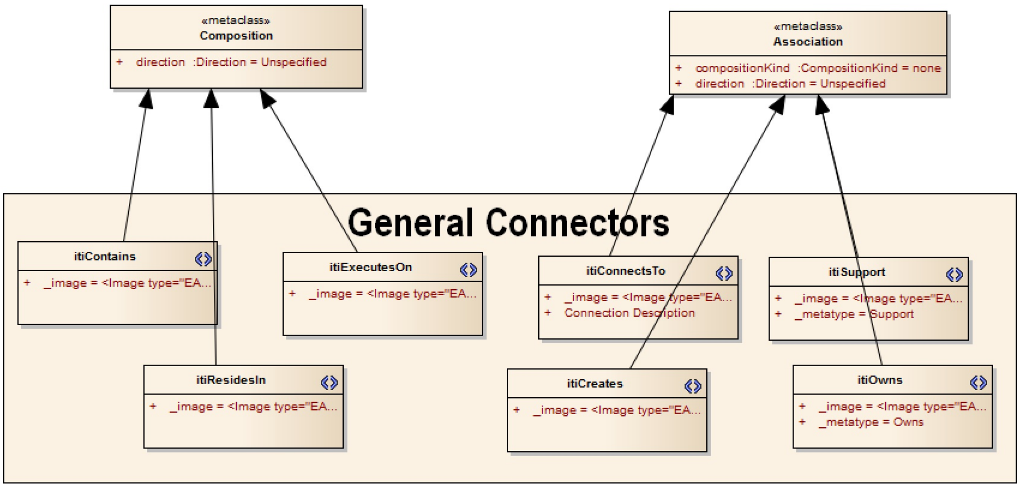


Figure C.12: Add ITI pattern from resources

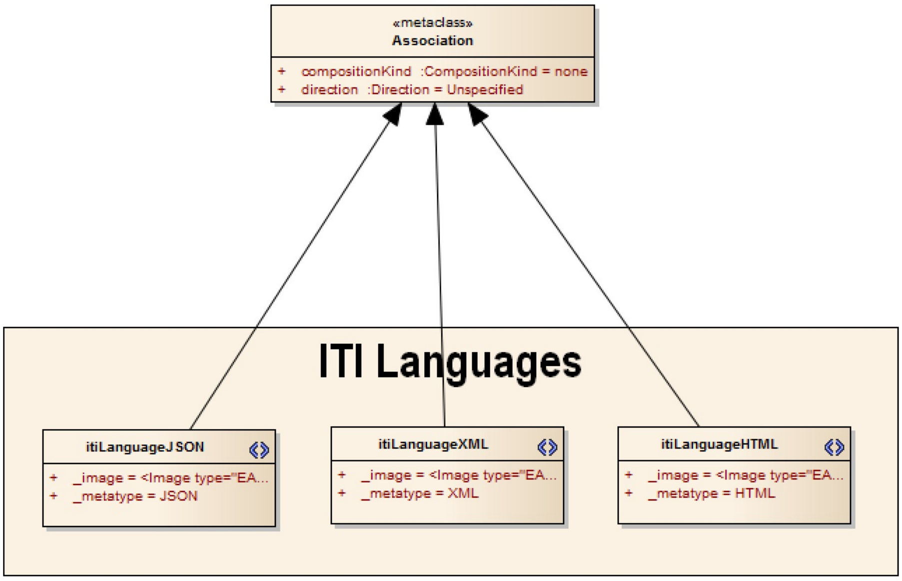


Figure C.13: ITI languages metaclasses

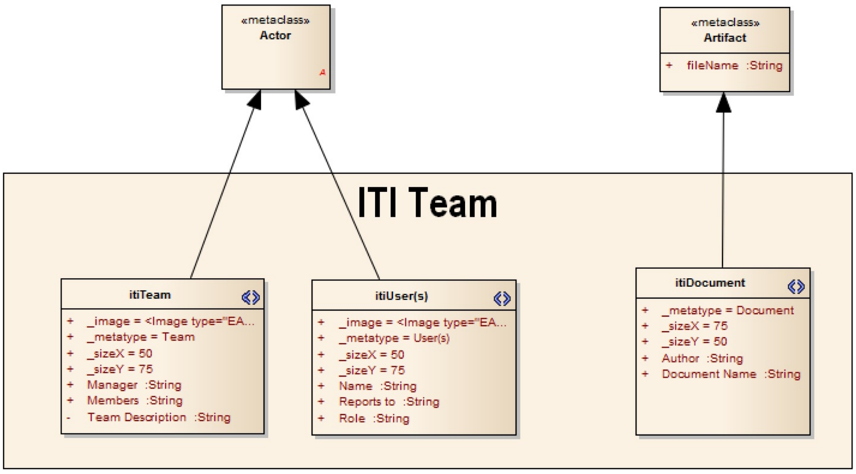


Figure C.14: ITI teams metaclasses

C.2 Diagram Toolbox

The ITI profile contains a diagram toolbox which is a panel of objects each with a specific icon that are used to create elements and connectors on the ITI diagram. Within the Toolbox, related elements and connectors are organized into pages, each page containing the elements or connectors used for a particular type of diagram. Notice that when a diagram is opened, the ITI toolboxes open automatically providing the elements and relationship pages corresponding to the diagram type.

In the toolbox of figure C.15 each object contains a different icon. The images can be associated to the ITI element or ITI connector or to the toolbox page, by assigning the special attribute icon to the stereotype element. This image definition for the Toolbox item can be overridden or replaced by extending the *ToolboxItemImage Metaclass*.

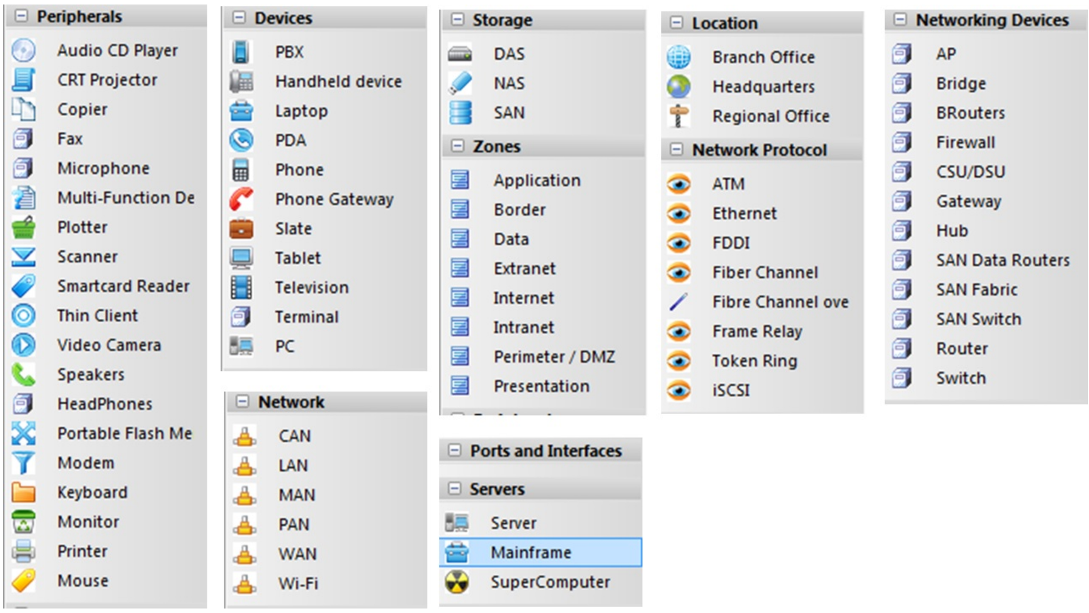


Figure C.15: Toolbox example 1

We decided to continue evolving the ITI metamodel and reflecting the changes in the toolbox in terms of number of ITI elements and relationships and move to future work the creation and association of icons to the toolbox which we considered optional to this research work. If there is no association to the icon attribute of the *ToolboxItemImage Metaclass*, the image defaults to a standard model image as can be seen in figure C.16.

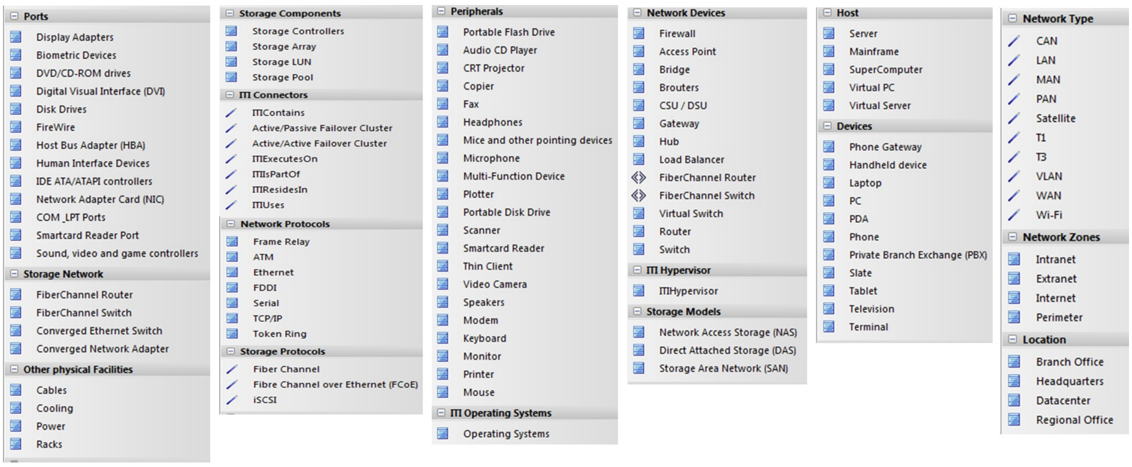


Figure C.16: Toolbox example 2

Appendix D

ITI Models and Pattern Diagrams

This appendix provides examples of ITI models created in our tool based on the ITI profile and ITI patterns. The following examples are provided in this chapter:

- ITI business intelligence architecture
- ITI network architecture
- ITI baremetal deployment
- ITI broad access
- ITI storage
- ITI perimeter network
- ITI remote software distributor
- ITI security teams
- ITI cloud portal
- ITI cloud load balancing
- ITI update domain
- ITI virtualization host
- ITI redundant cluster
- ITI load balancing
- ITI Global reference architecture
- ITI Global reference architecture (zoom 1)
- ITI Global reference architecture (zoom 2)
- ITI Global reference architecture (zoom 3)
- ITI Global reference architecture (zoom 4)

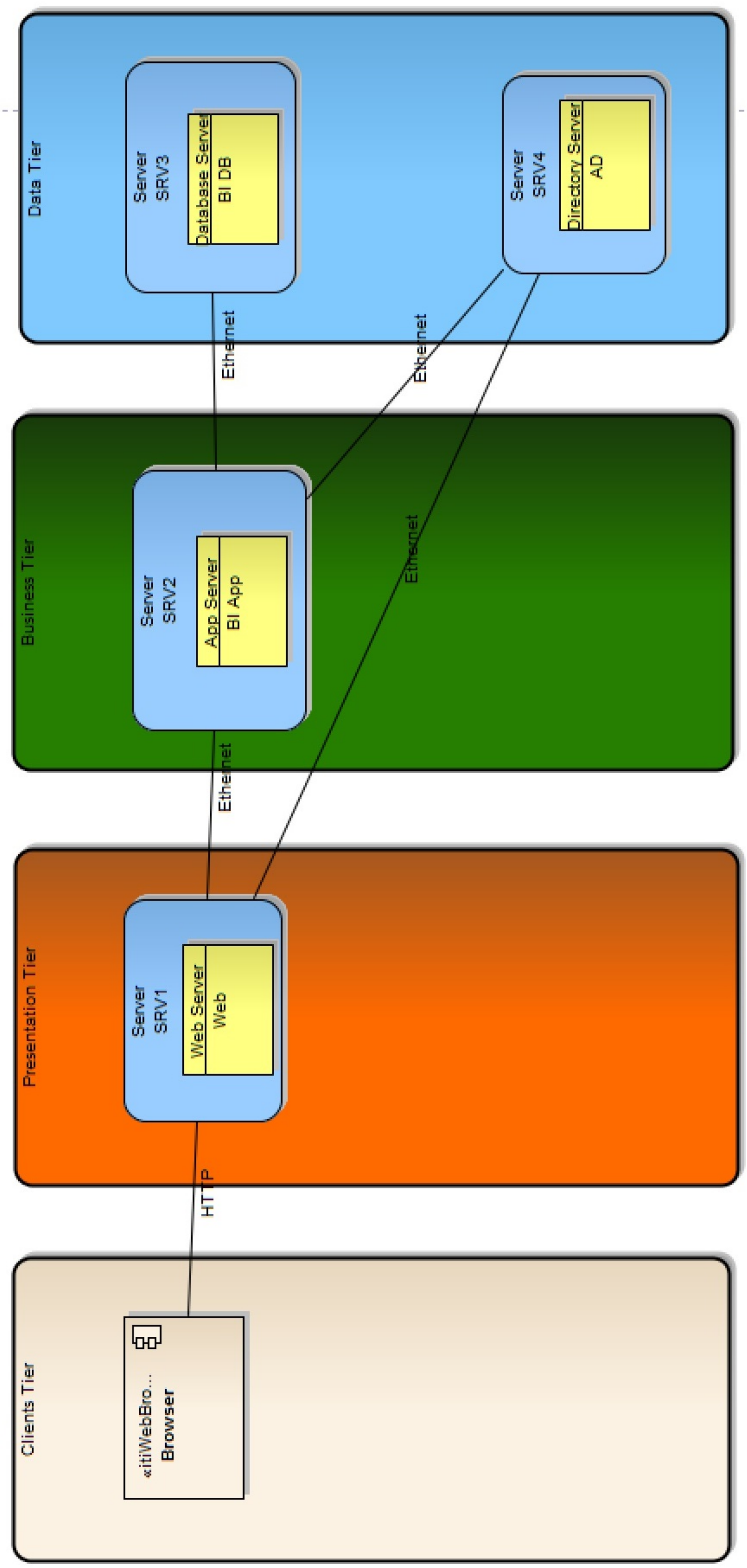


Figure D.1: ITI business intelligence architecture

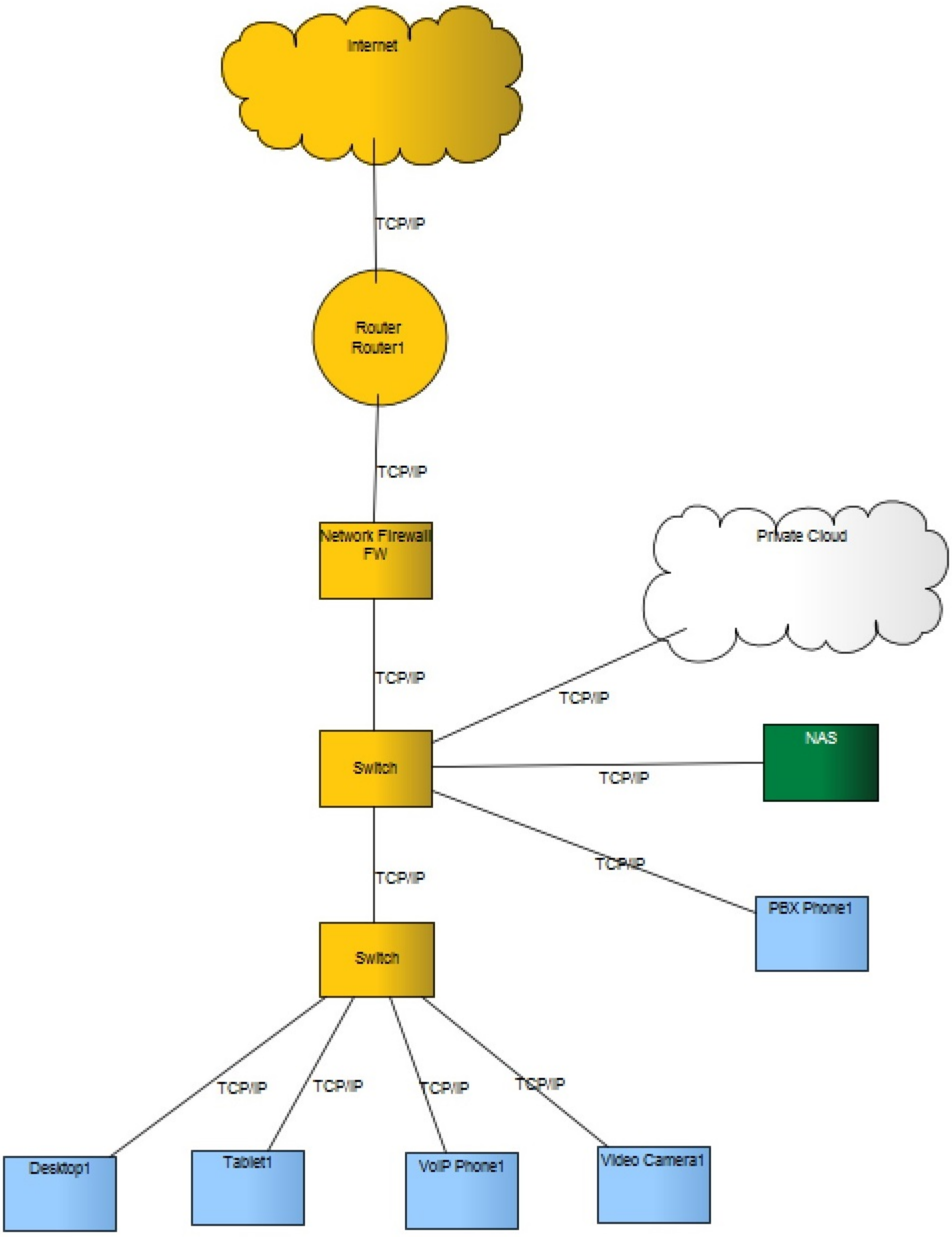


Figure D.2: ITI network architecture

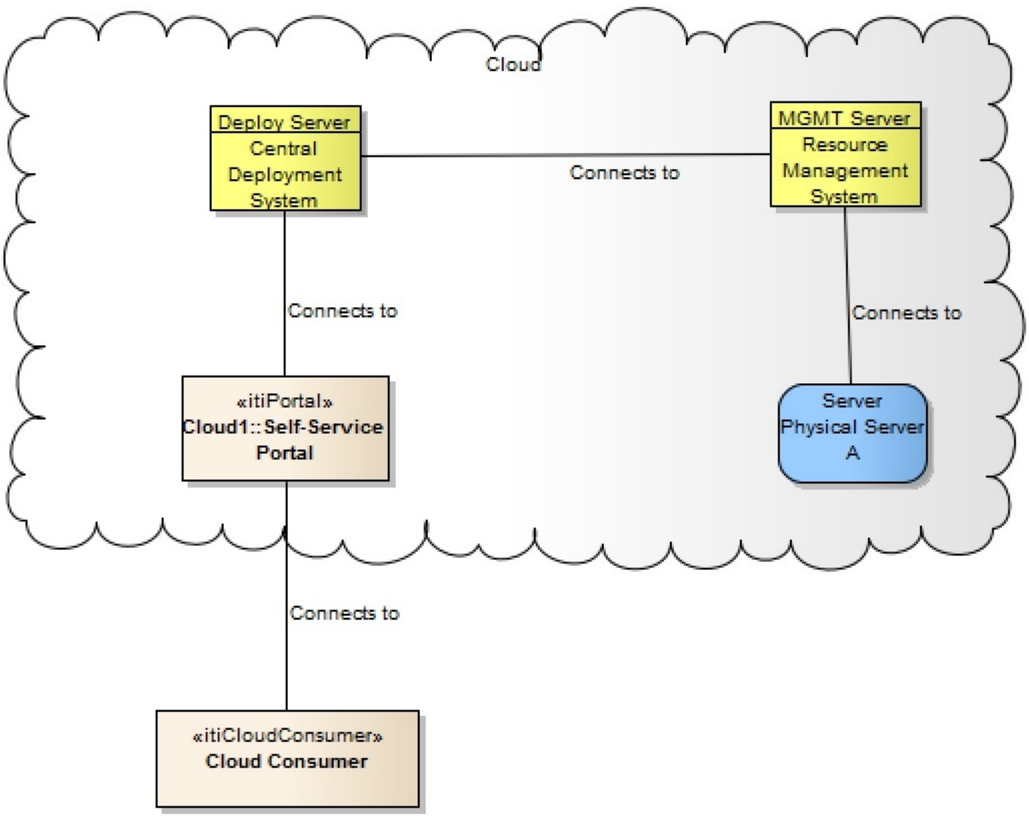


Figure D.3: ITI baremetal deployment

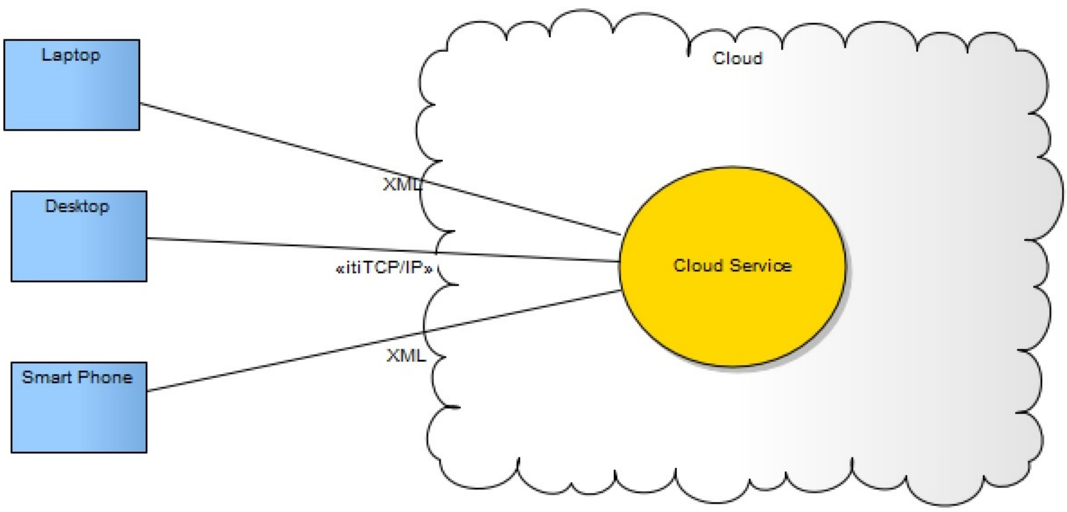


Figure D.4: ITI broad access

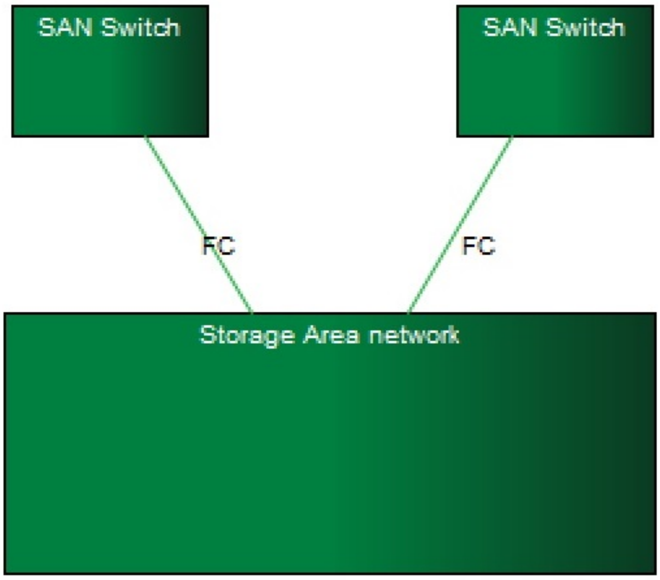


Figure D.5: ITI storage

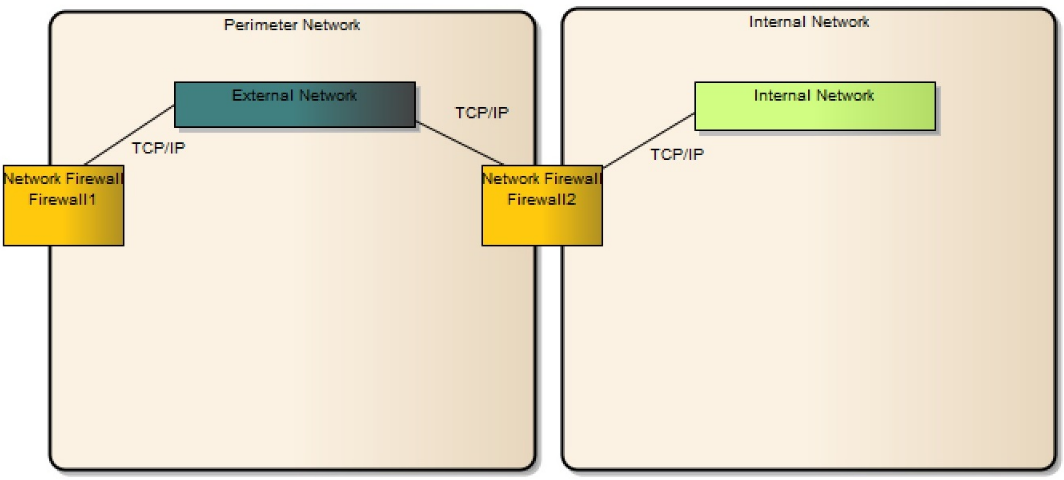


Figure D.6: ITI perimeter network

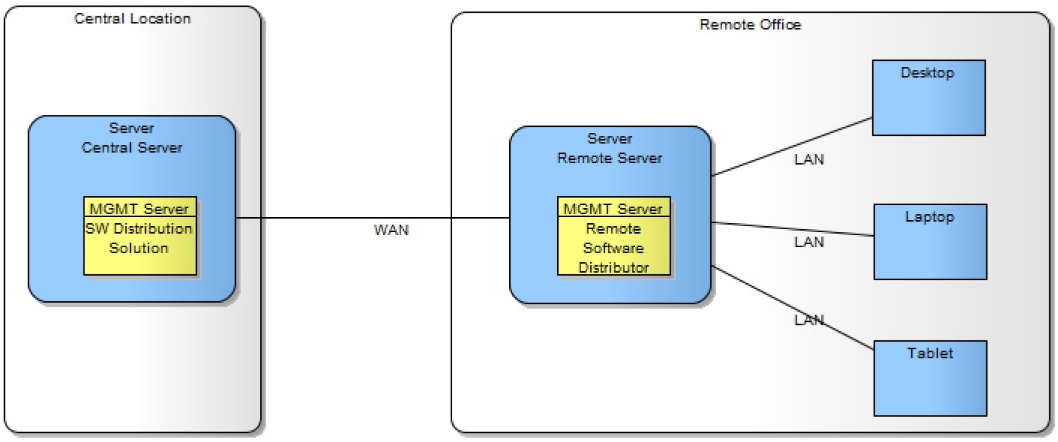


Figure D.7: ITI remote software distributor

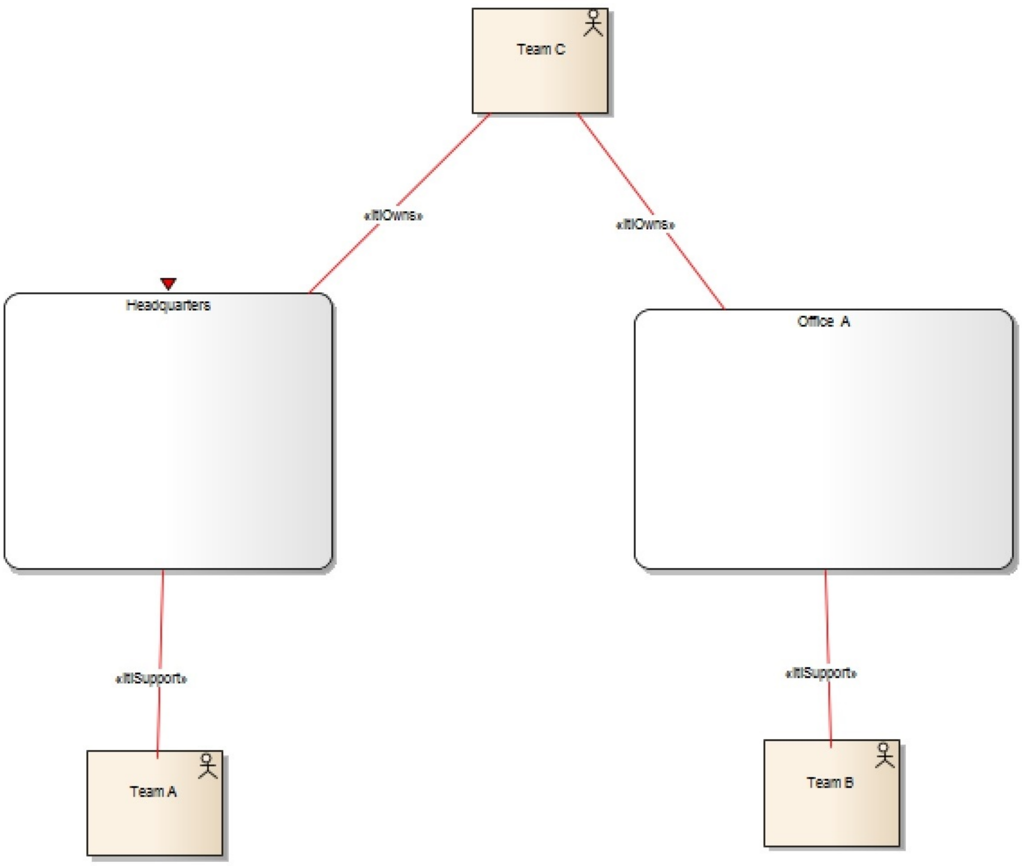


Figure D.8: ITI security teams

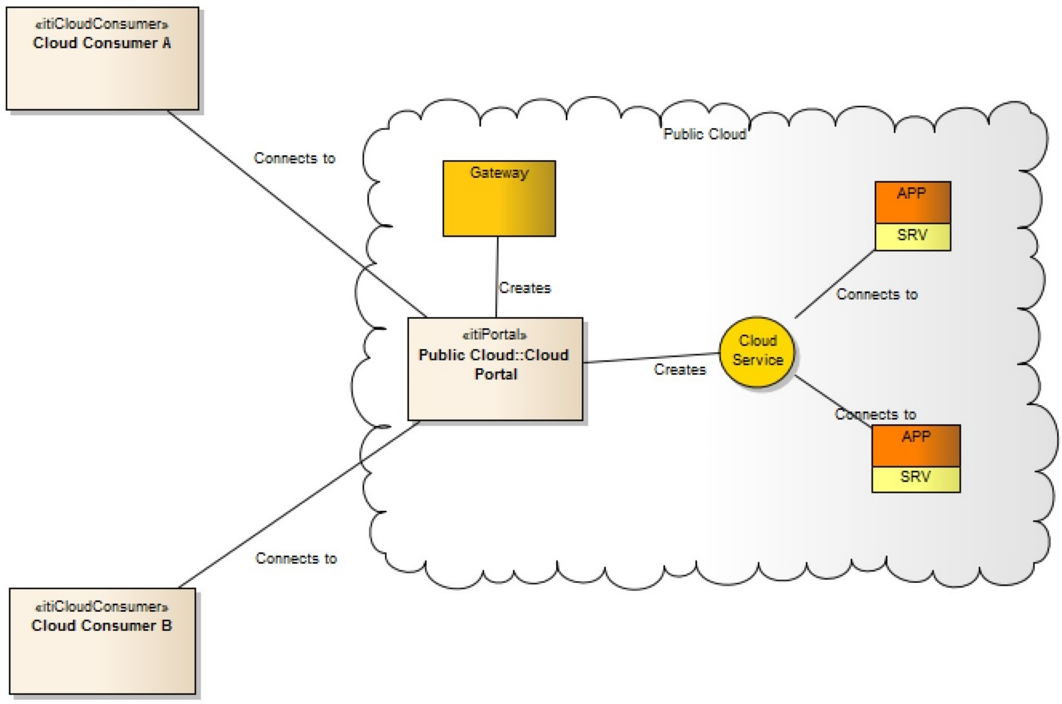


Figure D.9: ITI cloud portal

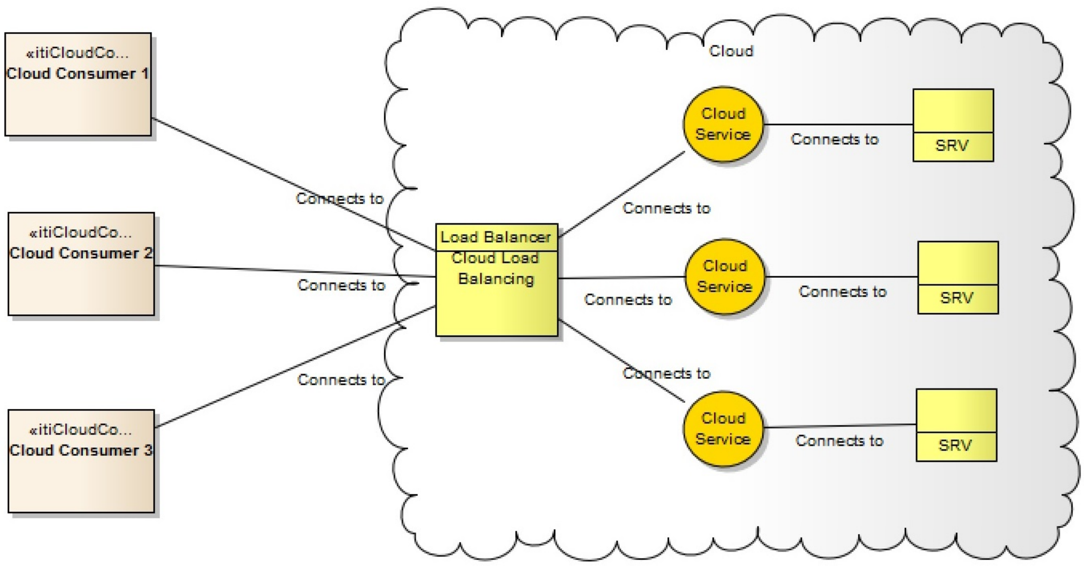


Figure D.10: ITI cloud load balancing

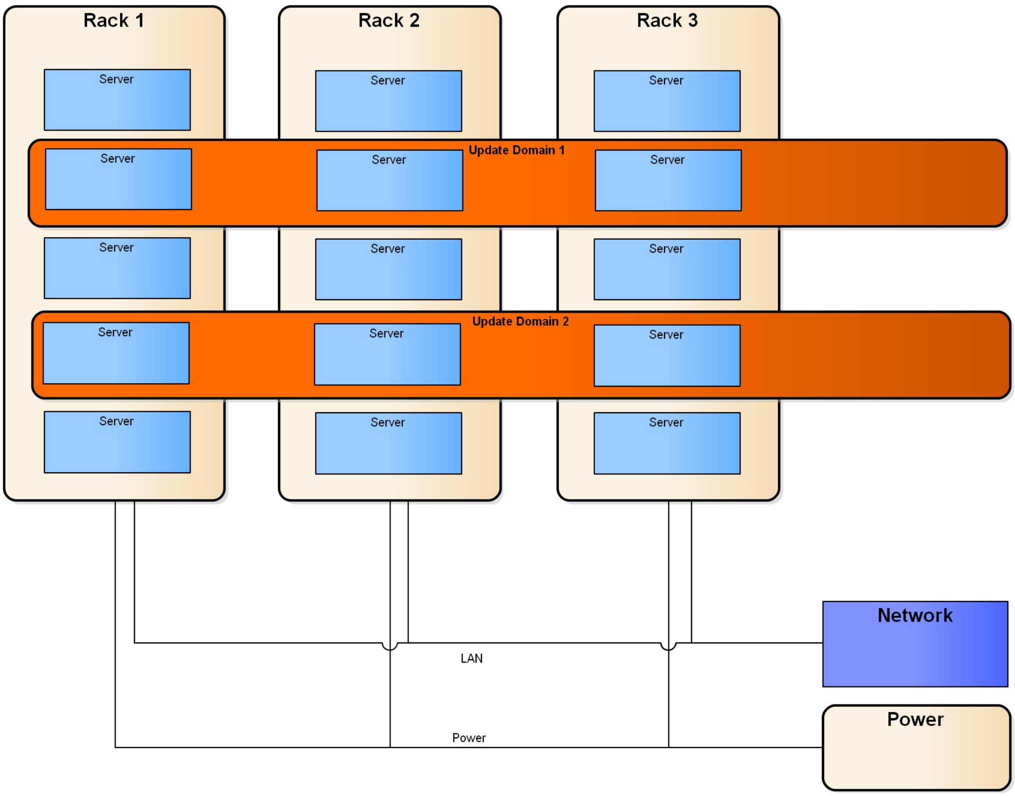


Figure D.11: Update domain

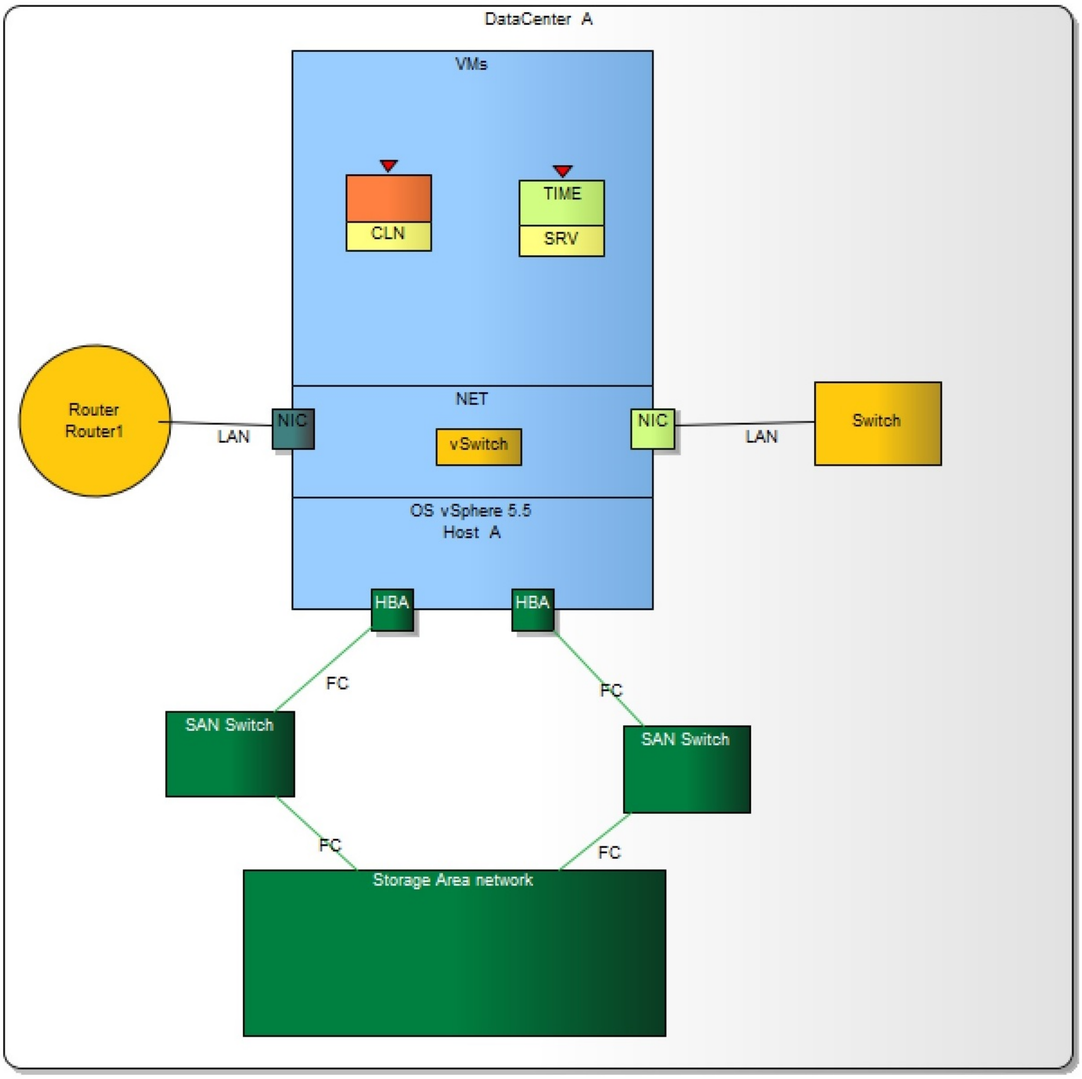


Figure D.12: ITI virtualization host

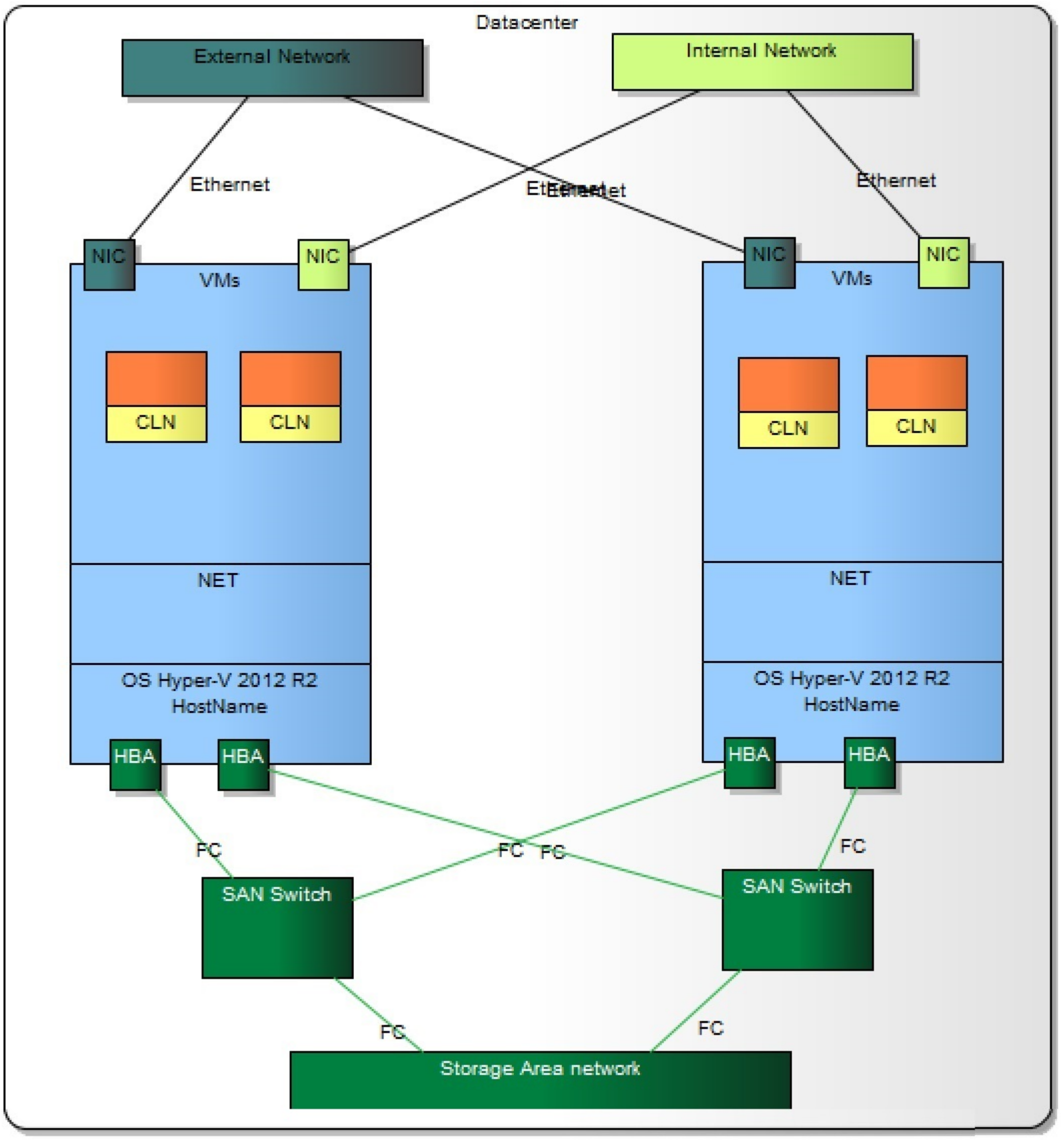


Figure D.13: ITI redundant cluster

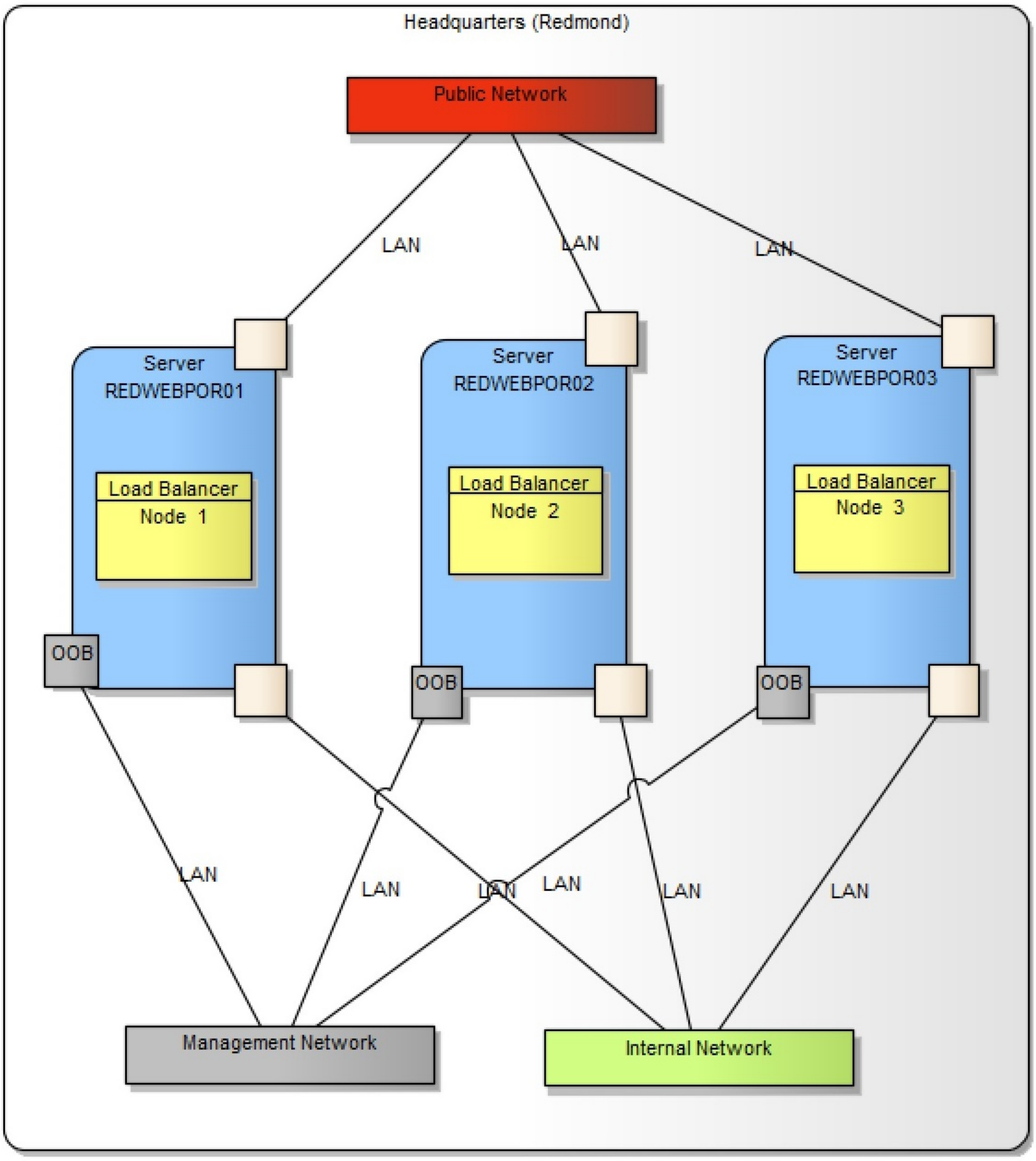


Figure D.14: ITI load balancing

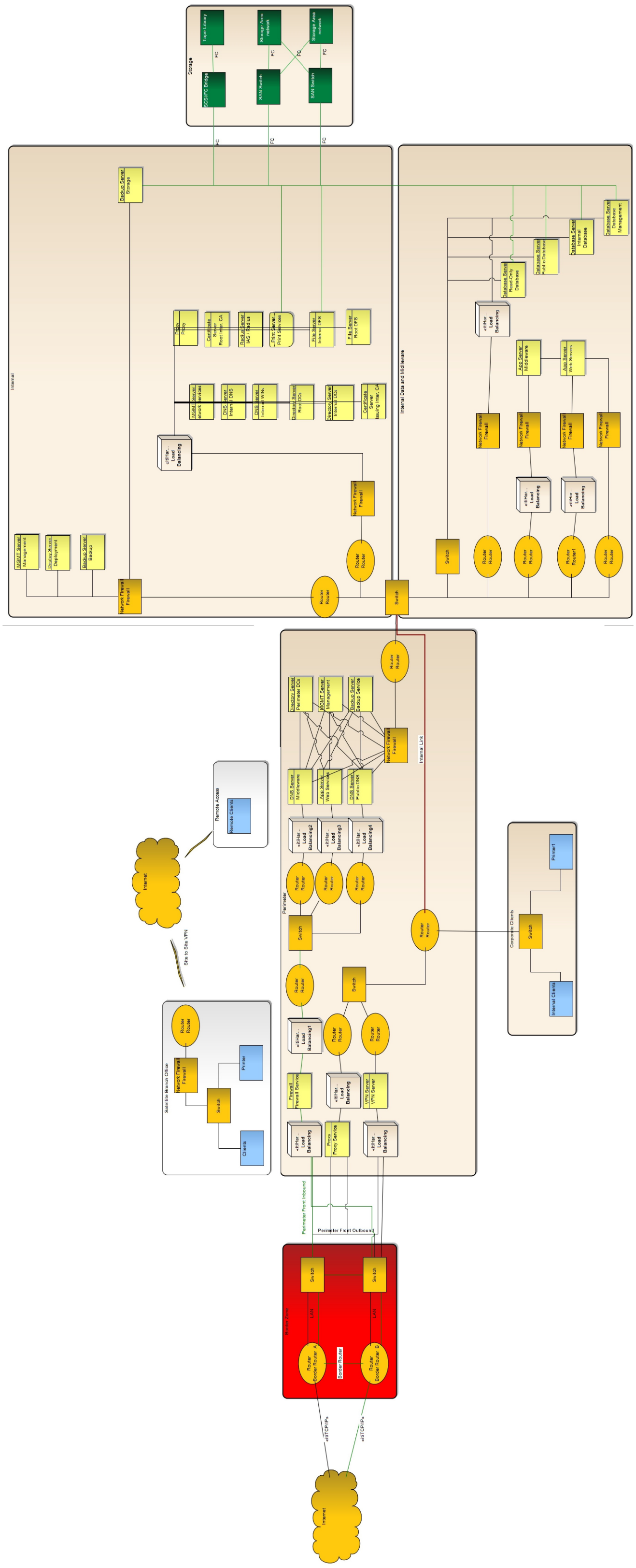


Figure D.15: ITI global reference architecture

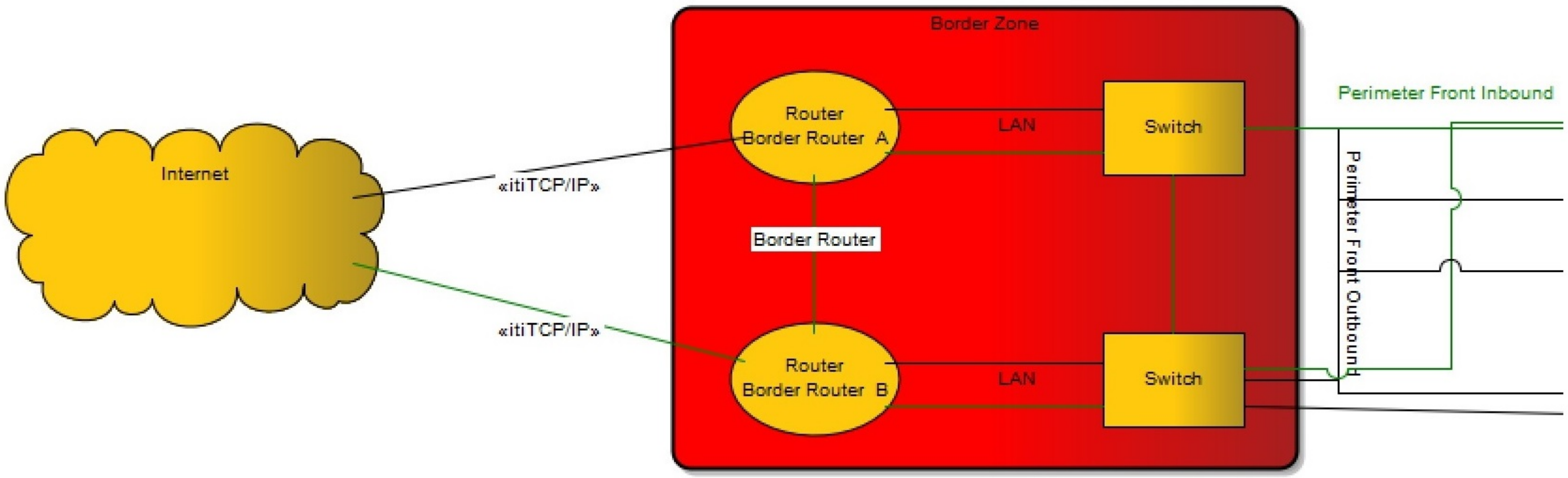


Figure D.16: ITI global reference architecture (zoom1)

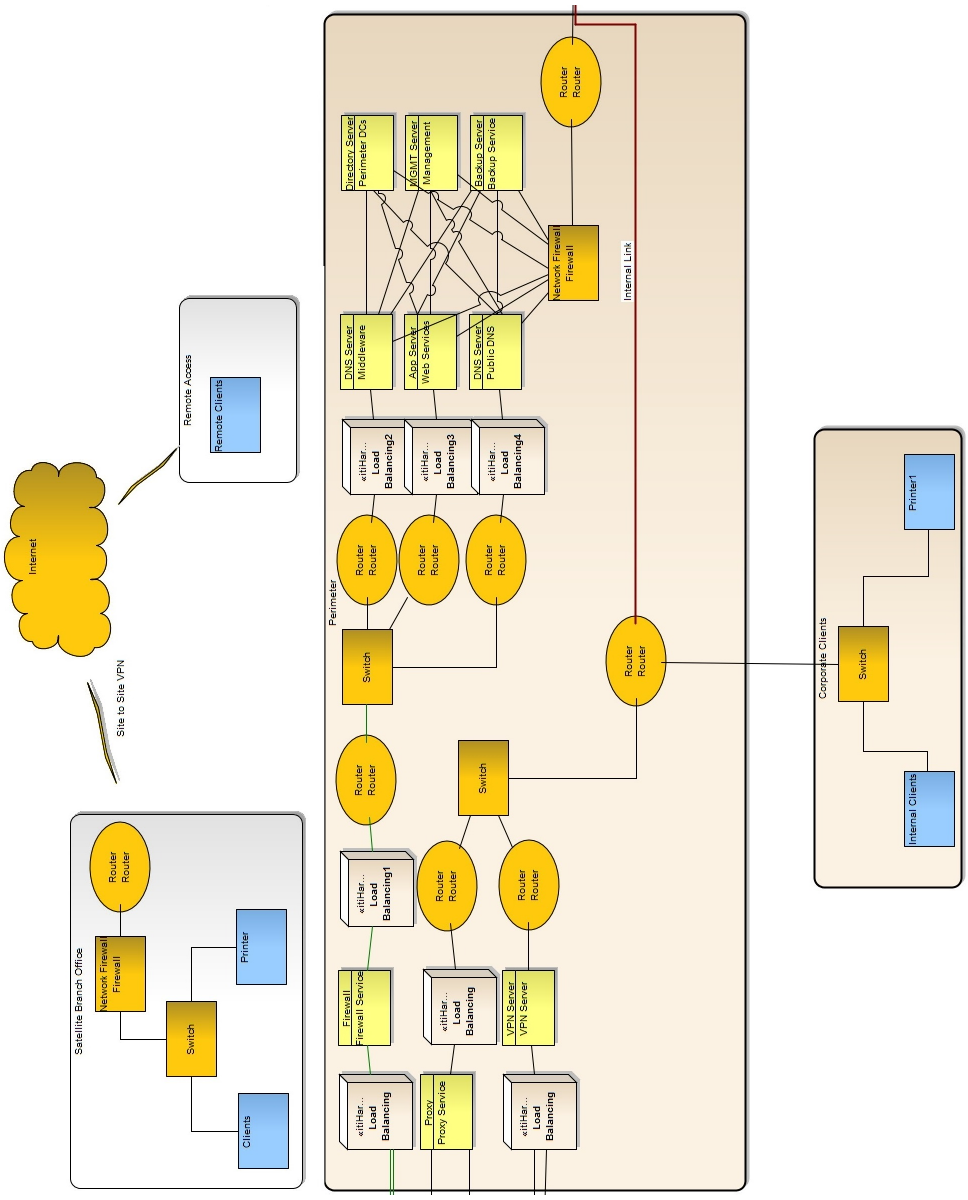


Figure D.17: ITI global reference architecture (zoom2)

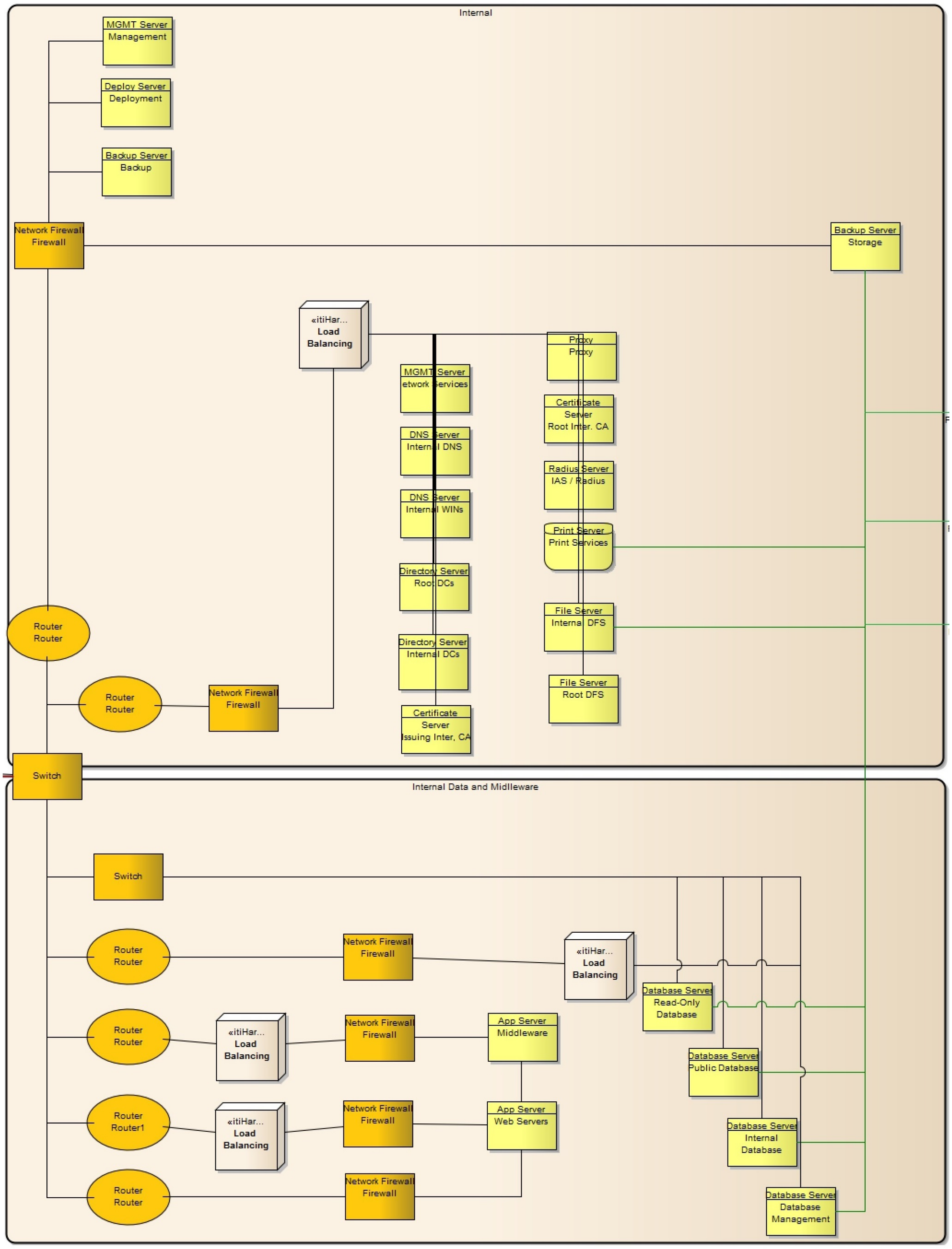


Figure D.18: ITI global reference architecture (zoom3)

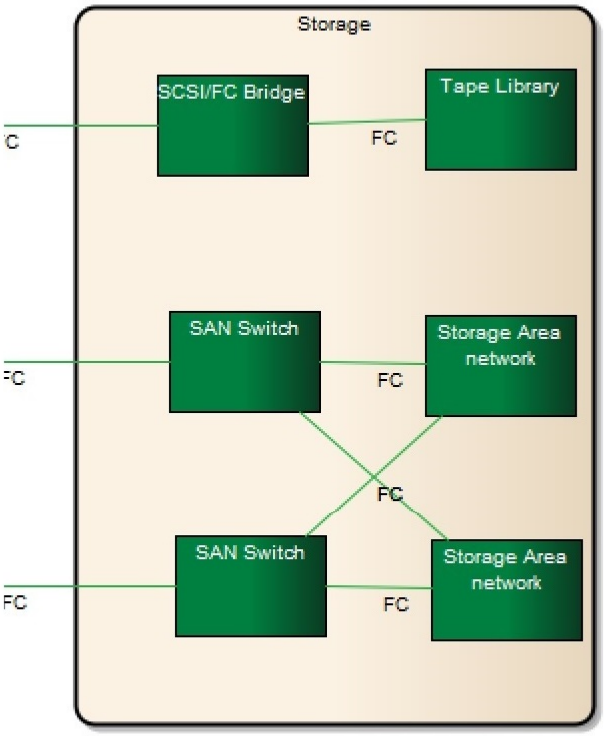


Figure D.19: ITI global reference architecture (zoom4)

Appendix E

Controlled Experiment Materials

This appendix, present details about the controlled experiment, performed to validate the results. It includes *(i)* the description of the five assignments used in the experiment, *(ii)* the solution for each assignment, the *(iii)* questionnaire answered by each participant and the results of the questionnaire.

E.1 Introduction

This section provides the five assignments that supported the experiment. There was an introduction to the organization named *Fictitious* (section E.2) and then the following assignments were delivered to each participant, according to the plan:

- Assignment A - Scientists (section E.2.1)
- Assignment B - OilSemaphore (section E.2.2)
- Assignment C - New Remote Offices (section E.2.3)
- Assignment D - Internet Access (section E.2.4)
- Assignment E - Payroll application (section E.2.5)

Each of the participants in the experiment will have a different assignment and to make sure that the results of each participant are comparable, each assignment has a similar level of complexity. More details will be provided in section E.4 regarding the level of complexity of each assignment, but the table E.1 provides the comparison.

Table E.1: Level of complexity comparison

Assignments	ITI Components	Mandatory Attributes	Optional Attributes	Possible Values
Assignment A	28	22	41	211
Assignment B	26	24	25	247
Assignment C	25	24	16	357
Assignment D	31	23	18	242
Assignment E	22	25	15	183

E.2 Assignments

Fictitious is a global enterprise in the oil and gas business. As the number of Oil supplies reduces, this business is becoming more and more competitive. *Fictitious* is based on Redmond, WA, operates in 50 locations worldwide and employs 30,000 people. This company employs several scientists that are defining new approaches, algorithms and systems for evaluating prospect fields. When they finish their work, they want to perform tests and put their work in production as quickly as possible. The users and business are increasingly dependent on IT services provided by the *Fictitious* IT infrastructure and there are several mission critical legacy applications that are required to be available 24x7 worldwide. The access to information in real-time is crucial to their success.

Enterprise Challenges. The existing IT infrastructure is outdated with high costs associated and existing IT services are provided by hundreds of physical servers with manifest poor performance that no longer meet business needs and limit the organization growth potential. As an example, deploying new applications into production takes several weeks.

Goal. *Fictitious* realized that the more dependent they are on technology, the more critical the infrastructure is, so they decided to do several projects that gradually redesign and modernize IT infrastructure, based upon standard, proven solutions and best practices to help *Fictitious* achieve more efficiency and agility and improve IT services currently provided.

E.2.1 Assignment A - Scientists

Fictitious decided to modernize part of the infrastructure to address some of the issues. There are currently 20 servers, each with 4GB of RAM and 4 cores used by scientists for several proposes. The nomenclature used by *Fictitious* is in the format LLL-FFFF-SS, where LLL corresponds to 3 alphabetic characters for Location, FFFF corresponds to 4 alphanumeric characters to describe the equipment function and SS are a 2 numeric characters used to identify the sequence number. Globally *Fictitious* faces the following issues:

- According to existing contracts, every 4 years, the servers are replaced since they become out of support.
- Replacing hardware implies complex activities such as reinstalling operating systems and applications, applying configurations, migrating data among other activities which have an high probability of failure.
- *Fictitious* is running out of space in the current datacenters because the number of physical servers is increasing.
- Most servers are using only 25 percent of its processing capabilities.
- The electricity costs and hardware maintenance is also increasing.
- There are some applications that in some periods of time require more resources than those physically available.

With budget under pressure, *Fictitious* wants to keep costs on capital acquisitions to the minimum and when possible leveraging previous investments. Recently *Fictitious* invested in:

- **Network**. To comply with security, each type of infrastructure traffic should be carried over physically separate networks and network adapters.
- **Storage**. 20 TB of Traditional SAN storage and Fiber Channel that *Fictitious* would like to preserve. Fiber Channel has historically been the storage protocol of choice due to their performance and low latency.
- **Servers**. There are four server/blade hardware with HBAs each with 12 CPU cores (two sockets), 192 GB of RAM, 2 on-board disks each with 300GB.

The goal is to design a highly available hardware and network infrastructure to support these 20 servers, while leveraging existing investments.

E.2.2 Assignment B - OilSemaphore

One of the most important IT services provided by *Fictitious* IT infrastructure is monitoring. There is an application that is responsible for monitoring the flow, pressure, process units and temperature and provide alerts in case of failures. The application is called *OilSemaphore* is deployed in the Europe datacenter and consists of 10 servers. The nomenclature used by *Fictitious* is in the format LLL-FFFF-SS, where LLL corresponds to 3 alphabetic characters for Location, FFFF corresponds to 4 alphanumeric characters to describe the equipment function and SS are a 2 numeric characters used to identify the sequence number. Globally *Fictitious* faces the following issues:

- According to existing contracts, every 4 years, the servers are replaced since they become out of support.
- Replacing hardware implies complex activities such as reinstalling operating systems and applications, applying configurations and migrating data, among other activities, which have an high probability of failure.
- *Fictitious* is running out of space in the current datacenters because the number of physical servers are increasing.
- Most servers are using only 25 percent of its processing capabilities.
- The electricity costs and hardware maintenance is also increasing.
- There are some applications that in some periods of time require more resources than what those physically available.

Due to the importance of this application, *Fictitious* intends to reformulate the IT infrastructure and when possible reduce its complexity. The plan is:

- Reduce the number of network adapters without sacrificing performance.
- Isolate traffic from each other.
- Use cost-effective and resilient storage solution.
- The servers have to be continuously available and resilient to hardware failures.

The goal is to design a highly available hardware and network infrastructure to support the *OilSemaphore* application.

E.2.3 Assignment C - New Remote Offices

Fictitious plans to open two new offices in the Middle East and since they will not have local IT staff, a cost-effective way to distribute software to those new locations is required. There is already a software distribution solution in-place and the plan is to extend it taking in consideration the following requirements:

- There are 10 client systems in each office
- Network links between headquarters and these new remote offices have a total of 256K bandwidth.
- Special care must be taken to make sure that the software distribution process does not overload the existent network link.
- The software distribution process must be automated, without manual activities.
- The local Internet Service Provider (ISP) run their service on a low quality/high latency network so that the available bandwidth does not exceed 256K.

E.2.4 Assignment D - Internet Access

Fictitious needs to expose some organization's services to the Internet, such a public website and because security is a major concern, *Fictitious* would like to protect the internal network from unwanted traffic using two firewalls. Since there are some concerns regarding performance, robustness, resilience, and support for high-speed connectivity *Fictitious* have two Internet service providers for redundancy and already acquired the following network devices for the border zone:

- 2 routers.
- 2 switches.

The desirable nomenclatures are:

- Border Routers: FFL-Border-Router-A and FFL-Border-Router-B
- Border Switches: FFL-Border-Switch-A and FFL-Border-Switch-B

The public IP addresses are:

- Internet Service Provider 1 Downstream: 209.217.184.2.
- Internet Service Provider 2 Downstream: 209.217.134.2.

The management IP addresses for border routers should be:

- 192.168.23.25
- 192.168.23.26

The management IP addresses for border switches should be:

- 192.168.23.27
- 192.168.23.28

The network devices should be distributed across the following racks:

- FFL-CP-RACK-01
- FFL-CP-RACK-02

The firewall that separates the border zone from the perimeter network should have the following configurations:

- Name: FFL-SA-FWP01
- Border Network IP Address: 208.217.184.25
- Border Network Subnet: 208.217.184.16/28
- Border Network Gateway: 208.217.184.19
- Perimeter Network IP Address: 192.168.12.170
- Perimeter Network Subnet: 192.168.12.0/24
- Perimeter Network DHCP: 192.168.12.97 and 192.168.12.45
- Perimeter Network Gateway: 192.168.12.3

E.2.5 Assignment E - Payroll application

To support the new payroll application *Fictitious* needs to design an infrastructure. Since this application is critical, the infrastructure must support the following operational requirements:

- **Architecture.** The application was developed to work over multiple servers rather than over multiple processors on a multiprocessor computer.
- **24x7 Payroll application availability.** Since *Fictitious* is a global organization the payroll application must be always available. Problems that cause downtime have a severe monetary impact.
- **No single points of failure.** A failure in a server should not affect the Payroll availability.
- **Capacity to scale.** Since the number of users may increase, the capacity to scale without affecting application availability is important.
- **Virtualization.** The design should include physical servers, instead of virtual environments.

It is important that the design takes in consideration that both hardware and software require maintenance activities such as software upgrades or hardware replacements. None of these activities should cause application downtime.

E.3 Results

As was mentioned before, each participant will solve different exercises using the ad-hoc, profile or pattern approach. The results achieved are measured in three different categories:

- **ITI Components.** ITI components or ITI objects are the main building blocks, such as a data centers, networks, virtualization hosts, SAN, etc.
- **Mandatory attributes.** The attributes considered mandatory are those that allow the identification of an ITI component such as the name of a server or network, the IP address in case of a NIC, etc.
- **Optional attributes.** All the other attributes that can contribute to a better understanding of the representation are considered optional.

More details regarding the results and assignments are provided in appendix E.4.

Table E.2: Participant's results on assignment A

Participant	Type	ITI Components	Mandatory Attributes	Optional Attributes	Results Achieved	Results %
P.1	Ad-hoc	26	16	0	**42**	19.9%
P.4	Profile	28	19	41	**88**	41.7%
P.2	Pattern	28	22	41	**153**	72.5%

Table E.3: Participant's results on assignment B

Participant	Type	ITI Components	Mandatory Attributes	Optional Attributes	Results Achieved	Results %
P.2	Ad-hoc	25	23	0	**48**	19.4%
P.5	Profile	26	18	19	**63**	25.5%
P.3	Pattern	26	24	25	**138**	55.8%

Table E.4: Participant's results on assignment C

Participant	Type	ITI Components	Mandatory Attributes	Optional Attributes	Results Achieved	Results %
P.3	Ad-hoc	24	12	0	**36**	10%
P.1	Profile	25	24	15	**64**	17.9%
P.4	Pattern	25	24	15	**204**	57.1%

Table E.5: Participant's results on assignment D

Participant	Type	ITI Components	Mandatory Attributes	Optional Attributes	Results Achieved	Results %
P.4	Ad-hoc	28	23	0	**51**	21%
P.2	Profile	29	21	16	**66**	27.2%
P.5	Pattern	31	23	16	**143**	59%

Table E.6: Participant's results on assignment E

Participant	Type	ITI Components	Mandatory Attributes	Optional Attributes	Results Achieved	Results %
P.5	Ad-hoc	21	20	0	**41**	22.4%
P.3	Profile	21	24	15	**60**	32.7%
P.1	Pattern	22	25	15	**144**	78.6%

E.4 Solutions

To give an idea of what was expected from each exercise, this section shows the ITI diagram with the correct design. The evaluation of each participant was based on the number of components and the number of attributes filled that are not visible in the diagrams. Notice that we balance the level of complexity of the assignments as can be seen in the level of complexity metrics available for each exercise. The main reason to balance the level of complexity was because we want to compare the results of multiple experiments. The decision to use multiple assignments was because each participant was used in multiple experiments and we wanted to block the learning effect to not compromise results.

E.4.1 Assignment A - Scientists Solution

Table E.7 shows the main ITI objects and connectors of Assignment A. The proposed design (figure E.1) consists of *Hosts* capable of running virtual machines connected to an internal and external network using *Ethernet* and having dedicated connections via *Fiber Channel* to the storage.

Table E.7: Level of complexity (Assignment A - Scientists)

Assignment A ITI objects	**ITI Components**	**Mandatory Attributes**	**Optional Attributes**	**Possible Values**
Datacenter	1	1	3	5
Virtualization Host	2	4	6	64
NIC on Host 1	2	2	4	28
NIC on Host 2	2	2	4	28
HBA on Host 1	2	0	4	10
HBA on Host 2	2	0	4	10
SAN Switches	2	2		18
SAN Storage	1	1		16
External Network	1	1		4
Internal Network	1	1		4
VM A - Host 1	1	3		4
VM B - Host 2	1	3		4
Total ITI Objects	**18**	**20**	**25**	**195**
ITI connectors				
Ethernet	4	2	4	4
Fiber Channel	6	0	12	12
Total connectors	**10**	2	**16**	**16**
Total	**28**	**22**	**41**	**211**

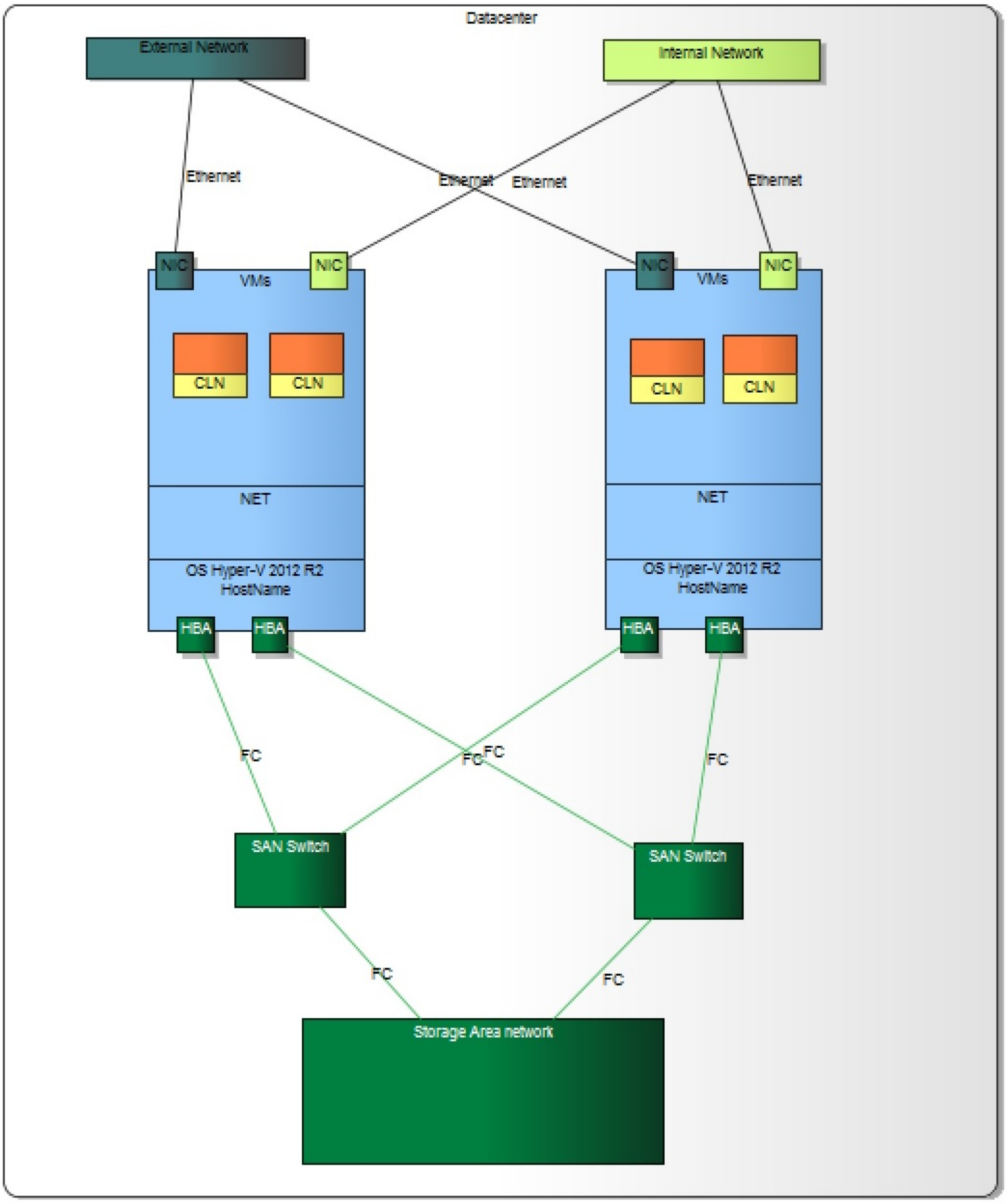


Figure E.1: ITI diagram (Assignment A - Scientists)

E.4.2 Assignment B - OilSemaphore Solution

Table E.8 shows the main ITI objects and connectors of Assignment B. The proposed design (figure E.2) consists of two database servers connected to switches via *Ethernet* and are connected via *iSCSI* to storage.

Table E.8: Level of complexity (Assignment B - OilSemaphore)

Assignment B ITI objects	**ITI Components**	**Mandatory Attributes**	**Optional Attributes**	**Possible Values**
Remote Office	1	1	3	5
Servers	2	10	6	64
NIC Server 1	3	3	6	42
NIC Server 2	3	3	6	42
Converged Switches	2	2	4	32
SAN	1	1		16
Cluster Network	1	1		4
Public Network	1	1		4
Database	2	2		8
Total ITI Objects	**16**	**24**	**25**	**217**
ITI connectors				
VLAN	4	0	0	12
iSCSI	6	0	0	18
Total connectors	**10**	**0**	**0**	**30**
Total	**26**	**24**	**25**	**247**

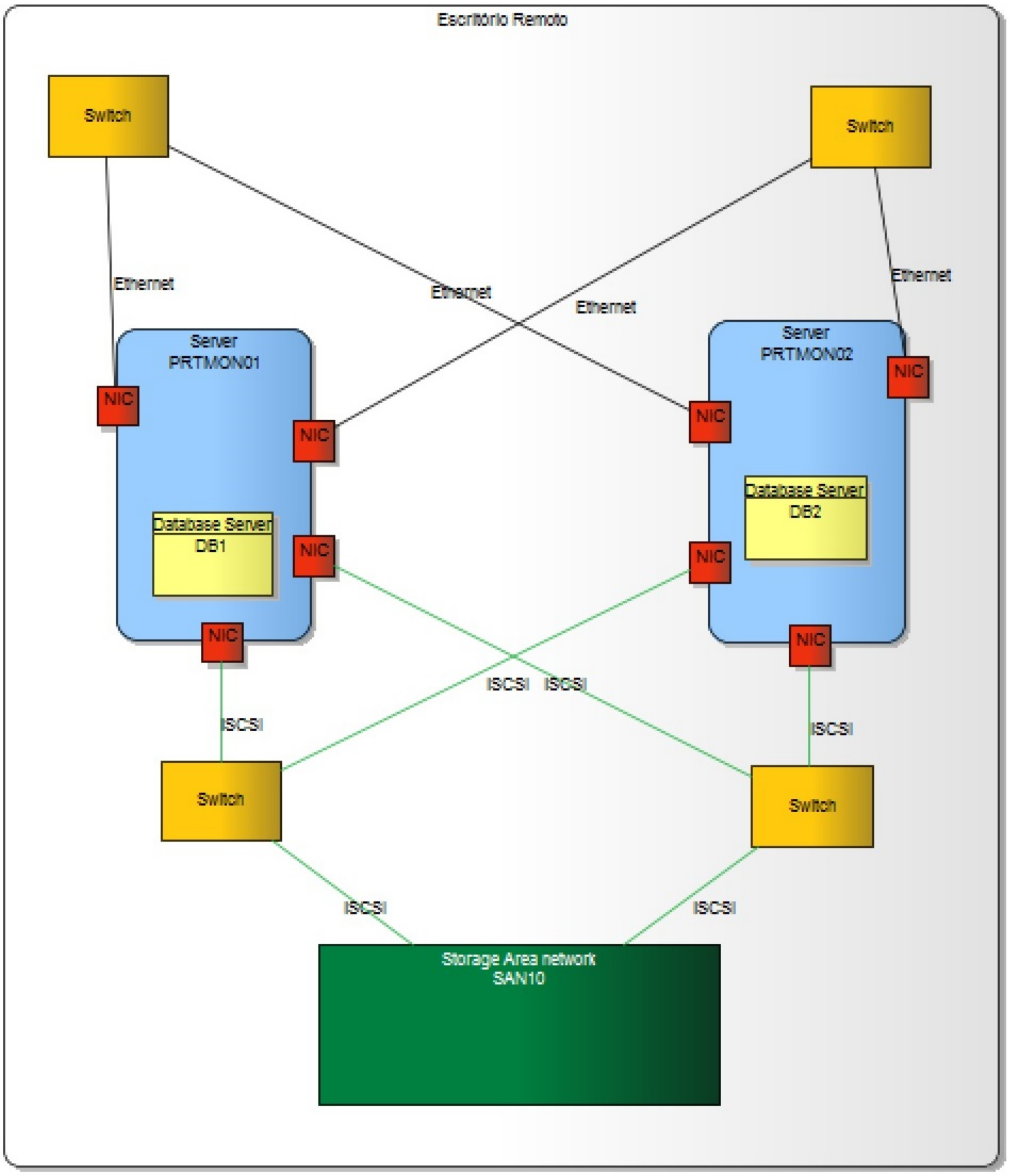


Figure E.2: ITI diagram (Assignment B - OilSemaphore)

E.4.3 Assignment C - New Remote Offices Solution

Table E.9 shows the main ITI objects and connectors of Assignment C. The proposed design (figure E.3) consists of an MultiProtocol Label Switching (MPLS) connection between headquarters and a remote office where multiple devices are connected to a server using an internal network.

Table E.9: Level of complexity (Assignment C - New Remote Offices)

Assignment C ITI objects	**ITI Components**	**Mandatory Attributes**	**Optional Attributes**	**Possible Values**
Datacenter	1	1	3	5
Remote Offices	2	2	6	10
Server 1	1	1	2	32
Server 2	1	1	2	32
Server 3	1	1	2	32
App. Software	3	12		12
Desktops	3	3		111
Ultrabooks	3	3		117
Cloud Internet	1	0		1
Total ITI Objects	**16**	**24**	**15**	**352**
ITI connectors				
LAN	6	0	0	0
WAN	2	0	0	0
ISP	1	0	1	5
Total connectors	**9**	**0**	**1**	**5**
Total	**25**	**24**	**16**	**357**

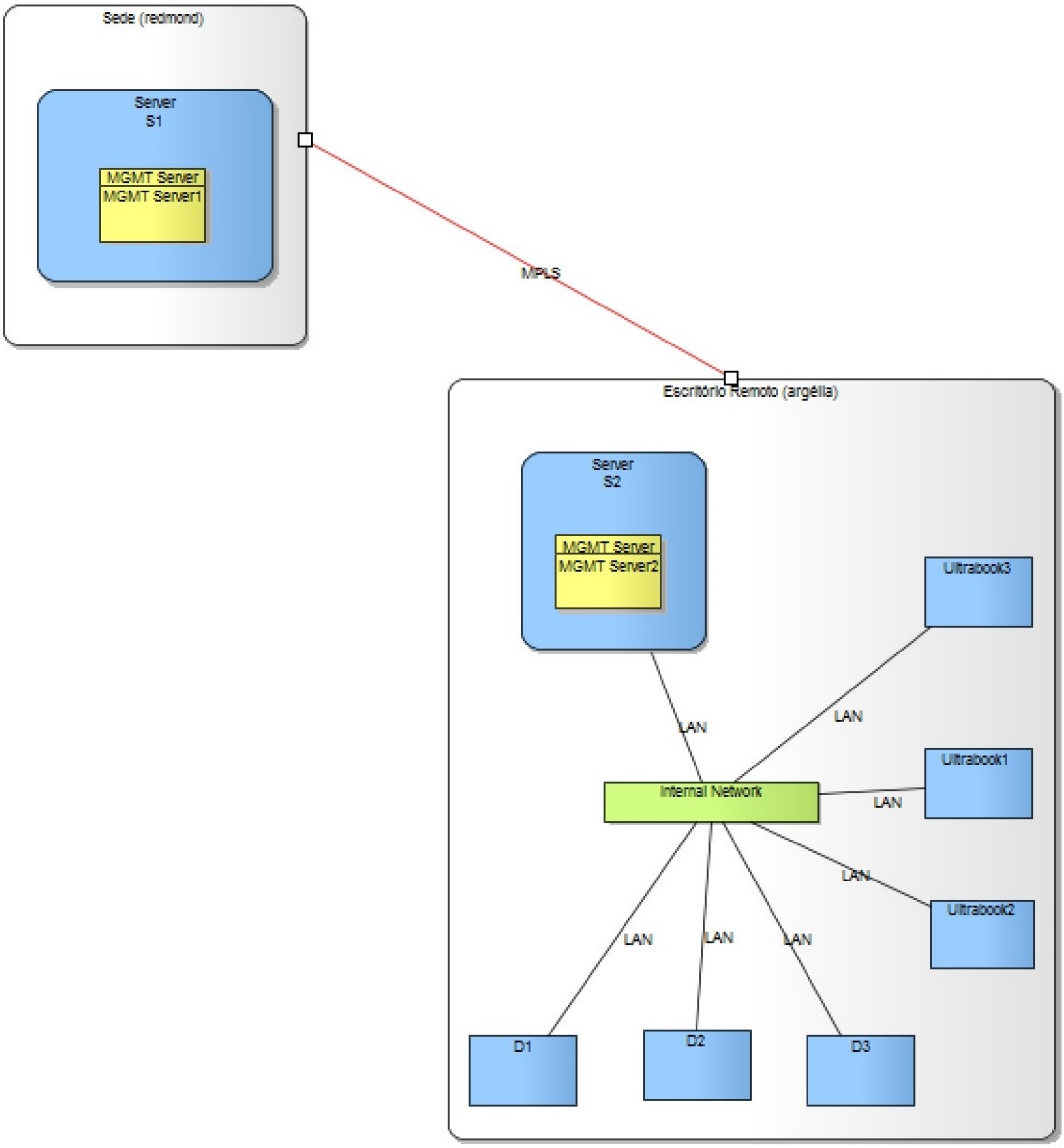


Figure E.3: ITI diagram (Assignment C - New Remote Offices)

E.4.4 Assignment D - Internet Access Solution

Table E.10 shows the main ITI objects and connectors of Assignment D. The proposed design (figure E.4) and consists mostly in network components such as *routers* and *switches* providing access to the Internet securely, using redundant connections.

Table E.10: Level of complexity (Assignment D - Internet Access)

Assignment D ITI objects	**ITI Components**	**Mandatory Attributes**	**Optional Attributes**	**Possible Values**
Border Zone	1	1		2
Perimeter Zone	1	1		2
Intranet Zone	1	1		2
Routers	2	6	4	28
Switches	2	4		32
Firewall	2	2	4	24
External Network	1	1		4
Internal Network	1	1		4
Cloud Internet	1	0		1
Servers	2	2	4	64
NICs	2	2	4	28
Web Software	2	2		8
Total ITI Objects	**18**	**23**	**16**	**199**
ITI connectors				
TCP/IP	11		33	
ISP	2	2	10	
Total connectors	**13**	**0**	2	**43**
Total	**31**	**23**	**18**	**242**

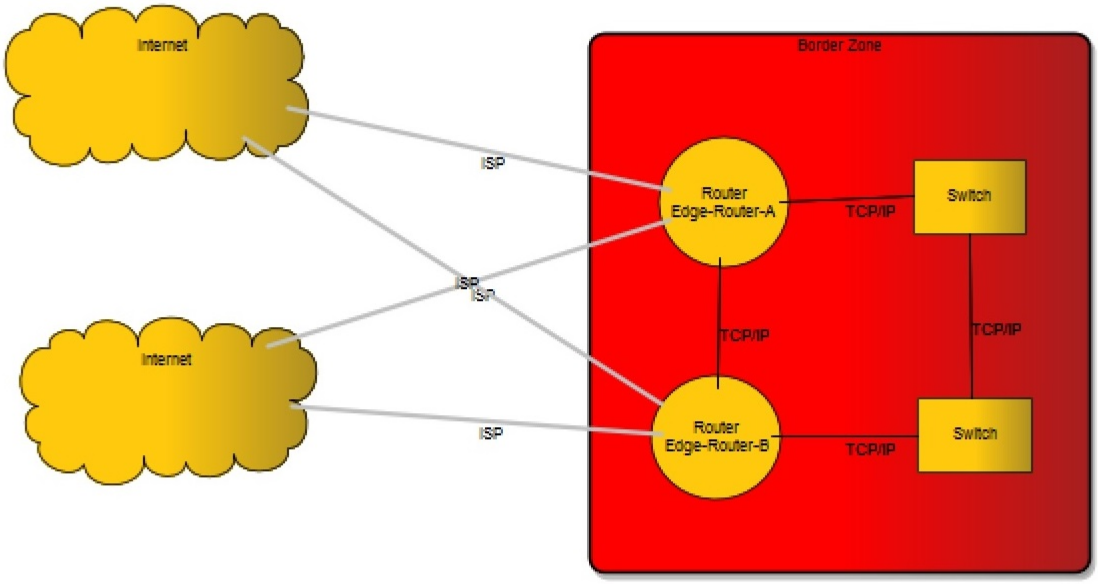


Figure E.4: ITI diagram (Assignment D - Internet Access)

E.4.5 Assignment E - Payroll Application Solution

Table E.11 shows the main ITI objects and connectors of Assignment E. The proposed design (figure E.5) consists in the representation of a location with the name *Lisboa* with two servers supporting websites through the use of *webservers* balanced with load balancing technology for high availability purposes.

Table E.11: Level of complexity (Assignment E - Payroll Application)

Assignment E ITI objects	**ITI Components**	**Mandatory Attributes**	**Optional Attributes**	**Possible Values**
Site	1	1	3	5
Server 1	1	2	2	32
Server 2	1	2	2	32
NIC Server 1	2	2	4	28
NIC Server 2	2	2	4	28
NIC OOB 3	2	2		22
Public Network	1	2		4
Internal Network	1	2		4
Mgm Network	1	2		4
Load Balancing	2	6		16
Application Server	2	2		8
Total ITI Objects	**16**	**25**	**15**	**183**
ITI connectors				
LAN	6	0	0	0
Total connectors	**6**	**0**	**0**	**0**
Total	**22**	**25**	**15**	**183**

E.5 Wrap-up Questionnaire

This instrument belongs to the Wrap-up phase of the experimental design and was designed to gather more information from the participants after the experiment. It contains questions related with their previous experience and questions regarding the experiments themselves.

Questionnaire

Please take a few minutes to fill out this questionnaire, which is intended to gather more information regarding the current methods, practices, experiences during the process of designing an IT infrastructure (ITI). This questionnaire also collects data about the exercises performed. Thank you for your participation.

Personal Information

1. Level of education/degree?

☐ Doctoral ☐ Master ☐ Graduate ☐ High School ☐ Other

2. How long are you working in the IT industry (Years)?

____

3. Excluding design activities, how long are you working with IT infrastructures (Years)?

____

4. How long are you working in the design of IT infrastructures (Years)?

____

Design of IT infrastructures

5. On average how many IT infrastructures sketches/diagrams do you create per month?

☐ 1 to 2 ☐ 3 to 4 ☐ 5 to 6 ☐ 7 to 8 ☐ More than 8

6. Are IT infrastructures sketches/diagrams carefuly revised before sending them to the customer?

☐ Never ☐ Almost never ☐ Sometimes ☐ Almost always ☐ Always

7. How much effective work time do you require to create an IT infrastructure sketch/diagram?

☐ Less than 1 day ☐ 1 to 2 days ☐ 3 to 5 days ☐ 6 to 15 days ☐ More than 15 days

8. What level of formality is normally applicable to sketches/diagrams produced? (The formality is added by following strict rules from a modeling language, such UML for instance)

☐ Very informal ☐ Informal ☐ Neutral ☐ Formal ☐ Very formal

9. When you request information about an existing IT infrastructure the result is often?

☐ Sketches ☐ Diagrams ☐ Textual specifications ☐ Other

10. How often the sketch/diagram is the main outcome of an infrastructure design project?

☐ Never ☐ Less than 25% ☐ 26 to 50% ☐ 51 to 75% ☐ More than 75%

11. Thinking about the time spent and comparing with benefits of sketches and diagrams, how do you classify the effort spen in making those diagrams?

☐ Completely useless ☐ Useless ☐ Neutral ☐ Useful ☐ Very useful

12. Considering the IT infrastructure sketches/diagrams that you have access to, how many contain enough information to support the implementation?

☐ None ☐ Less than 25% ☐ 26 to 50% ☐ 51 to 75% ☐ More than 75%

Experimental exercise

13. Please rate the difficulty of the ad-hoc exercise

☐ Very easy ☐ Easy ☐ Neutral ☐ Difficult ☐ Very difficult

14. Please rate the difficulty of the profile exercise

☐ Very easy ☐ Easy ☐ Neutral ☐ Difficult ☐ Very difficult

15. Please rate the difficulty of the patterns exercise

☐ Very easy ☐ Easy ☐ Neutral ☐ Difficult ☐ Very difficult

Additional Feedback

Please share any additional comments.

Thank you for taking the time to fill out this questionnaire.

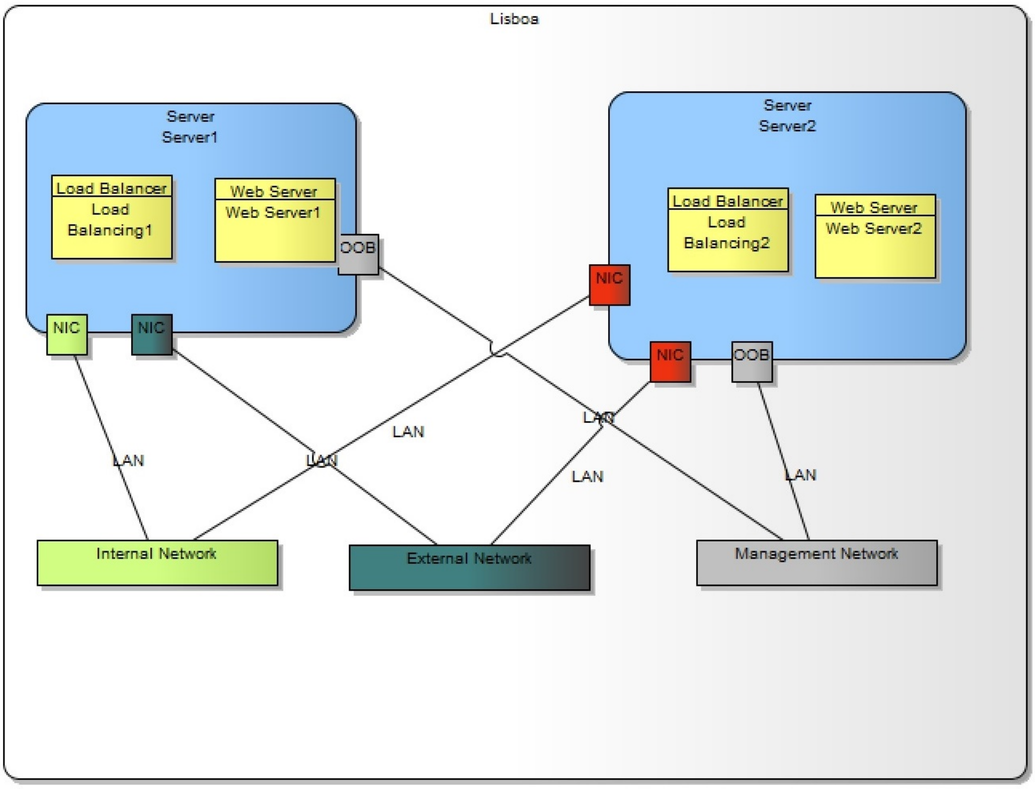


Figure E.5: ITI diagram (Assignment E - Payroll Application)

E.6 Questionnaire Results

The questionnaire results are summarized in the following tables. The header contains the number of the question with one or two words to help identify the question.

Table E.12: Personal information

Participant	Q1-Education	Q2-Years working	Q3-Years designing
1	Master	14	3
2	Graduate	3	2
3	High School	15	7
4	High School	30	10
5	High School	14	4

Table E.13: Experimental exercise results (questions 13,14,15)

Participant	**Type**	**Exercise**	**Perceived Difficulty**
1	Ad-hoc	Scientists	Very Difficult
1	Profile	New Remote Office	Neutral
1	Patterns	Internet Access	Very Easy
2	Ad-hoc	OilSemaphore	Difficult
2	Profile	Intranet	Easy
2	Patterns	Scientists	Very Easy
3	Ad-hoc	New Remote Office	Neutral
3	Profile	Internet Access	Easy
3	Patterns	OilSemaphore	Very Easy
4	Ad-hoc	Intranet	Difficult
4	Profile	Scientists	Easy
4	Patterns	New Remote Office	Very Easy
5	Ad-hoc	Internet Access	Neutral
5	Profile	OilSemaphore	Easy
5	Patterns	Intranet	Very Easy

Table E.14: Design of ITIs

Par.	Q5 Designs month	Q6 Diagram revised	Q7 Time to create	Q8 Formality	Q9 Common format	Q10 Main outcome	Q11 Time vs benefit	Q12 Implem-entation
1	4	Never	3 to 5 days	Very Infor.	Sketches	51 to 75%	Useful	Never
2	2	Always	1 to 2 days	Very Infor.	Sketches	26 to 50%	Completely Useless	Less than 25%
3	6	Sometimes	6 to 15 days	Formal	Word	Never	Neutral	26 to 50%
4	8	Almost Never	3 to 5 days	Neutral	Diagrams	51 to 75%	Useless	Never
5	2	Always	6 to 15 days	Very Info.r	Sketches	More than 75%	Useless	Never

[This page has been intentionally left blank]

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Luís Silva

A Pattern-Based Approach to Scaffold the IT Infrastructure Design Process

2017