UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

VINÍCIUS TAVARES GUIMARÃES

A Reuse-based Approach to Promote the Adoption of InfoVis Techniques for Network and Service Management Tasks

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dr. Lisandro Z. Granville
Coadvisor: Profa. Dra. Liane Margarida
Rockenbach Tarouco

Porto Alegre
October 2016

CIP — CATALOGING-IN-PUBLICATION

Guimarães, Vinícius Tavares

A Reuse-based Approach to Promote the Adoption of InfoVis Techniques for Network and Service Management Tasks / Vinícius Tavares Guimarães. – Porto Alegre: PPGC da UFRGS, 2016.

134 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Advisor: Lisandro Z. Granville; Coadvisor: Liane Margarida Rockenbach Tarouco.

1. Network Management. 2. Service Management. 3. Information Visualization. 4. Productivity. 5. Cost Reduction. 6. Software Reuse. I. Granville, Lisandro Z.. II. Tarouco, Liane Margarida Rockenbach. III. Título.

"If money is your hope for independence you will never have it. The only real security that a man will have in this world is a reserve of knowledge, experience, and ability."

— Henry Ford

AGRADECIMENTOS

À minha família. Eles foram, são e sempre serão fundamentais em qualquer etapa da minha vida.

Ao meu orientador, professor Lisandro Zambenedetti Granville, e a minha co-orientadora, professora Liane Margarida Rockenbach Tarouco, por terem oportunizado meu ingresso no Grupo de Redes da UFRGS, bem como pela dedicação no processo de orientação. À professora Carla Maria Dal Sasso Freitas pela atenção, dedicação e disponibilidade nos trabalhos que foram desenvolvidos em colaboração.

Aos amigos Glréderson Lessa dos Santos, Guilherme da Cunha Rodrigues, Oscar Mauricio Caicedo Rendon e Roberto Irajá Tavares da Costa Filho pelas inúmeras contribuições técnicas e pessoais.

Aos colegas da sala 221 pelo agradável convívio diário: Marcelo Caggiani Luizelli, Weverton Luis da Costa Cordeiro, Rafael Hansen da Silva, Gustavo Miotto, Leonardo Richter Bays, Rafael Pereira Esteves, Jeferson Campos Nobre, Carlos Raniery Paula dos Santos, Flávio Roberto Santos e Márcio Barbosa de Carvalho.

Aos demais professores e colegas do Grupo de Redes da UFRGS.

Ao IFSul por ter me concedido a licença de afastamento para qualificação. Tal período foi fundamental para o desenvolvimento do doutorado.

ABSTRACT

Throughout the years, several tools have being used by network administrators to accomplish the management tasks (*e.g.*, management protocols and network monitoring solutions). Among such tools, this thesis focuses on Information Visualization one (a.k.a InfoVis). Mainly, it is understood that the ultimate goal of these management tools is to decrease the complexity and, consequently, optimize the everyday work of administrators. Thus, they can increase their productivity, which leads to the cost reduction. Based on this assumption, this thesis aims at investigating how to promote the adoption of InfoVis techniques by network administrators, focusing on enhancing productivity and lowering costs. The key insight is that, in most cases, network administrators are unskilled on InfoVis. Therefore, the choice to adopt visualizations can require an immersion into the unknown that can be too risky regarding productivity and cost. In essence, this thesis argues that the employment of InfoVis techniques by administrators can be very laborious by spending a significant amount of effort that decreases their productivity and, consequently, increases the management costs. To overcome this issue, an approach to promote the adoption of InfoVis techniques by encouraging their reuse is introduced. It is argued that the concepts and principles of software reuse proposed and standardized in the software engineering field can be adapted and employed once the building up of visualizations (*i.e.*, the design and development) can be defined primarily as a software development task. So, the evaluation of the proposal introduced in this thesis employs the Common Software Measurement International Consortium (COSMIC) Functional Size Measurement (FSM) method that measures software sizing through Function Points (FP). From this method, it was possible estimating effort and, consequently, productivity and costs. Results show the feasibility and effectiveness of the proposed approach (in terms of productivity and cost) as well as some indirect benefits that the systematic reuse can provide in the adoption of InfoVis techniques to assist in the management tasks.
Keywords: Network Management. Service Management. Information Visualization. Productivity. Cost Reduction. Software Reuse.

Uma Abordagem Baseada em Reuso para Promover a Adoção de Técnicas de Visualização de Informações no Contexto de Gerenciamento de Redes e Serviços

RESUMO

Ao longo dos anos, diferentes ferramentas vem sendo utilizadas pelos administradores de rede para realizar as tarefas de gerenciamento (por exemplo, protocolos de gerência e soluções de monitoramento de rede). Dentre tais ferramentas, a presente tese foca em Visualização de Informações (ou simplesmente InfoVis). Essencialmente, entende-se que o objetivo final dessas ferramentas de gestão é diminuir a complexidade e, consequentemente, otimizar o trabalho diário dos administradores. Assim, eles podem melhorar sua produtividade, o que incide diretamente na redução de custos. Com base nesse pressuposto, esta tese tem como objetivo investigar como promover a adoção de técnicas InfoVis pelos administradores de rede, com foco em melhorar produtividade e diminuir custos. A percepção chave é que, na maioria dos casos, os administradores de rede não são habilitados no domínio InfoVis. Desse modo, a escolha por adotar técnica InfoVis requer a imersão em campo desconhecido, podendo gerar, assim, um risco elevado nos indicadores de produtividade e custos. Em essência, essa tese argumenta que o emprego de técnicas InfoVis pelos administradores pode ser muito trabalhosa, despendendo um montante muito significativo de tempo, o que leva a diminuir produtividade e, consequentemente, eleva os custos de gerenciamento. Focando essa questão, é apresentada uma proposta para promover adoção de técnicas InfoVis, pelo encorajamento do reuso. Argumenta-se que os conceitos e princípios de reuso propostos e padronizados pelo campo da engenharia de software podem ser adaptados e empregados, uma vez que a construção de visualizações (ou seja, o projeto e desenvolvimento) é, primariamente, uma tarefa de desenvolvimento de software. Assim, a avaliação da proposta apresentada nesta tese utiliza o método Common Software Measurement International Consortium (COSMIC) Functional Size Measurement (FSM), o qual permite estimar o dimensionamento de software através de pontos por função. A partir deste método, torna-se então possível a estimativa de esforço e, consequentemente, produtividade e custos. Os resultados mostram a viabilidade e eficácia da abordagem proposta (em termos de produtividade e custos), bem como os benefícios indiretos que o reuso sistemático pode fornecer quando da adoção de visualizações para auxílio nas tarefas de gerenciamento de redes.

Palavras-chave: Gerência de Redes, Gerência de Serviços, Visualização de Informações, Produtividade, Redução de Custos, Reuso de Software..

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

AJAX	Asynchronous JavaScript and XML
ANSI	American National Standards Institute
API	Application Program Interface
BFC	Base Functional Component
BGP	Border Gateway Protocol
CapEx	Capital Expenditure
CBSE	Component-Based Software Engineering
CBD	Component-Based Development
CFP	Cosmic Function Point
COSMIC	Common Software Metrics International Consortium
CRUD	Create/Read/Update/Delete
CSS	Cascading Style Sheets
D3	Data-Driven Documents
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DBMS	Database Management Systems
EDI	Electronic Data Interchange
EMANICS	European Network of Excellence for the Management of Internet Technologies and Complex Services
FCC	Federal Communication Commission
FP	Function Points
FPA	Function Point Analysis
FiSMA	Finnish Software Metrics Association
FSM	Functional Size Measurement
FUR	Functional User Requirement

GUI	Graphical User Interface
HCI	Human-Computer Interaction
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IFPUG	International Function Points Users Group
InfoVis	Information Visualization
IT	Information Technology
ITU	International Telecommunication Union
IRTF	Internet Research Task Force
ISO	International Organization for Standardization
ISP	Internet Service Provider
JTC	Joint Technical Committee
JSON	JavaScript Object Notation
KOS	Simple Knowledge Organization System
LAN	Local Area Network
MANET	Mobile Ad-hoc Networks
MBA	Measuring Broadband America
MIB	Management Information Base
MVC	Model-View-Controller
NESMA	Netherlands Software Metrics Association
NFS	Network File System
NMRG	Network Management Research Group
OASIS	Organization for the Advancement of Structured Information Standards
Ofcom	Office of Communication

OpEx	Operational Expenditure
OSI	Open Systems Interconnection
OOAD	Object-Oriented Analysis and Design
QoS	Quality of Service
REST	Representational State Transfer
RFC	Request for Comments
ROA	Resource-Oriented Architecture
RSL	Reusable Software Library
RTT	Round-Trip Time
SC 7	Subcommittee 7
SESC	Software Engineering Standards Committee
SDN	Software-Defined Networking
SKOS	Simple Knowledge Organization System
SLA	Service Level Agreement
SLS	Service Level Specification
SLR	Systematic Literature Review
SMO	System Management Overview
SNMP	Simple Network Management Protocol
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TCO	Total Cost of Ownership
VoIP	Voice over IP
VPN	Virtual Private Network
UK	United Kingdom

URI	Uniform Resource Identifier
US	United States
XML	eXtensible Markup Language
W3C	World Wide Web Consortium
WAN	Wide Area Network
WSN	Wireless Sensor Networks
WSDL	Web Services Description Language
WWW	World Wide Web

LIST OF TABLES

CONTENTS

1 INTRODUCTION

This introduction starts using the Formula 1 world championship as a metaphor to emphasize the importance of network and service management. The underlying assumption is that a great driver needs a competitive car to be a world champion. So, what should be done to get a competitive race car? Obviously, this question has some straightforward answers. For example, building up a competitive race car requires, minimally, a well-developed design, a qualified engineers team, and high-quality gears (such as mechanical and electrical equipment). Nevertheless, these requirements are not enough to ensure the car performance throughout the season. If this assumption were true, once a race car has been built and tested, no more improvements would be needed. Instead, Formula 1 teams normally have a huge budget to invest in computational capabilities such as monitoring performance by telemetry, and software tools for prediction and trend analysis. These are typical examples of resources that help teams to maintain the race car efficient and competitive.

Based on that metaphor, it is possible to outline a similar context for networked systems. Indeed, a significant portion of the performance of a networked system is obtained by proper design, an expert team of engineers and network administrators, infrastructure, and suitable hardware and software. However, this is only an important phase in the process, as well as the race car designing and building. Network administrators and engineers teams should be enabled, for example, to monitor, measure, and analyze the networked ecosystem to keep it healthy as long as possible. Thus, the fundamentals and practices defined by the network and service management discipline are mandatory.

Efforts on network management standardization were started by the Open Systems Interconnection (OSI), within the International Organization for Standardization (ISO) in conjunction with the telecommunication standardization sector of the International Telecommunications Union (ITU-T). In this context, this thesis highlights the widely known OSI Management Framework (ISO/IEC, 1989) and OSI Systems Management Overview (SMO) (ISO/IEC, 1998), which divided management functions into five functional areas. These areas are commonly denoted by the term "FCAPS", an acronym for **F**ault, **C**onfiguration, **A**ccounting, **P**erformance, and **S**ecurity management.

In general, the standardization of the OSI reference model is the basis for other network management definitions. Here, two definitions of network and service management are introduced. According to Pras (PRAS, 1995), network management is the act of initializing, monitoring, and modifying the operation of the primary network functions, where primary network functions are those functions that directly support user requirements. Clemm (CLEMM, 2006) defines network management as the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems.

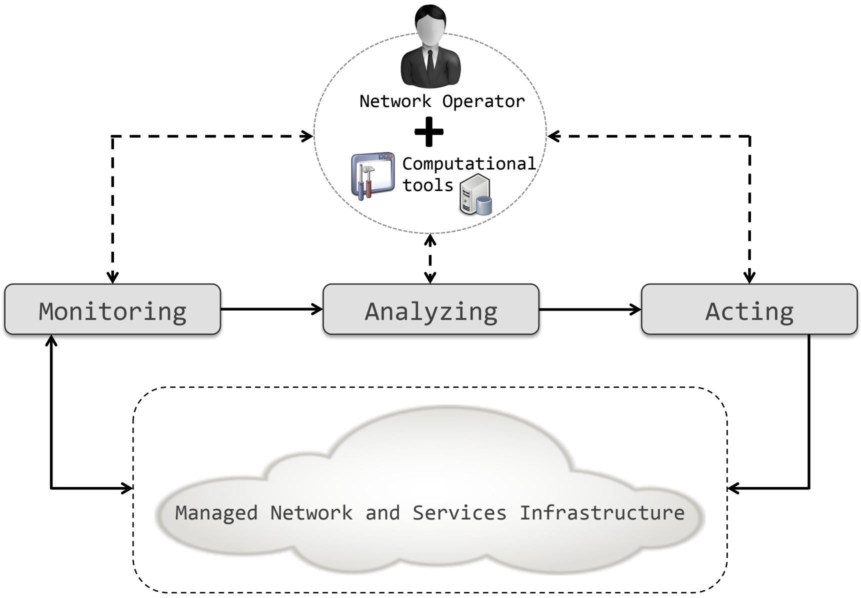
Nowadays, services and applications that comprise business processes are supported by networked systems. Typical examples are Electronic Data Interchange (EDI) systems. Enterprises may incur extreme losses (*e.g.*, billing, productivity, image degradation) because of interruption or degradation on their EDI system. From this example, it is possible to understand how sensitive business processes are in the event of disruption or instability in the network and service infrastructure. Thus, the network and service management is no longer a relevant issue only for network administrators and engineers teams. The concerns about an efficient network and service management became part of the business goals since it is one of the pillars to ensure competitive advantages in the market.

Based on such context, computational tools are essential to perform network and service management tasks. Basically, they are means to support the management workflow as a whole. In a general way, the management workflow is based on three main axes: (*i*) monitoring, (*ii*) analyzing, and (*iii*) acting. The monitoring process is characterized by obtaining raw data from the managed environment (*e.g.*, configuration and performance data). In turn, the analysis process is based on the interpretation and reasoning over the collected data. Finally, in the acting phase, actions (*e.g.*, reaction to a failure event, reconfiguration, optimization) are performed. Figure 1.1 depicts a generic network and service management workflow based on these three main axes.

Over the years, several computational tools have been developed to support the management activities. The Simple Network Management Protocol (SNMP) (HARRINGTON; PRESUHN; WIJNEN, 2002), frequently referred to as the *de facto* management protocol for TCP/IP networks, is a typical example of such tools. This thesis, however, focuses on another important tool that assists network administrators in daily tasks: Information Visualization, from now on called simply as InfoVis.

As stated by Shneiderman (SHNEIDERMAN, 1996): *"a picture is often cited to be worth a thousand words and, for some (but not all) tasks, it is clear that a visual presentation - such as a map or photograph - is dramatically easier to use than is a textual description or a spoken report*". Actually, visual representations are very efficient even

Figure 1.1: A generic network and service management workflow based on three main axes: monitoring, analyzing, and acting. Two roles are depicted at the top: the network administrator and the computational tools



Source: by author (2016).

in ordinary daily activities. The sign boards on pavements, streets, and subways are the most simple examples. Another good illustration is the maps that explain the railway lines of subways, which are likely to be understood by people not speaking the country's language.

The field of InfoVis forms a whole research area in computer science. This discipline was formally recognized in the late 80's, coincidentally close to the time when the network and service management community started to grow. Card *et al.* (CARD; MACKINLAY; SHNEIDERMAN, 1999) define InfoVis as the use of computer-supported, interactive, visual representations of **abstract** data to amplify cognition. The word "abstract" is highlighted in bold in the previous sentence because it plays a significant role in the definition of InfoVis. The visualization field has been subdivided into two main subfields: scientific visualization and information visualization. While scientific visualization deals with scientific data, InfoVis deals with abstract data.

Most of the time one distinguishes the areas based on the spatial domain of scientific visualization applications as opposed to a non-spatial (*e.g.*, a set of elements) domain of InfoVis applications. According to Tory and Möller (TORY; MöLLER, 2004):

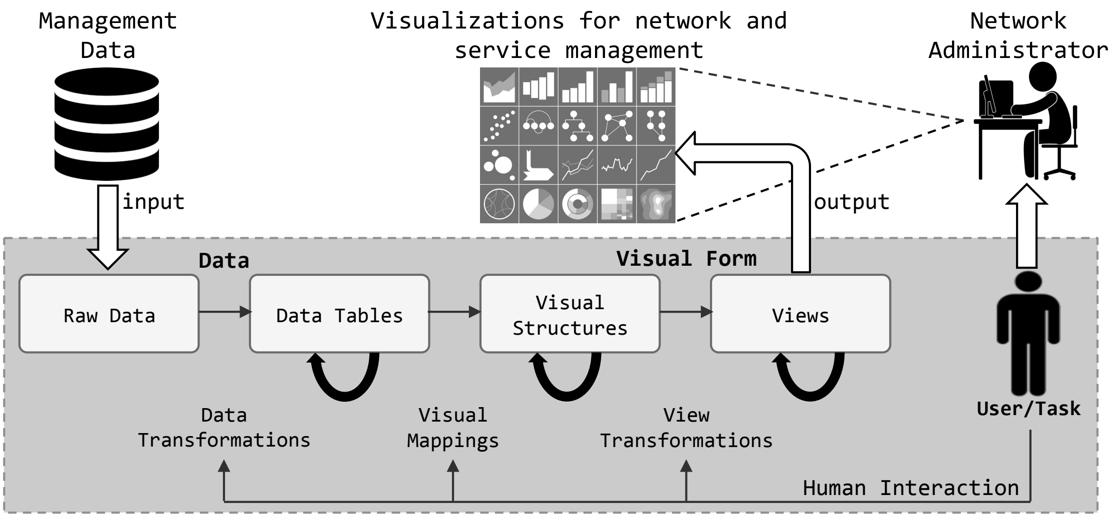
- Scientific visualization is typically categorized by the dimensionality of the data values (number of independent variables), and whether the data is scalar, vector, tensor, or multivariate (having more than one dependent variable).

- InfoVis can be similarly organized by data type. Typical categories are multi-dimensional databases (often containing more than three dimensions), text, graphs, and trees.

Based on the above definitions, it is clear that the nature of network and service management data are aligned with InfoVis assumptions. Datasets such as texts (*e.g.*, logs or configuration settings) and graphs (*e.g.*, logical connections among IP addresses) are typical examples of data handled in network and service management tasks; they are clearly defined over a non-spatial domain and can be classified as abstract data.

Card *et al.* (CARD; MACKINLAY; SHNEIDERMAN, 1999) defined a reference model for InfoVis that is described as follows (see the gray area of Figure 1.2). At the first step, raw data (*i.e.*, collected or synthesized) are represented as data tables to generate the desired information. Next, data tables (*i.e.*, resulting from transformations on raw data) are manipulated and then transformed into one or more visual representations. At the last step, the end-user manipulates and interacts with the visual representation in one or more views. Figure 1.2 places the management data and the network administrator together with the InfoVis reference model. The management data are the input for the reference model whereas the outputs of the model are visualizations that administrators use to assist in daily tasks.

Figure 1.2: The management data and the network administrator together with the InfoVis reference model (the gray area) (CARD; MACKINLAY; SHNEIDERMAN, 1999)



Source: by author (2016).

1.1 Motivation & Objective

Network administrators have several primary tasks. A non-exhaustive list of these tasks is: (*i*) installing and supporting networked systems (*e.g.*, WANs and network segments), (*ii*) monitor the network to ensure security and availability, (*iii*) troubleshooting and repairing of network devices, (*iv*) assigning routing protocols and routing table configuration, (*v*) evaluating and tuning the network performance; and (*vi*) ensuring Quality of Service (QoS) and the Service Level Agreements (SLAs). In this sense, as previously presented, there are many tools that assist network administrators to perform management tasks, such as management protocols (*e.g.*, SNMP), network monitoring solutions (*e.g.*, Zabbix), packet and protocol analyzers (*e.g.*, Wireshark), and configuration assistants (*e.g.*, Cisco Configuration Assistant). In short, the ultimate goal of such tools is to provide means to decrease the complexity and, consequently, to optimize the daily work of administrators. Thus, network administrators can increase their productivity, benefiting company's profits by reducing costs. Based on that, this thesis aims at investigating how to promote the adoption of InfoVis techniques by network administrators, focusing on enhancing productivity and lowering costs.

The primary motivation to conduct this investigation is based on the observation that, although visualizations can be a powerful supporting tool for network administrators, its adoption by administrators (as a day-to-day tool) is not a reality yet. The key insight is that, in most cases, network administrators are unskilled on InfoVis. Thus, the choice to adopt visualizations can require an immersion into the unknown that can be too risky in terms of productivity and cost. In essence, this thesis argues that the employment of InfoVis techniques by administrators can be very laborious by spending a significant amount of effort that decreases their productivity and, consequently, increases the management costs. To better explain the arguments and motivation, this section checks two alternatives regarding the adoption of visualizations by network administrators as follows (see Figure 1.3):

(a) The first alternative is placed in the network and service management domain. In this case, the network administrators adopt visualizations provided by traditional management tools. In general, such tools are proprietary or open-source. For the proprietary ones (*e.g.*, Solar-Winds, Hyperglance, Manage-Engine OpManager, and IBM Tivoli), two major drawbacks are identified: (*i*) it increases the Capital Expenditure (CapEx); and (*ii*) it affects the Total Cost of Ownership (TCO). Regarding open-source tools (*e.g.*, Zabbix, Nagios, MRTG, CACTI, and Elasticsearch-Logstash-Kibana - a.k.a. ELK), it does not generate direct financial expenses. However, it can bring hidden costs that may increase the Operational Expenditures (OpEx). For example, the maintenance costs and the costs associated with the spending time to deploy and setup an open-source solution. Although this alternative can be feasible, in most cases, it does not promote the adoption of visualizations by network administrators because they can lose focus on their core tasks. Another important weakness of this alternative is related to the fact that the visualizations available by those tools are “as is”. Therefore, administrators are not able, for example, to explore ground-breaking visualizations that can better fit with their specific management needs.

(b) The second alternative is in the InfoVis domain. In essence, this thesis envisages two possibilities for this alternative. In the first one, the network administrator seeks for end-user tools, *i.e.*, tools that generate visualizations and do not require any coding. Some examples of such tools are Tableau Public, Datawrapper, Raw, Excel, and R. For network administrators, the first pitfall of this alternative could be the need of working with tools and technologies that are outside of their scope of skills, which can laborious and time-consuming. Another aspect refers to the general appeal of such tools. Actually, it is a strength that is not pretty helpful for the management domain. For instance, Tableau Public is an excellent tool for Storytelling (KOSARA; MACKINLAY, 2013) about wide-range topics like global health, policy, and sports but it is far from network and service management issues. Thus, this possibility does not promote the adoption of visualizations by administrators. The second possibility is the use of developer tools (*e.g.*, D3, Google Charts, FusionCharts, and JqPlot), *i.e.*, the network administrator builds the visualization in a clean slate approach. In essence, he/she must have skills to choose an adequate visualization (including interaction features, color effects, and so on), to select the development framework, to seek for supporting material (*e.g.*, Web tutorials) for programming, deploying and maintaining. Besides being time-consuming, building up visualizations is out of the scope of network administrators’ responsibilities. Another path for this possibility is outsourcing. In this case, the administrator consults or hires InfoVis designers and developers to build up custom visualizations. Although it can be less expensive concerning effort, it can strongly affect CapEx and OpEx.

Reviewing the literature, the employment of InfoVis techniques for network and service management is not a recent research topic (PRAS et al., 2007) and several in-

Figure 1.3: The alternatives regarding the adoption of visualizations

Source: by author (2016).

vestigations were carried out throughout the years (GUIMARÃES et al., 2016). Still in the 90s, for example, Becker *et al.* (BECKER; EICK; WILKS, 1995) presented the first relevant work in the field. In that work, they describe in details the visualization techniques (*e.g.*, link-node maps and matrix display) and interactive tools (*e.g.*, manipulating line length between links, line thickness, symbol size and color, animation speed, zooming, and brushing) used to aid in the management of the AT&T's network. Other proposals address InfoVis, for example: (*i*) to assist in the understanding of the use of the SNMP (GUIMARÃES et al., 2014)(BARBOSA; GRANVILLE, 2010)(SALVADOR; GRANVILLE, 2008), (*ii*) to help in the management of Wireless Sensor Networks (WSN) (SATO K.; SHIMADA, 2015)(KASTURE; RAUT; THOOL, 2014)(KARAPISTOLI; ECONOMIDES, 2012) and Mobile Ad-hoc Networks (MANET) (TSUTSUI et al., 2014)(KOYAMA; KAMAKURA; BAROLLI, 2009), (*iii*) to present meaningful information regarding the behavior of the Domain Name System (DNS) (LAI et al., 2015), (*iv*) to allow network administrators to easily understand and control OpenFlow-based Software-Defined Networking (SDN) (ISOLANI et al., 2015); and (*v*) to comprehend the Internet routing data based on information from the Border Gateway Protocol (BGP) (PAPADOPOULOS; THEODORIDIS; TZOVARAS, 2013)(BIERSACK et al., 2012)(PAPADOPOULOS; MOUSTAKAS; TZOVARAS, 2012).

Another important aspect regarding investigations on InfoVis for network and service management refers to the fact that most of the investigations are focused on security management (SHIRAVI; SHIRAVI; GHORBANI, 2012). In such a context, InfoVis has been investigated, for instance: (*i*) to facilitate browsing on network log files to investigate suspicious activities (STANGE et al., 2014), (*ii*) to support security analysts with analytic reasoning and decision making in response to ongoing web server attacks (ALSALEH et al., 2013), (*iii*) to enable operators to grasp visually an overview of alert circumstances based on large-scale darknet monitoring (INOUE et al., 2012), (*iv*) to help in security situational awareness (LAKKARAJU; YURCIK; LEE, 2004)(YIN et al., 2004), (*v*) to detect port-based security events (MCPHERSON et al., 2004); and (*vi*) to assist in intrusion detection (CORCHADO; HERRERO, 2011)(ERBACHER; WALKER; FRINCKE, 2002).

Although the investigations mentioned above have achieved interesting results and consolidated an important background in the field, to the best of our knowledge, no work aims at promoting the employment of InfoVis techniques by network administrators, focusing on enhancing productivity and lowering costs. Thus, with this end, this thesis introduces an approach to promote the employment of InfoVis techniques by encouraging their reuse. This thesis advocates that a reuse-based approach is promising in four-fold: (*i*) it takes advantage of already developed visualizations, (*ii*) it provides an easy path to build up, share and reuse groundbreaking visualizations (*iii*) it makes possible to create background knowledge and consensus building through systematic reuse, and (*iv*) it has as ultimate goal the improving of productivity and the decreasing of costs. In this sense, it is argued that the concepts and principles of software reuse proposed and standardized in the software engineering field can be employed once the building up of visualizations (*i.e.*, the design and development) can be defined primarily as a software development task.

1.2 Hypothesis & Research Questions

Based on the context previously presented, this thesis presents the following hypothesis.

***Hypothesis:* a reuse-based approach can be effective to decrease cost and improve the productivity of network administrators when adopting InfoVis techniques to assist in everyday tasks.**

In order to guide the investigations conducted in this thesis, the following research questions (RQ) associated with the hypothesis are defined and presented.

RQ I. *How a reuse-based approach can decreasing costs and increase the productivity of administrators when adopting InfoVis techniques?*

RQ II. *What methods/mechanisms could be used in the design and development of a reuse-based approach?*

RQ III. *What is the performance, in terms of cost and productivity, of a reuse-based approach?*

The methodology employed to show the feasibility of the proposed approach is based on an evaluation that comprises two network management scenarios. Such scenarios are explored with the aim to demonstrate that the proposed approach can be effective to diminish costs and improving productivity. The first evaluation focuses on a traditional network management scenario by using two open-access datasets of Internet performance measurements provided by (*i*) the Federal Communications Commission (FCC) - the national regulator in the United States - in a campaign named as measuring broadband America; and (*ii*) the Office of Communications (Ofcom), the national regulator in the United Kingdom. The second evaluation explores a modern scenario of network management, specifically, an SDN environment. Based on the fact that the building up of visualizations is, by its nature, a software development activity, the evaluation employs the Common Software Measurement International Consortium (COSMIC) (ISO/IEC, 2011) Functional Size Measurement (FSM) method that measures software sizing through Function Points (FP). From this method, it was possible estimating effort and, consequently, productivity and costs.

1.3 Contributions

In addition to verifying the hypothesis, this thesis introduces other contributions that can be divided into two categories: conceptual and specific. Conceptual contributions are those identified during the investigation of the literature and the experiences gathered in this process. In contrast, specific contributions are associated with individual solutions developed in this thesis. Such contributions are listed below.

- Conceptual Contributions:
  - A comprehensive survey on InfoVis for network and service management that was carried out through a Systematic Literature Review (SLR).
  - Examination of the principles, concepts, and guidelines regarding software reuse field which may be tailored and applied to the design and development of reuse-based solutions.
- Specific Contributions:
  - A reuse-based approach that aims at promoting the reuse of software components to the launching of visualizations for network management, focusing on improving productivity and cost reduction.

- The employment of a software sizing method to quantify and demonstrate that building up from scratch is time-consuming and expensive, and then to ratify that the proposed approach can be an effective path to decrease cost and improve productivity.

1.4 Organization

The remainder of this thesis is organized as follows.

In Chapter 2 is outlined a comprehensive survey of investigations regarding InfoVis for network and service management through an SLR. Moreover, the most important background concepts and studies related to this thesis are reviewed.

In Chapter 3, the reuse-based approach proposed in this thesis is detailed. First, the stakeholders are described together with the conceptual architecture. Afterward, an overview of the proposed solution is introduced, explaining the central concepts, the asset classification scheme, the asset storage and retrieval mechanism, and the asset usage recording. Next, this chapter presents each tier of the architecture, and the software elements and its interactions. Finally, the prototype implementation is outlined.

In Chapter 4, an evaluation of the proposed approach is presented. To achieve that, two assessment scenario were developed. Both scenarios aim at demonstrating how reuse can improve productivity and decrease cost when network administrators adopt visualizations to assist on everyday management tasks. The first assessment scenario uses Internet performance measurements datasets whereas the second one explores an SDN-based environment.

In Chapter 5, some final remarks and conclusions are introduced. Moreover, answers to the fundamental research questions are discussed. Finally, opportunities to develop future work are identified and detailed.

2 BACKGROUND & LITERATURE REVIEW

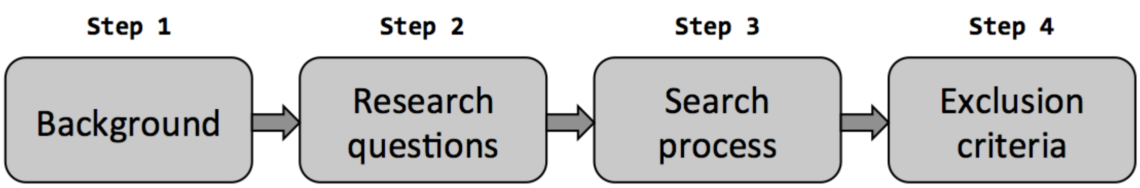
The goal of this chapter is two-fold. First, it introduces a comprehensive SLR about the employment of InfoVis for network and service management. Section 2.1 starts presenting the review protocol followed by the background (Section 2.1.1), research questions (Section 2.1.2), search process (Section 2.1.3), and exclusion criteria (Section 2.1.4). Afterward, Section 2.1.5 presents the classification of the surveyed works according to a network and service management taxonomy and an InfoVis taxonomy. In short, the SLR constitutes one of the pillars of this thesis because it surveys the literature on InfoVis and network and service management in a large (*i.e.*, the number of articles and papers) and comprehensive way (*i.e.*, several topics on network and service management).

In a second moment, Section 2.2 introduces the background regarding software reuse that forms the base of the reuse-based approach proposed in this thesis. First, it presents general concepts about software reuse as part of the software engineering field. Next, Section 2.2.1 addresses the international standard ISO/IEC 12207 / IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes. The Component-Based Development (CBD) paradigm and the architectural style with the focus on reuse are discussed, respectively, in Section 2.2.2 and 2.2.3. Finally, Section 2.2.4 presents a brief explanation on Semantic Web and how it can help to promote the reuse.

2.1 SLR on InfoVis for Network and Service Management

In this thesis, the concepts of SLR proposed by Keele (KEELE, 2007) are adapted as a means to structure and organize the research concerning InfoVis for network and service management. As a first step, an *ad-hoc* review protocol is defined. Figure 2.1 depicts the review protocol, and the following subsections describe each step of it.

Figure 2.1: Review protocol of the SLR



Source: by author (2016).

2.1.1 Background

In the context of an SLR, the background refers to the rationale of the survey. In short, the SLR presented in this section introduces a thorough survey of the employment of InfoVis for network and service management. In this sense, this SLR has provided the basis to build up this thesis, since it provides a in-depth understanding of how the research on InfoVis for network and service management was conducted until now.

2.1.2 Research questions

The research questions are those that this survey intends to answer. The questions for this SLR are:

RQ1: What is the big picture (from a historical overview of the state of the art) regarding the literature about InfoVis for network and service management?

RQ2: What are the most explored topics on network and service management in which InfoVis was investigated to help network administrators in daily tasks?

RQ3: What are the most employed InfoVis techniques and tasks/interactions for network and service management?

RQ4: What related insights are revealed by the proposed classification? For example, what are the most widely used InfoVis techniques for a given network and service management topic?

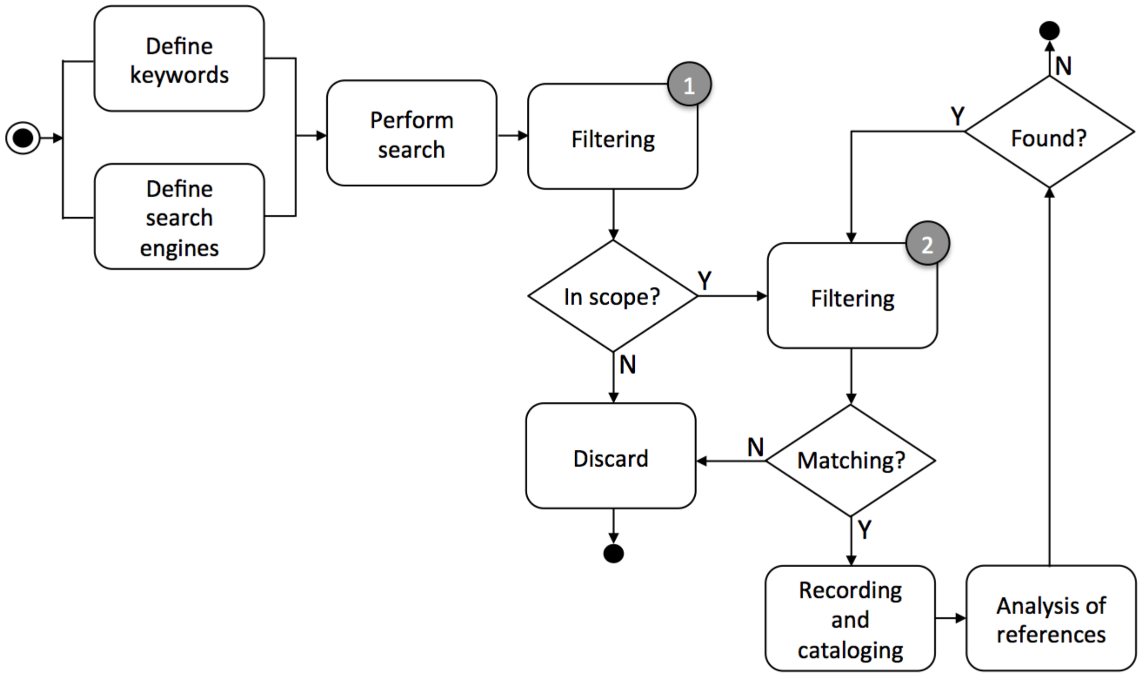
2.1.3 Search process

First, as shown in Figure 2.2, the keywords and the academic search engines are defined as follows:

1. **Keywords:** “visualization AND network management”, “network AND service AND visualization”, “network visualization”, “network AND visual”, and “visualization AND security”. The use of the “security” keyword was based on a previous analysis, in which a large number of works was identified in this network and service management subtopic (this issue will be addressed in the following sections).

2. **Academic search engines:** Goggle Scholar, Microsoft Academic Search, Scopus, IEEExplore Digital Library, and ACM Digital Library.

Figure 2.2: The flowchart depicts the search process defined as the Step 3 in Figure 2.1



Source: by author (2016).

After defining keywords and academic search engines, the search itself was performed. Starting now, to better explain each step after "Perform search" (see Figure 2.2), it is used an example where the Google Scholar is the search engine and "visualization AND network management" is the query. Basically, the first step of filtering (after "Perform search" in Figure 2.2) refers to the analysis of the search results (*e.g.*, title and brief description) to identify articles and papers that are in or out of scope. By using the example, Google Scholar returned around 578,000 results. The results were sorted by relevance and analyzed the first three hundred of them. The analysis of the first three hundred was defined because, after the first hundred, the results started to be unrelated to the research goal. Among the first ten results of the exemplified query, the work of Corchado and Herrero (CORCHADO; HERRERO, 2011), Le Grand and Soto (GRAND; SOTO, 2000), and Itoh (ITOH et al., 2006) are examples of articles/papers that were selected for the second step of filtering. On the other hand, also among the first ten results, the work of Provan and Kenis (PROVAN; KENIS, 2008) was directly discarded because it was identified as out of scope.

Next, as the second step of filtering (see Figure 2.2), the abstract and keywords of each paper selected in the first filtering were analyzed. At this point, if the article/paper somehow matches with the research goal, it is recorded and cataloged. Otherwise, the article/paper is discarded. For each recorded and cataloged article/paper, its references were analyzed in order to seek for other relevant works in the field. In this stage, the title and the publication venue of each reference were verified. If the title and the publication venue of the reference were promising, the second step of filtering was performed over the article/paper selected from the reference. In essence, it was an iterative process, *i.e.*, for each new article/paper its references were also analyzed.

Taking into account the example mentioned above, Corchado and Herrero (CORCHADO; HERRERO, 2011) and Itoh *et al.* (ITOH et al., 2006) are examples of articles/papers that match with the research goal. Thus, they are stored and cataloged, and their references are verified. In the analysis of the references from the work of Corchado and Herrero (CORCHADO; HERRERO, 2011), for example, the work of Koike *et al.* (KOIKE; OHNO; KOIZUMI, 2005) was found. On the other hand, the work of Le Grand and Soto (GRAND; SOTO, 2000) is an example of work that was discarded after the analysis of the abstract and keywords (*i.e.*, the second step of filtering). Once finished the search process, a set of 403 articles and papers remained selected. These publications are from a time interval ranging from 1985 to 2015.

2.1.4 Exclusion criteria

After recording and cataloging those 403 articles and papers, a deeper analysis of each one was performed where the introduction, proposal overview, results, and conclusions were verified. This step aimed to identify articles and papers that match or not the five exclusion criteria defined for this research. The five exclusion criteria are defined as follows:

1. **Gray area for network and service management:** articles and papers that are not clearly in the network and service management scope, *i.e.*, they could not be a consensus in the community. For instance, Veras *et al.* (VERAS; THORPE; COLLINS, 2012) introduced an investigation into the semantic patterns underlying user choice in passwords. It is a very interesting work but falls under this criterion.

2. **Gray area for InfoVis:** articles and papers that do not present the employed InfoVis techniques clearly. For instance, the work proposed by Dinh-Duc *et al.* (DINH-DUC; DANG-HA; LAM, 2012).

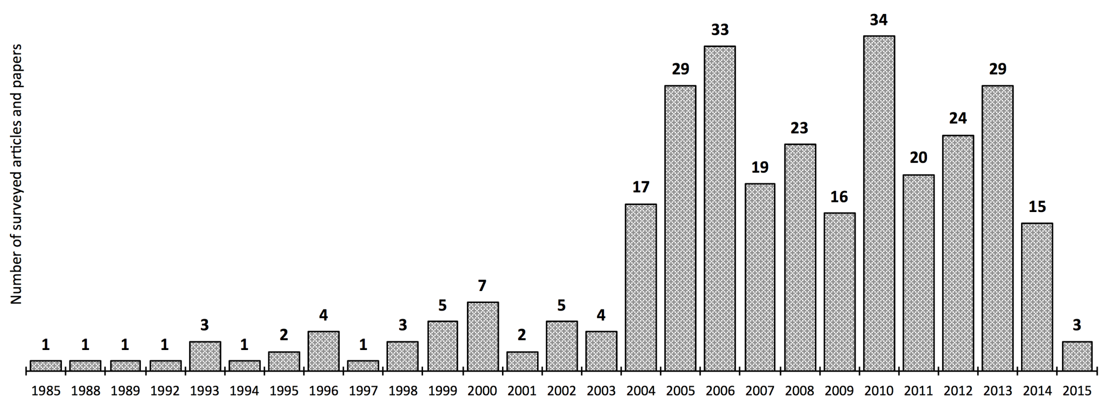
3. **Surveys/Evaluation**: articles and papers that introduce a research addressing some topic in the field. For instance, Goodall (GOODALL, 2009) introduced an interesting comparative evaluation of a visualization application and a traditional interface for analyzing network packet captures. Shivari *et al.* (SHIRAVI; SHIRAVI; GHORBANI, 2012) introduced a valuable survey of Visualization Systems for Network Security.

4. **Same as or related to**: articles and papers that were published in more than one venue (*e.g.*, journal, conference, etc.) or articles and papers that are improvements/variations of previous work. For instance, the works presented in (KOYAMA et al., 2012), (KOYAMA; KAMAKURA; BAROLLI, 2009), (SATO; KOYAMA; BAROLLI, 2011), and (INOUE; TAKAHASHI; KOYAMA, 2013). In these cases, only the most recent work is kept (*i.e.*, in the example, the work presented by Inoue *et al.* (INOUE; TAKAHASHI; KOYAMA, 2013)).

5. **Withdraw**: articles and papers that, although having passed in the first and second step of filtering, after a deeper analysis, they were classified as out of scope for this SLR. For example, the work of Fang *et al.* (FANG; MILLER; KUPSCH, 2012).

Based on these five criteria, 100 articles and papers were left out of the survey. Thus, 303 articles and papers were selected from the originally 403 recorded and cataloged during the searching steps. These 303 articles were used for the classification described in Section 2.1.5. Figure 2.3 shows the number of surveyed articles and papers per publication year.

Figure 2.3: The number of surveyed articles and papers per year of publication



Source: by author (2016).

2.1.5 Classification

This section shows the classification of the 303 articles and papers according to both a network and service management taxonomy (Section 2.1.5.1) and an InfoVis taxonomy (Section 2.1.5.2). However, before to introduce the taxonomies and the classification itself, this section presents an interesting outcome regarding the surveyed articles and papers which was observed during the search process. In essence, the papers and articles are divided into two well-defined segments as follows: (*i*) security field - works focused on InfoVis techniques for security management, and (*i*) other fields - works that address other topics of network and service management instead of security management (*e.g.*, analysis of routing data (BURCH; CHESWICK, 1999)(TEOH et al., 2002)(COLITTI et al., 2005) and configuration management (YANG; EDWARDS; HASLEM, 2010)).

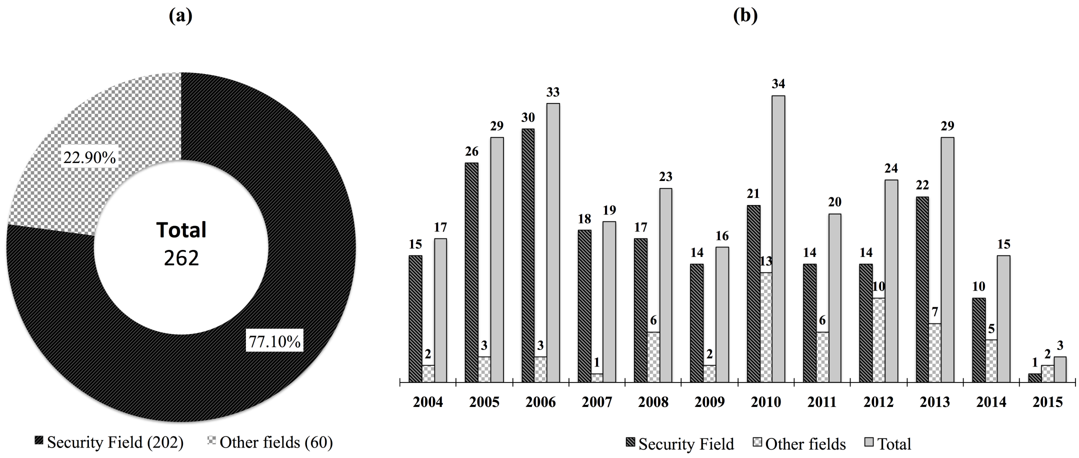
The aforementioned division becomes more evident on publications from the 2000s. Specifically, after 2004, the number of publications addressing security issues has an expressive increase, since at that year the first edition of VizSec forum has occurred. VizSec focuses on visualization and data mining for computer security, and it started as a workshop. Over the years, the event got bigger and, since 2010, VizSec began to publish proceedings with ACM International Conference Proceeding Series (ACM ICPS).

VizSec has a significant contribution to the field. For instance, this survey found a total of 33 articles and papers in 2006, where 16 were published in VizSec (*i.e.*, 48.5%). However, this behavior is not only because of VizSec. Articles and papers published in other venues were also focused on security issues. In 2007, for example, only one work did not address security management (*i.e.*, 5.26%). The period between 2009 to 2015 shows the same trend. Among 141 surveyed articles and papers in this time interval, there are 96 (68.1%) addressing security management. In 2009, from 16 surveyed articles and papers, there are only two articles/papers that discuss other topics of network and service management. Figure 2.4 depicts these numbers in the time interval between 2004 and 2015.

2.1.5.1 Network and Service Management Taxonomy

Regarding network and service management, the proposed classification uses the taxonomy jointly defined by the Committee on Network Operations and Management (CNOM) of IEEE Communications Society, the Working Group 6.6 (IFIP WG 6.6) of the International Federation for Information Processing (IFIP), the Network Management

Figure 2.4: (a) Percentage of articles and papers addressing security and other fields in a time interval ranging from 2004 to 2015; (b) Distribution of articles and papers in the same time interval.



Source: by author (2016).

Research Group (NMRG) of the Internet Research Task Force (IRTF), and the European Network of Excellence for the Management of Internet Technologies and Complex Services (EMANICS) (CNOM, 2013). Such taxonomy has two levels: the first one indicates a broad area, whereas the second level more precisely refines that area. Table A.2 shows the network and service management taxonomy (see Appendix A).

This taxonomy was chosen because it is specific for network and service management, as well as being supported by that community. Moreover, keywords of the taxonomy are used by journals and conferences in the field, in two ways: (*i*) for authors to annotate their papers; and (*ii*) for researchers to indicate their area of expertise and interest. As such, this taxonomy matches with the purpose of this SLR.

2.1.5.2 InfoVis Taxonomy

Regarding the InfoVis taxonomy, there were identified three main criteria to classify the surveyed papers as follows: (*i*) dataset types, (*ii*) InfoVis techniques, and (*iii*) the available tasks/interactions for end-users. Although the InfoVis field has a set of taxonomies described in the literature, some of them are either too specific for the proposed classification (*e.g.*, (PRICE; BAECKER; SMALL, 1993) and (LEE et al., 2006)), or do not cover the aforementioned criteria (*e.g.*, (CHI, 2000) and (TORY; MöLLER, 2004)).

Price *et al.* (PRICE; BAECKER; SMALL, 1993) presents a taxonomy for classifying software visualization systems in a hierarchy of categories from a high-level division in six categories ranging from the scope of the software to the interaction it provides. Lee *et al.* (LEE et al., 2006) strongly focus on users' tasks: they describe a taxonomy of tasks for graph visualizations, so it could be used here to verify graph visualization techniques in terms of user tasks support.

On the other hand, Chi (CHI, 2000) and Tory and Möller (TORY; MöLLER, 2004), although proposing more general taxonomies, base their classifications on specific criteria. For example, Chi (CHI, 2000) actually proposes a reference model for visualization based on data transformation and then analyzes different techniques in terms of the operators they implement, while Tory and Möller (TORY; MöLLER, 2004) categorize visualization techniques based on their design models and not data, because to a certain extent a design model incorporates user needs, which is ultimately related to data.

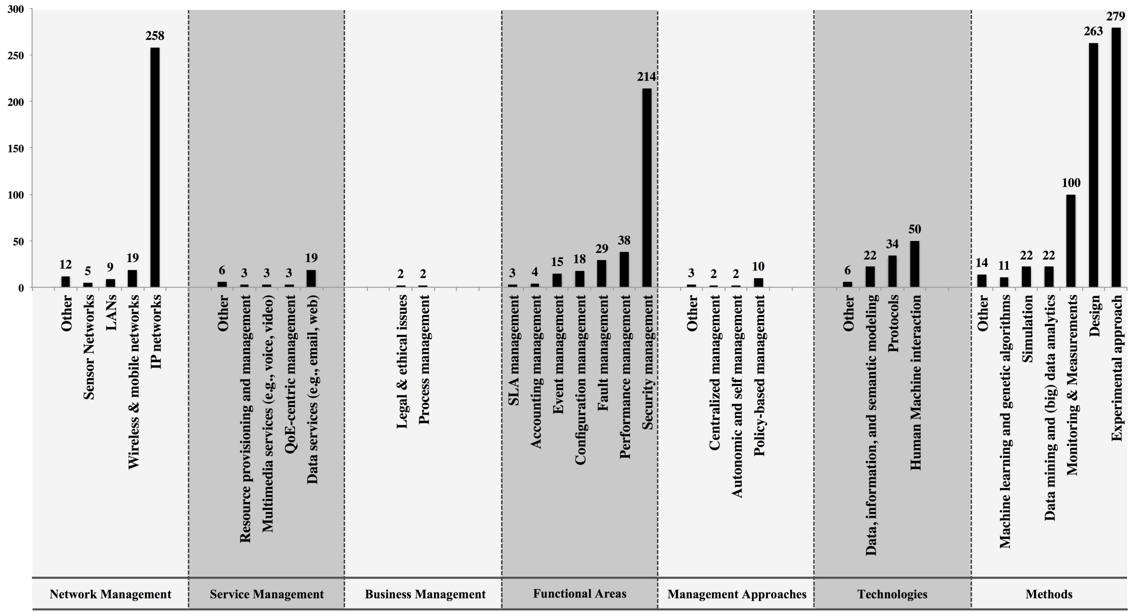
Additionally, by observing the taxonomy that covers those criteria (KEIM, 2002), if applied alone, is not suitable for the purpose of this SLR. Keim's taxonomy (KEIM, 2002) is based on a very general classification of data and, regarding interaction, there are some user tasks missing. Moreover, since 2002, taxonomies of users' tasks have been thoroughly discussed, and new application domains have contributed with new data types to this discussion.

Based on these issues, the proposed classification merges two taxonomies and a classification of dataset types to achieve the adequate framework needed for this SLR. For the first criterion, there was used the dataset types classification proposed by Munzner (MUNZNER, 2014). For the other two criteria, there were used the taxonomies proposed by Keim (KEIM, 2002) and Shneiderman (SHNEIDERMAN, 1996), which is a classical one. Such taxonomies are widely accepted and referenced by the visualization community. The taxonomy proposed by Keim was used to classify the InfoVis techniques. This decision came from the fact that the taxonomy proposed by Shneiderman is focused on tasks and data types. Then, for the tasks/interactions criterion, the proposed classification merges Keim's and Shneiderman's taxonomies. A new task called *Moving/Rotate* was also added to include 3D visualization techniques. Table A.1 shows the InfoVis taxonomy defined for this SLR (see Appendix A).

2.1.5.3 Results and Discussion

First, it is important to highlight that the classification on taxonomies subtopics is not mutually exclusive. Thus, the same article may appear on more than one subtopic of each taxonomy. The analysis of the classification begins pervading each topic of the network and service management taxonomy as shown below. Fig.2.5 shows the number of classified articles/papers in each subtopic of such a taxonomy.

Figure 2.5: The 303 surveyed articles and papers classified following the network and service management taxonomy. Each topic of the taxonomy is identified in the bottom of the chart. Moreover, subtopics are separated by dashed lines. Black bars depict the number of articles and papers classified in the taxonomy subtopics



Source: by author (2016).

1. **Network management:** IP networks prevails over other subtopics with 258 classified papers. This result is expected since the IP protocol is widely adopted. On the other hand, there are almost no works on hot topics such as virtual networks, data center networks, and software-defined networks. Still within the network management topic, it is possible to identify some efforts on wireless and mobile networks.

2. **Service management:** here, the most frequent topic is Data Services (*e.g.*, e-mail, Web) with 19 classified articles/papers. Indeed, it is curious the result regarding the service management subtopic because data services, multimedia services, hosting, and cloud services are consolidated for the industry as well as the academic community has widely researched them over the last years. Thus, it was expected much more efforts on these fields since they are aligned to the myriad of services currently offered by providers.

3. **Business management:** based on this classification, it is possible to state that this topic is off target for both communities, and it may be regarded as an open issue.

4. **Functional areas:** as previously stated, security management is predominant in this topic. Another important aspect of this topic refers to the SLA management and account management because there are a few efforts regarding InfoVis for these subtopics. In the age of Cloud computing, SLA management, and account management are critical management areas, since a huge amount of tools and applications are offered as a service and on a pay-per-use model. Thus, InfoVis could be helpful for network administrators in SLA and account management.

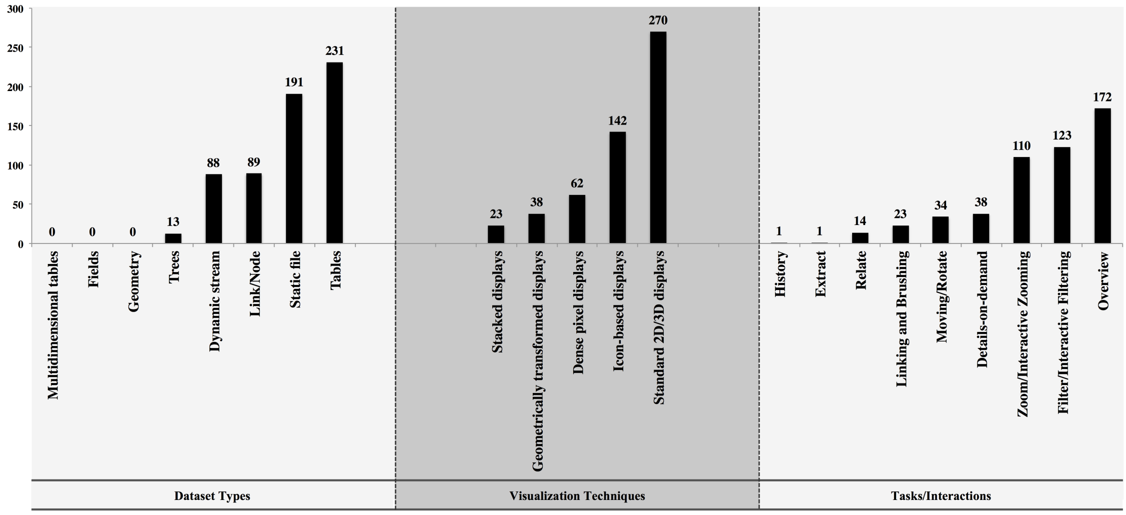
5. **Management approaches:** the numbers reveal fewer efforts in this topic. The most frequent subtopic is policy-based management that appears with only ten classified articles/papers. Autonomic and self-management, centralized management, and distributed management show only five articles/papers. Moreover, no works addressing energy-aware network management, which is an interesting topic to be explored through visualizations, in special for green awareness management.

6. **Technologies:** here, the most frequent subtopics are human machine interaction (with 50 classified articles/papers), protocols (with 34 classified articles/papers), and data, information, semantic modeling (with 22 classified articles/papers). Again, there is a low number of articles/papers addressing Cloud computing. Additionally, there are almost no articles/papers on subtopics such as grids and peer-to-peer.

7. **Methods:** on this topic, the two predominant subtopics (design and experimental approach) appear in almost all articles/papers. In general, surveyed articles and papers are structured describing the design of the proposal, and experimental approaches are used to show obtained results. Regarding monitoring & measurements subtopic, it was expected to find more than 100 articles/papers ($\simeq 33\%$ from the total of 303 classified articles and papers) in this subtopic due to the fact that Monitoring & measurements are one of the pillars of network and service management. Moreover, monitoring & measuring generate a plentiful amount of data that network administrators need to analyze to extract relevant information and, then, perform management tasks. Finally, up to now, the data mining and (big) data analytics subtopic had small attention from the network community (22 classified papers).

For the classification following the InfoVis taxonomy, each topic of the taxonomy is also discussed as shown below. Fig. 2.6 shows the number of classified articles/papers in each subtopic of such a taxonomy.

Figure 2.6: The 303 surveyed articles and papers classified following the information visualization taxonomy. Each topic of the taxonomy is identified in the bottom of the chart. Moreover, subtopics are separated by dashed lines. Black bars depict the number of articles and papers classified in the taxonomy subtopics



Source: by author (2016).

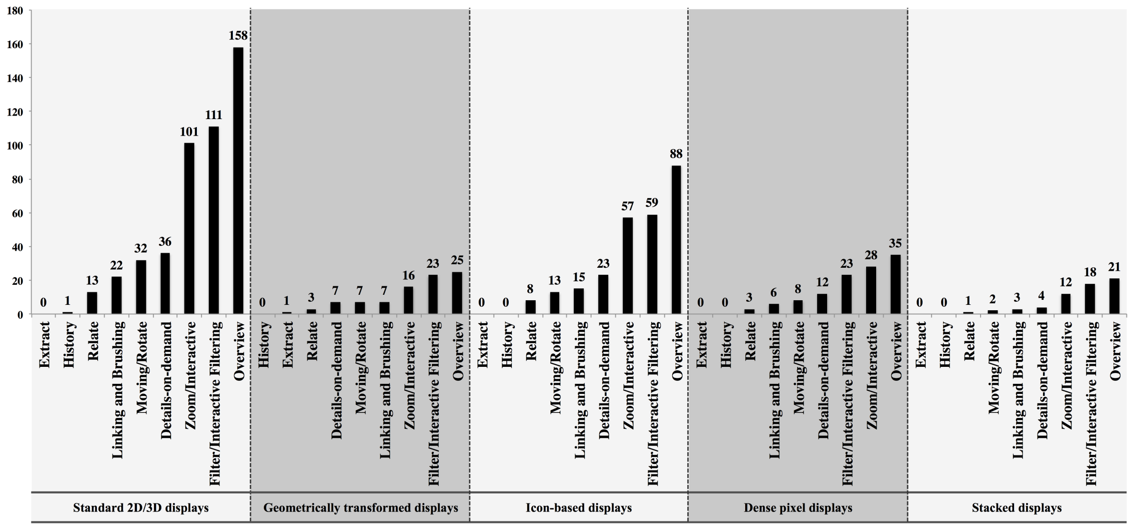
1. **Dataset types:** in this topic, tables, and static files prevail over other subtopics, followed by link/node and dynamic stream. Tree appears only in a few works. In general, during the analysis of the surveyed articles and papers, it was identified that some works do not clearly explain the used dataset. Thus, in such cases, it was necessary to analyze some peripheral information along the article/paper to infer about the used datasets. This finding is pointed out here because information regarding the dataset types could be helpful in two ways at least: (*i*) for researchers that are interested in comparing/reproducing experiments using the same dataset type; and (*ii*) for general readers, since information about datasets make clear from which data the proposed visualizations are built.

2. **Visualization techniques:** in this topic, standard 2D/3D displays subtopic prevails. There were analyzed several works using 3D views to allow better visual analysis. Icon-based displays are also found in several works. In this subtopic, Glyph-based representations appear in many cases. For instance, in a graph layout, nodes have a specific shape according to their role or status. Dense pixel displays are represented by colored pixel maps. Geometrically transformed displays are represented by parallel coordinates. In essence, several works use parallel coordinates to outline patterns and behaviors. Stacked displays appear through TreeMap views.

3. **Tasks/interactions:** in this topic, three subtopics prevail as follows: overview, filter, and zooming. Moving/rotate, details on demand, and linking and brushing features are proposed only in a few works, in special comparing with the total of 303 surveyed articles and papers. History and extract each appear once, in only one article. It is also important to highlight that there is a significant number of articles and papers (specifically, 106 articles and papers) that were not classified into Tasks/Interactions topic. In several cases, it is unclear whether authors do not highlight these features (*e.g.*, to save paper space in order to show other relevant information) or, in fact, the proposed visualization does not provide such features. Based on these findings, it is possible to assume that tasks and interactions can be an important issue to be addressed in future proposals.

After showing how surveyed articles and papers fit in both taxonomies separately, the visualization techniques and tasks/interactions topics are correlated. Specifically, this correlation intends to highlight how each tasks/interactions subtopic is explored in the scope of visualization techniques and it may provide useful insights such as (*i*) few articles and papers that have explored standard 2D/3D displays use linking and brushing interaction; and (*ii*) overview appears as the most explored task/interaction for all visualization techniques. Fig. 2.7 shows the number of papers that use each tasks/interactions subtopic for each of the visualization techniques.

Figure 2.7: A classification that shows how tasks/interactions subtopics appear together with subtopics of the visualization techniques topic. Each subtopic of visualization techniques is identified in the bottom of the chart. Moreover, visualization techniques subtopics are separated by dashed lines. Black bars depict the number of articles and papers that use each subtopic of the tasks/interactions topic



Source: by author (2016).

Finally, the classification enables to depict how each network and service management taxonomy subtopic is related to information visualization taxonomy subtopics. Fig. 2.8 shows a heatmap where rows represent the network and service management taxonomy and columns the information visualization taxonomy. Such a representation outlines the big picture regarding information visualization for network and service management. Based on Fig. 2.8, one can extract insights such as: (*i*) in security management the most used visualization techniques are standard 2D/3D displays and Icon-based displays, (*ii*) in monitoring & measurements only eight surveyed articles and papers have used geometrically transformed displays, and (*iii*) by correlating Fig. 2.8 and Fig. 2.5 it is possible to infer that all works addressing performance management (*i.e.*, 38 articles and papers) have used standard 2D/3D displays as a visualization technique.

2.2 Software Reuse

According to the Oxford Dictionaries, the verb reuse is defined as "*use again or more than once*" and the noun as "*the action of using something again*". In the context of computer programming, the reuse is an early practice that starts by the simple act of copy pieces of code used in a certain program to be *reused* in another one. In the software engineering domain, the concept of software reuse was born in the 1968 NATO Software Engineering Conference that has focused on the software crisis, *i.e.*, *the problem of building large, reliable software systems in a controlled, cost-effective way* (KRUEGER, 1992).

Nowadays, software reuse is a distinct research line that aims at investigating patterns, strategies, and mechanisms to support the optimal reuse of resources (*e.g.*, software components, libraries, and knowledge) in the software industry. Mohagheghi and Conradi (MOHAGHEGHI; CONRADI, 2007) provide a comprehensive definition of software reuse as follows:

"*Software reuse is the systematic use of existing software assets to construct new or modified ones or products. Software assets in this view may be source code or executables, design templates, free standing Commercial-Off-The-Shelf (COTS) or Open Source Software (OSS) components, or entire software architectures and their components forming a product line or product family.*"

The use of existing software artifacts or knowledge (also referred as reusable assets) to create new software is a key method for significantly improving software quality

		Dataset types								Visualization Techniques					Tasks/Interactions								
		Tables	Multidimen-sional tables	Link/Node	Trees	Fields	Geometry	Static file	Dynamic stream	Standard 2D/3D displays	Geome-trically transformed displays	Icon-based displays	Dense pixel displays	Stacked displays	Overview	Zoom/ Interactive Zooming	Filter/ Interactive Filtering	Details-on-demand	Relate	History	Extract	Linking and Brushing	Moving/ Rotate
Network Management	Ad-hoc networks	1		1				1	2	4		2	1		4	2	2	1	1			1	2
	Wireless & mobile networks	9		9				8	8	19		9	3		12	6	5		3			1	3
	IP networks	208		67	11			168	73	227	34	121	58	22	142	94	105	32	9	1	1	20	29
	LANs	8						7	2	7	2	5			6	5	4	1				1	1
	Optical Networks	1							1	1		1			1		1						
	Sensor Networks			4				1	3	5		1			4	2	2						
	Overlay Networks																						
	Virtual Networks			1				1		1		1			1		1						
	Software Defined and Programmable Networks	1		2					2	2		2			2			1					
	Data Center Networks	1		1				3		2	1	1			1							1	
	Smart Grids			1					1	1					1	1	1	1	1				
Service Management	Multimedia services (e.g., voice, video)	1		1				2	1	3		2	1		1	1							
	Data services (e.g., email, web)	11		6	1			12	6	19	1	8	2	2	11	3	11	1	1			1	5
	Hosting (virtual machines)	1		1				1	1	2		1			2		1						1
	Grids																						
	Cloud services	1		1				1		2		2			1	1						1	
	Resource provisioning and management			2	1			2		2		3			3	1	1	1					1
	QoE-centric management	1		3				1	2	3		1	1		1	1							
	Service discovery, migration and orchestration	2		1				1	1	2		2			1	1	1						
B.M.*	Legal & ethical issues	1		1				2		2					1	1							
	Process management	1		2				1		1		2			2	1	1	2	1				1
Functional Areas	Fault management	21		7				19	9	27		17	2	1	13	5	8	1				2	3
	Configuration management	10		6				10	8	17	1	13	2		3		3					1	
	Accounting management	3		1				2	2	4		2			2	1	1	1					
	Performance management	27		9				18	18	38		19	3	2	18	9	12	5				3	5
	Security management	173		55	8			136	62	183	35	93	55	22	122	80	96	25	11	1	1	17	25
	SLA management	1		1	1			1	2	3		2			2	1	1	1					
	Event management	12		5	1			9	5	14	2	9	1	2	8	3	6	3				2	1
Management	Centralized management			2					1	2					1	1							
	Distributed management								1	1		1											
	Autonomic and self management	1		1				1	1	2		1	1		2		2						
	Policy-based management	5		3	2			4	3	9		5	1		4	3	3	1				1	2
	Federated network management	1							1	1			1		1		1						
	Pro-active management	1						1		1		1	1		1	1	1						1
	Energy-aware network management																						
Technologies	Protocols	23		15	4			26	7	32	2	18	7	2	15	8	10	1	1	1		3	2
	Middleware																						
	Mobile agents																						
	P2P	2		1				2		2		2			2	1							
	Grid							1			1												
	Data, information, and semantic modeling	18		5	1			15	5	20		12	2	2	11	8	13	4	1			1	
	Cloud computing			1	1			2		2	1	1			1	1	1						
	Internet of Things																						
	Human Machine interaction	39		13	1			29	19	43	9	28	10	5	38	22	30	9	3			7	11
	Operations and Business Support Systems																						
Methods	Control theories							1		1					1		2						1
	Optimization theories	1		2				1	1	2					1		1						
	Economic theories																						
	Machine learning and genetic algorithms	10		4				9	2	8	2	5	5		5	1	5	2	1				2
	Logics																						
	Probabilistic, stochastic …**	6		5				7	3	8	2	3	3	1	4	1	4	3	1			2	
	Simulation	13		12				15	3	20	2	10	6	1	14	10	9	2	2			2	5
	Experimental approach	213		84	12			177	80	250	32	128	55	20	158	101	112	36	10	1		20	32
	Design	202		79	11			165	76	237	28	128	57	18	142	98	103	32	9	1	1	18	28
	Monitoring & Measurements	80		22	4			55	42	89	9	42	27	10	58	37	37	12	2			6	9
	Data mining and (big) data analytics	18		6				17	3	21	1	12	4	2	11	7	10	1	2		1	1	2

** Business Management (B.M.)*
*** Probabilistic, stochastic processes, queuing theory*

0	1-5	6-15	16-20	21-25	26-40	41-65	66-200	> 200

Range of values

Source: by author (2016).

and teams' productivity with the ultimate goal of maximizing profits (FRAKES; KANG, 2005) (FRAKES; TERRY, 1996). In this sense, software reuse has been studied and evaluated from different perspectives concerning the business and financial points of view such as cost/productivity models, quality of investment, business reuse metrics, and the relation of reuse to quality and productivity (FRAKES; TERRY, 1996). Another important aspect related to the software reuse discipline is the reusability that indicates the degree of reuse of a reusable asset. Typically, the reusability is the most important metric to qualify a reusable asset.

As previously introduced (see Section 1.1), this thesis claims that the building up of InfoVis techniques is primarily a software development task. In this way, this thesis investigates the adaptation of software reuse concepts, relying on the legacy and maturity of this research field. Next sections present the background and literature review regarding the central concepts that were investigated to build the proposal introduced in this thesis. First, Section 2.2.1 introduces a brief review of software reuse standards. Next, Section 2.2.2 presents the principles of CBD. Afterward, Section 2.2.3 brings the architecture patterns to CBD. Finally, Section 2.2.4 presents the Semantic Web as a mean to promote and enhance the reuse.

2.2.1 ISO/IEC - IEEE Standard 12207

Efforts on software reuse standards started in April 1995 when the IEEE Software Engineering Standards Committee (SESC) created the Reuse Planning Group. SESC is the authority to develop standards for software engineering, and it is under the umbrella of the Computer Society of the IEEE. In summary, the Reuse Planning Group was created to perform the following task (BALDO JR.; MOORE; RINE, 1997):

"*The Reuse Planning Group will define for SESC a statement of direction for IEEE standards related to the analysis, design, implementation, validation, verification, documentation, and maintenance of reusable software assets as well as their supporting infrastructure in the creation of new applications.*"

Nowadays, the Software Reuse Process is standardized in the Software Specific Processes defined in the international standard ISO/IEC 12207 / IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes (ISO/IEC/IEEE..., 2008). The first version of the ISO/IEC 12207 was published on 1 August 1995 and was the first International Standard to provide a comprehensive set of life cycle processes, activities, and tasks for software that is part of a larger system, and for stand-alone software products and services. The current version of the ISO/IEC 12207 / IEEE Std 12207-2008 is the result of a coordinated effort by IEEE and ISO/IEC JTC 1/SC 7. In this document the Software Reuse Process is divided into three main axes as follows:

1. **Domain Engineering Process** - The domain engineering is an approach based on reuse that aims at defining the scope (the domain definition), specifying the structure (the domain architecture), and building the assets (e.g., requirements, designs, software code, documentation) for a class of systems, subsystems, or applications. Thus, the goal of the Domain Engineering Process is to develop and support domain models, domains architectures, and assets for the domain.

2. **Reuse Asset Management Process** - The purpose of the Reuse Asset Management Process is to manage the life cycle of reusable assets. In this sense, this process provides the guidelines to control reusable assets from their conception to retirement.

3. **Reuse Program Management Process** - The goal of the Reuse Program Management Process is to plan, establish, manage, control, and monitor the program of reuse of a certain organization mainly to exploit reuse opportunities systematically.

By analyzing in details each one of the above Software Reuse Process axes, it is identified that the Reuse Asset Management Process fits with the requirements of the reuse-based approach proposed in this thesis mainly because it focuses on the guidelines to managing reusable assets. In essence, there are seven expected outcomes as a result of successful implementation of the Reuse Asset Management Process as follows:

1. The documentation of the strategy to manage assets, *i.e.*, a management plan to define the resources and procedures for managing assets.

2. The establishment of an asset classification scheme, *i.e.*, the documentation that describes how the assets must be classified.

3. The definition of a set of criteria to the acceptance, certification, and retirement of assets.

4. The implementation of a mechanism for the storage and retrieval of assets.

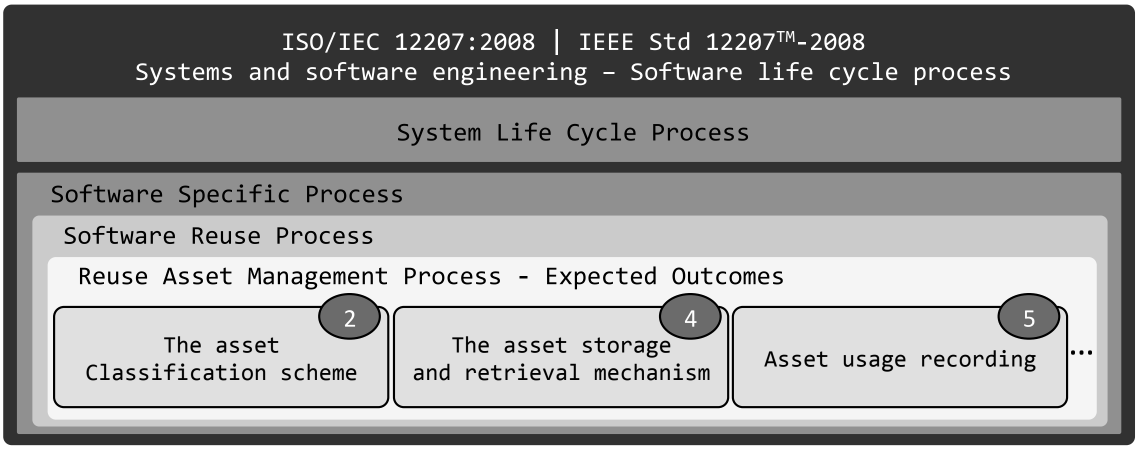
5. The development of a mechanism that allows recording the use of assets, *i.e.*, the registration of how the assets have been used.

6. The deployment of a mechanism to control changes in assets.

7. The employment of a tool to notify users of assets regarding problems detected, modifications made, new versions created and deletion of assets in the mechanism of storage and retrieval.

For adapting the Reuse Asset Management Process, this thesis uses the following expected outcomes presented in the list above (see Figure 2.9): item 2 (*i.e.*, the asset classification scheme), item 4 (*i.e.*, the asset storage and retrieval mechanism), and item 5 (*i.e.*, asset usage recording). Although the other outcomes defined by the Reuse Asset Management Process are also relevant, they are not analyzed and detailed in this thesis. The details about the adaptation of the expected outcomes of the Reuse Asset Management Process are presented in Chapter 3.

Figure 2.9: Reuse Asset Management Process with the three expected outcomes adapted to this thesis



Source: by author (2016).

2.2.2 Component-Based Development

Over the years, investigations on software reuse grew mainly because software engineers have agreed that software does not have to be developed from scratch all the times. In this sense, CBD - also referred as Component-Based Software Engineering (CBSE) - is a structured approach with the primary focus on reuse, providing techniques that aim at improving the software development process, reducing the time to market and enhancing the quality of delivered software products (MAHMOOD; LAI; KIM, 2007). According to Lau *et al.* (LAU et al., 2013) the goal of CBD is to (*i*) build components and store them into a repository; and (*ii*) use or reuse these pre-existing components to create many different systems by assembling the components using pre-defined composition mechanisms.

In CBD, a software component is the fundamental building block of a software system. There are several definitions of the concept of a software component. This section checks three of them. Hopkins (HOPKINS, 2000) provides an abbreviated definition where "*a software component is a physical packaging of executable software with a well-defined and published interface*". D'Souza and Wills define software component as "*a coherent package of software artifacts that can be independently developed and delivered as a unit and that can be composed, unchanged, with other components to build something larger*". Szyperski (SZYPERSKI, 2002) introduces the following definition: "*A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties*".

Vitharana (VITHARANA, 2003) identifies the following key advantages of CBD: (*i*) it reduces the lead time by building complete business applications from an existing pool of components; (*ii*) it decreases cost by developing individual components and reusing them in multiple contexts; (*iii*) it enhances quality because components are reused and tested in many different applications; and (*iv*) it allows easy replacement of obsolete components with new enhanced ones.

Another important aspect regarding the CBD paradigm is the possibility of black-box reuse. In essence, the reuse can be done in two ways referred as white-box and black-box reuse. In the first one, programmers usually have access to the implementation code, and they can modify it to match with their specific requirements. Ravichandran and Rothenberger (RAVICHANDRAN; ROTHENBERGER, 2003) indicate potential disadvantages of white-box reuse that include the large up-front investment necessary to populate a repository with reusable components, the problems associated with the classification and retrieval of reusable components, potential lack of management support, and the need to change the organizational structure and processes. On the other hand, black-box reuse entails using software components "as is", *i.e.*, with no code modification. Thus, developers shall only know the functionality of a software component and how it interfaces (*i.e.*, the inputs and outputs), and they do not need make changes in the internal logical and code of the component.

In a recent investigation, Sinha and Jain (SINHA; JAIN, 2013) introduced an empirical comparison in terms of ease of reuse between the CBD (in a black-box fashion) and Object-Oriented Analysis and Design (OOAD) paradigms. They used a real-world business case, which describes the requirements for developing an auto insurance claims processing system, from a multinational Information Technology (IT) consulting company. Seventy-six IT professionals with zero to minimal experience in OOAD and CBD methods participated in the study. The goal was to examine how easy it is to reuse assets from component/class libraries, even without prior experience in those methods. The results showed that IT professionals found it easier to reuse components than objects.

In this thesis, it is understood that the CBD paradigm delivers the key characteristics to design and build the proposed reuse-based approach. First, as previously highlighted, the primary focus of CBD is reusability. Second, CBD enables a black-box reuse which can significantly reduce cost and improve productivity (HAINES; ROTHENBERGER, 2010). Third, CDB provides an easy way to achieve the fourth expected outcome of the Reuse Asset Management Process (see Section 2.2.1), since each component is an independent unit. Finally, the work of Sinha and Jain (SINHA; JAIN, 2013) demonstrated that IT professionals find it easier to reuse components than objects. This is an important finding, since this thesis understands that the profile of IT professionals and network administrators is too similar, specifically because both, in general, are non-experts in programming (RENDON; ESTRADA-SOLANO; GRANVILLE, 2013).

2.2.3 Architecture

According to Bass *et al.* (BASS; CLEMENTS; KAZMAN, 2003) "*The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.*" Frakes and Kang (FRAKES; KANG, 2005) highlight that architectural decisions have been recognized as an important consideration for reusing software because they occur early in the software lifecycle, have a substantial impact on system quality attributes, and are difficult to change late in the lifecycle. Currently, with the advent and consolidation of the Web technologies, two software architecture styles are wide adopted with the intention to promote reuse: Service-oriented Architecture (SOA) and Resource-Oriented Architecture (ROA).

Papazoglou and Georgakopoulos (PAPAZOGLOU; GEORGAKOPOULOS, 2003) defined SOA as an architectural style for building software applications that use available services in a network. In such a context, a service is in an application function packaged as a reusable component for use in a business process (GRIFFIN; PESCH, 2007). In general, the SOA architecture employs the *Big*[1] Web services technology stack that includes the Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), WS-Notification, WS-Security, and eXtensible Markup Language (XML). Also, an SOA-based service must be designed and developed following specific architectural principles as shown below (BRANCA; ATZORI, 2012):

- **Encapsulation -** a service has well-defined capabilities and is able to communicate with other services through standardized interfaces.
- **Loose coupling -** services maintain a relationship with others that minimize dependencies and only require they maintain an awareness of each other.
- **Contract -** services subscribe to a communication agreement with other services.
- **Abstraction -** services hide their own internal logic from the outside world.
- **Autonomy -** services have to be well-defined, complete and independent from the context or the state of other services.
- **Reusability -** the logic is divided into services with the intention of promoting reuse.
- **Composability -** many services can be coordinated and assembled to form complex services.
- **Statelessness -** services minimize retaining information specific to an activity.
- **Discoverability -** services are defined to be externally descriptive so that they can be found and assessed via an available discovery mechanism.

Regarding ROA, the term was coined by Richardson and Ruby (RICHARDSON; RUBY, 2007) and, in summary, it can be defined as an architecture style for RESTful Web services, *i.e.* Web services that are based on Representational State Transfer (REST) design principles. The ROA architecture has two major entities: (*i*) resources; and (*ii*) URIs. A resource is something that can be stored on a computer and represented as a stream of bits (*e.g.*, a software component, a document, or an application). In its turn, the URI is the name and address that identifies a resource. ROA has four basic properties:

[1] The term *Big* Web Services was introduced by Richardson and Ruby (RICHARDSON; RUBY, 2007)

1. **Addressability -** An application is addressable if it exposes its data as resources that are accessible through URIs.

2. **Statelessness -** It means that every resource request happens in complete isolation, *i.e.*, when the requester makes a request, it includes all information necessary to fulfill that request without relying on information from previous requests.

3. **Connectedness -** It is the quality of resources to have links that put them in different states just by following links. In essence, resources should link to each other in their representations.

4. **A uniform interface -** There is a uniform interface from which every service use HTTP's interface the same way. For example, across the Web, the GET operation means "read", no matter the resource that is using it.

Investigations that compare RESTful Web Services and Big Web Services in terms of reusability and loose coupling (PAUTASSO; WILDE, 2009)(PAUTASSO; ZIMMERMANN; LEYMANN, 2008) suggest that Big Web Services are preferred for professional enterprise application integration scenarios, whereas RESTful Web Services for tactical, *ad-hoc* integration over the Web (GUINARD; ION; MAYER, 2012). In essence, RESTful Web services are regarded as an efficient and lightweight alternative to the Big Web Services (BELQASMI et al., 2012). In contrast to the Big Web services, RESTful Web services rely only on HTTP to exchange messages at the application layer by offering four basic operations, Create/Read/Update/Delete (CRUD), that are implemented using the following four HTTP methods: POST, GET, PUT, and DELETE (FU; BELQASMI; GLITHO, 2010). Based on the presented characteristics, the proposal of this thesis relies on the ROA software architecture style because it is simpler, straightforward, and compliant with the CBD principles.

2.2.4 Semantic Web

The term Semantic Web was defined by Tim Berners-Lee as a giant global semantic network of data that is directly consumable and understandable to machines (CHEN; WU; CUDRE-MAUROUX, 2012). Essentially, in the *traditional Web*, texts are linked to other texts through hyperlinks. These hyperlinks can connect to texts at the same site or texts stored in other places (*i.e.*, hyperlinks to external resources). The Semantic Web, in its turn, intends to connect data objects instead of only text. Thus, data objects are linked to other data objects across the Web by means of formal semantics (*e.g.*, ontologies).

Nowadays, the Semantic Web and its related technologies are led by the World Wide Web Consortium (W3C) and aims to "*enable people to create data stores on the Web, build vocabularies, and write rules for handling data*" (W3C, 2016). The principles that materialize the Semantic Web can be summarized as follows (CHEN; WU; CUDRE-MAUROUX, 2012) (SHADBOLT; HALL; BERNERS-LEE, 2006):

1. Resources are identified by a Uniform Resource Identifier (URI), *i.e.*, a global naming convention defined as "*a compact sequence of characters that identifies an abstract or physical resource*" (BERNERS-LEE; FIELDING; MASINTER, 2005).

2. The URIs must have global scope, be interpreted across contexts, and be referred to and de-referenced by both humans and machines.

3. When a URI is dereferenced, the information about the resource must be published in a machine-readable format by using semantic representation.

4. URI-identified resources are linked together with semantic annotations to facilitate discovery of related information.

This thesis argues that the Semantic Web concepts and its technologies provide helpful capabilities for supporting and improving the reuse of assets. In special, the Semantic Web allows to achieving the expected outcomes (items 2 and 5) of the Reuse Asset Management Process (see Section 2.2.1), which are adapted for the reuse-based approach proposed in this thesis. In essence, it is understood that the software reuse process is based on a set of relevant information that describes reusable assets by documenting, for example, the input parameters and the output data generated by a software element. Such information is essential to assist stakeholders for reusing assets in an easier way and, consequently, to improve their reusability. In this sense, such relevant information must be organized, stored, and shared to avoid, among other things, the loss of background knowledge, or ambiguities caused by heterogeneous descriptions.

2.3 Summary

This chapter presented an SLR regarding the employment of InfoVis for network and service management. The SLR introduces an important contribution of this thesis by (*i*) drawing a historical overview of the research on InfoVis for network and service management based on 303 articles and papers published from 1985 to 2015, (*ii*) classifying each article/paper according to both a network and service management taxonomy and an information visualization taxonomy, highlighting how each taxonomy is filled by the surveyed work; and (*iii*) surveying the literature on information visualization and network and service management in a large (*i.e.*, number of articles and papers) and comprehensive (*i.e.*, several topics on network and service management) way. Through that SLR, it is also possible to check how the investigations on InfoVis for network and service management were conducted up to now. For example, several proposals investigate InfoVis to help in security management, whereas a few works address trending topics (such as SDN and IoT) in the network and service management domain. Further than that, the comprehensive understanding of the state-of-the-art provided by the SLR was fundamental for this thesis because: (*i*) it corroborates InfoVis for network and service management as an significant research topic; and (*ii*) it shows that there are no efforts addressing reuse as a means to promote the adoption of visualizations by network administrators.

Also, this chapter presented the background regarding software reuse and the concepts that are leveraged in this thesis. The international standard ISO/IEC 12207 / IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes denotes the maturity of the software reuse process and provides a general and flexible guidelines to the reuse asset management process. Aligned to this standard, the CBD and the ROA architecture style provide the fundamental concepts and best practices to the design and development of a reuse-based approach. The Semantic Web, in its turn, defines principles and technologies that can be applied to organize, store, and share relevant information to assist stakeholders in the process of reuse.

3 PROPOSAL

This chapter presents the main concepts that drive the reuse-based approach proposed in this thesis. First, Section 3.1 introduces the three types of stakeholders: network administrators (Section 3.1.1), InfoVis designer (Section 3.1.2), and web developer (Section 3.1.3). Next, Section 3.2 presents an overview of the proposed solution. Essentially, Section 3.2 begins by explaining the concept of reusable asset and the two types of reusable software components (*i.e.*, *Data Wrapper* and *Visualization*) in the context of the proposed approach. Still in Section 3.2 is described the adaptation of three expected outcomes of the Reuse Asset Management Process as follows: the asset classification scheme (Section 3.2.1), the asset storage and retrieval mechanism (Section 3.2.2), and the asset usage recording (Section 3.2.3). Section 3.3 details the conceptual architecture together with the explanation about the software elements and its interactions. Finally, Section 3.4 overviews the prototype implementation that was developed as a proof-of-concept of the proposed approach.

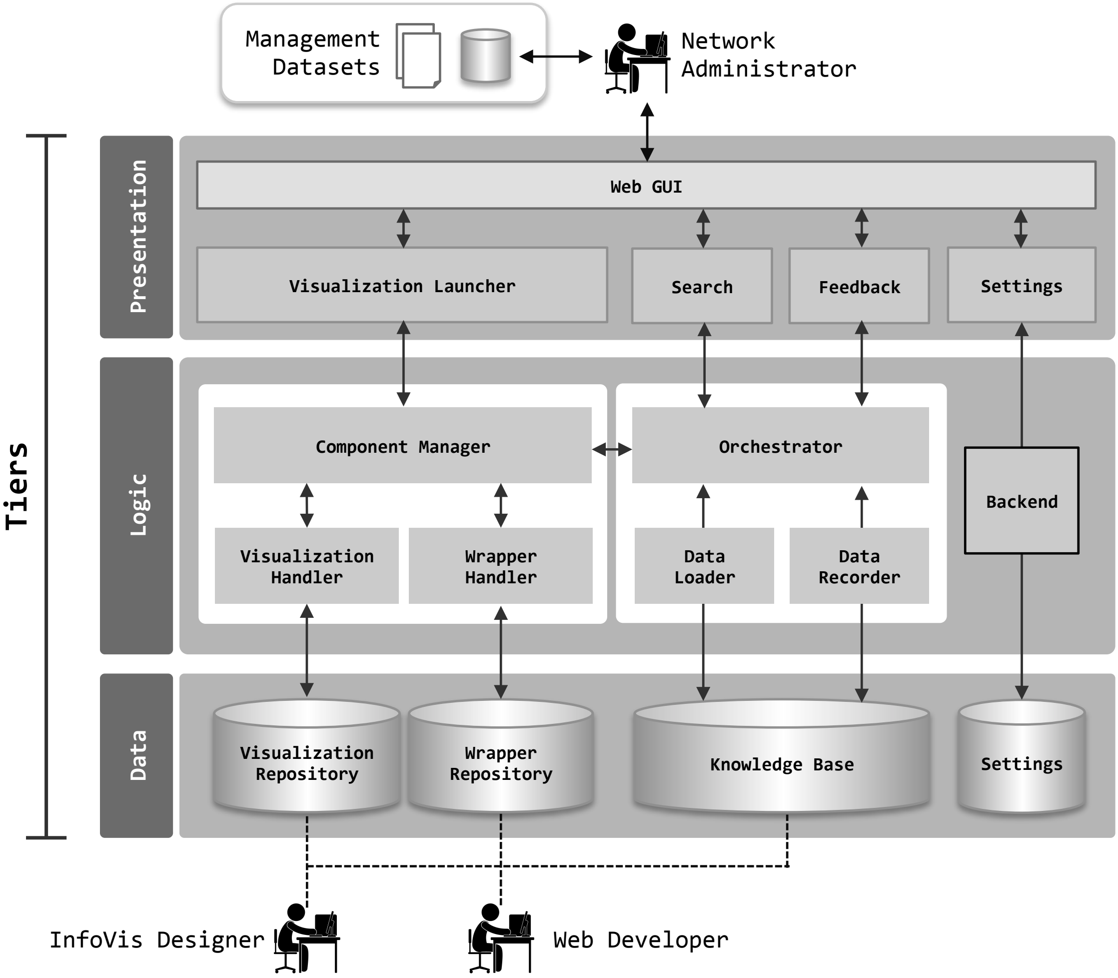
3.1 Stakeholders

As previously introduced, this thesis argues that network administrators are unskilled on InfoVis as well as they are non-experts in programming (RENDON; ESTRADA-SOLANO; GRANVILLE, 2013). Thus, the proposed solution envisages three types of stakeholders that perform complementary roles: the Network Administrator, InfoVis Designer, and Web Developer. Figure 3.1 depicts the conceptual architecture together with these stakeholders. Also, this section describes the role of each stakeholder separately, in Sections 3.1.1, 3.1.2, and 3.1.3.

3.1.1 Network Administrator

In a broad vision of network and service management, the network administrator has several responsibilities and tasks that aim at maintaining the managed infrastructure in a consistent and, preferably, optimized state. In 1990, Bokkelen, in the Request For Comments (RFC) 1173 (BOKKELEN, 1990), presented the responsibilities of “Network Managers” in the context of the “Oral Tradition” of the Internet. In essence, Bokkelen stated

Figure 3.1: Conceptual architecture and stakeholders



Source: by author (2016).

that "Network Managers" are responsible for "*every IP net or subnet that is connected to the Internet*". Nowadays, the role of network administrators is well-consolidated, and the description of their primary responsibilities and tasks can be found in a different level of details according to the requirements of each company. This thesis summarizes a non-exhaustive list of network administrators' responsibilities and tasks as follows:

- Designing and planning logical and physical network by making decisions about the type of network that best suits the needs of the organization.
- Building up configuration standards based upon best practices.
- Setting up networked systems such as Local Area Networks (LANs), network segments, Wide Area Networks (WANs), Virtual Private Networks (VPNs), Voice over IP (VoIP), Intranet, and the Internet.

- Administering network services such as e-mail, directory services, DNS, Dynamic Host Configuration Protocol (DHCP), Network File System (NFS), and application and database services.
- Monitoring and analyzing the network performance to ensure Quality of Service (QoS) and the Service Level Specifications (SLSs) defined in SLAs.
- Ensuring the security and availability of the network, services, and systems.
- Assigning routing protocols and routing table configuration.
- Developing and reviewing operational, technical and procedural documentation concerning network and service.
- Troubleshooting and repairing of network devices and services.
- Tuning network and its services to keep performance requirements and optimization (*i.e.*, doing more with less).

In the context of the proposed solution, the network administrator is the key stakeholder. In essence, he/she is a professional interested in employing InfoVis techniques to help him/her in performing, for example, the above-mentioned tasks list. This thesis also divides the network administrators into two groups as follows:

(a) **Senior -** He/she has an in-depth knowledge and large experience in network and service management. Examples of a Senior can be team leaders with wide expertise in a certain management field such as security, performance monitoring, cloud services, or SDN-based networks.

(b) **Mid-level/Beginner -** This group includes professionals that do not have advanced skills in network and service management. In general, both are part of a management team. The mid-level ones are skilled in dealing with a pre-defined set of technical issues (*e.g.*, support and configuration), but they do not have a deep knowledge in the field. The beginners are professionals who are entering the network and service management world. In general, they are young and inexperienced.

Both groups take advantage of the proposed solution to facilitate the employment of InfoVis techniques for everyday tasks through the reuse of software components (see Section 3.2). The first group is also encouraged to contribute with its expertise to improve the knowledge database. For example, a senior network administrator provides background knowledge that helps to enhance the knowledge base regarding the network and service management domain, by describing management scenarios where reusable

InfoVis techniques can be helpful to analyze a particular type of network behavior. In the other hand, the second group is encouraged to rate and provide feedbacks about reusable visualizations that they have used, and to learn from the knowledge supplied by seniors.

3.1.2 InfoVis Designer

Before to explain about the InfoVis Designer in the proposed architecture, this thesis introduces a general overview concerning the role of designers in the InfoVis field. First of all, it is important to highlight an important point: InfoVis designer is not the same as a graphic designer. Graphic designers have skills to planning and projecting ideas and experiences through visual and textual content which includes images, words, or graphic forms (CEZZAR, 2016). In essence, the primary goal of such a professional is to make the presentation of information more attractive and generate an emotional response in the viewer. Although InfoVis designers also target appealing visualizations as a final product, their priority is on the accuracy and ease of data consumption within an enjoyable graphical interface. Thus, in general, a good InfoVis designer is also a good graphic designer.

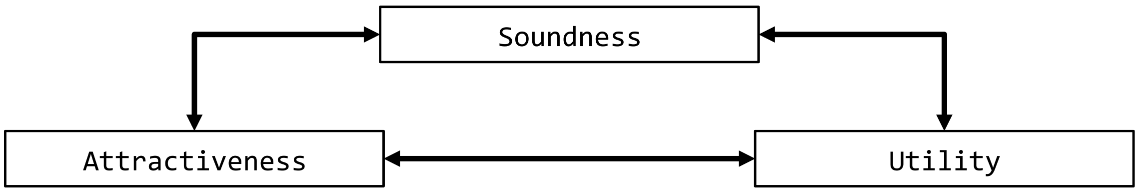
Moere and Purchase (MOERE; PURCHASE, 2011) highlight that the design and development of InfoVis techniques are beyond of just creating a reasonable working solution. In this sense, they argue that the design of InfoVis techniques must achieve a workable equilibrium between the typical architectural design requirements (see Figure 3.2): utility, soundness, and attractiveness. A brief explanation of how this triad of design requirements are related to the InfoVis context is presented below (MOERE; PURCHASE, 2011):

(a) **Utility -** It corresponds to quantitative performance measures such as functionality, usability, and usefulness. In InfoVis, these aspects are generally defined in terms of (*i*) effectiveness, *i.e.*, the accuracy and completeness with which users achieve specific tasks; and (*ii*) efficiency, *i.e.*, the number of consumed resources in relation to the effectiveness criterion.

(b) **Soundness -** This aspect addresses reliability and robustness. In InfoVis, in general, soundness relates to (*i*) the quality of the visualization presentation algorithm (*e.g.*, Does it crash? or How efficient is the code?); and (*ii*) the the extent of the algorithm, outside of its initial presentation (*e.g.*, the parallel coordinates technique can be considered “sound” because it has been widely adopted, optimized and used in several contexts after their initial conception).

(c) **Attractiveness -** It refers to the aesthetics characteristics of a visualization, *i.e.*, its appeal or beauty. The aesthetic is not limited to the visual form, but it also includes subjective features like originality, innovation and novelty, and other factors comprising the user experience.

Figure 3.2: Typical architectural design requirements



Source: by author (2016).

In the context of the proposed solution, the InfoVis designer is a professional with excellent skills in the InfoVis domain, in special to meet the design requirements above presented. He/she must also be qualified to designing and developing visualizations for Web-based interfaces (*i.e.*, using cutting-edge web front-end technologies like Hypertext Markup Language version 5 (HTML5), Cascading Style Sheets version 3 (CSS3), Scalable Vector Graphics (SVG), and JavaScript). In essence, the InfoVis designer is encouraged to exploring the network and service management domain to develop groundbreaking reusable visualizations (*i.e.*, “sound” InfoVis techniques), which is a promising way to enhance the *Visualization Repository* with novel visualizations. In addition, is expected that the InfoVis designer shares his/her know-how to: (*i*) help network administrators on features of reusable visualizations (*e.g.*, how to better explore interactive features of a given visualization); and (*ii*) provide background knowledge to improve the knowledge base in the InfoVis domain (*i.e.*, to enable collaboration with other InfoVis designers).

3.1.3 Web Developer

The third stakeholder of the proposed solution is the Web Developer. The name Web Developer may seem a little bit confusing due to the wide-range of tasks that a Web Developer can perform (STATISTICS, 2016). In this sense, this section starts with a brief contextualization in order to clarify the meaning of Web Developer in the proposed approach. In short, Web Developer is a professional that has skills on developing applications for the World Wide Web (WWW) or distributed network applications that run over the HyperText Transfer Protocol (HTTP) protocol (WIKIPEDIA, 2016). Thus, depending on the context, the responsibilities of a Web Developer can vary from the development of a website layout by using HTML and CSS technologies to the implementation of back-end business logic to retrieve and record data. Based on that, this section divides the Web Developer into three specialized types:

(a) **Back-end developer -** The back-end developer is in charge of programming the server-side features in a client-server architecture. For example, when the Model-View-Controller (MVC) design pattern is adopted, the back-end developer primarily handles the Model layer which manages the data, logic and business rules of a Web application. Currently, with the advent of the “Web of Data” and “Semantic Web” (W3C, 2016), it is also expected that back-end developers have skills to gather and integrate data from different sources (*e.g.*, back-end services and databases), and expose such data in standardized formats (*e.g.*, XML or JavaScript Object Notation (JSON)) via Web Services.

(b) **Front-end developer -** They can be subdivided into two categories. The former category is the web designer, who is responsible for creating the site’s layout (*i.e.*, how the website looks) and integrate content into the site. In general, the web designer is skilled on web-design applications and markup and formatting languages like HTML and CSS. The second is the front-end developer, who is specialized in programming the website on the client-side by using, in most cases, the JavaScript language and its related technologies (*e.g.*, Asynchronous JavaScript and XML (AJAX)).

(c) **Webmaster -** They are responsible for maintaining websites and to keep them updated. It includes ensuring that the website operates correctly and test potential errors like broken image/links. In many cases, webmasters are also in charge of replying to user comments.

Taking into account the above list, the Back-end developer is the specialized type of Web Developer that matches with the requirements of the proposed approach. Thus, from now on, the term Web Developer designates the stakeholder with similar skills of a Back-end developer. So, in the context of this proposal, the Web Developer has advanced skills in programming (*i.e.*, on Web technologies), in special, on Application Programming Interfaces (API) (*e.g.*, to pull data from management datasets). Thus, Web developers are encouraged to increase the Wrapper Repository by developing new wrappers (*i.e.*, software components). They also take advantage of the background knowledge available in the knowledge base to understand the problem domain. For example, to develop a new wrapper software component, the Web Developer must be aware of the specific characteristics of the management data in order to pull and parse data properly.

3.2 Overview

As previously presented, this thesis argues that the building up of visualizations is essentially a software development task. Thus, it is understood that the knowledge provided by the software reuse discipline can be leveraged for the proposed approach since it is a traditional and well-established practice in software development focusing on, among other things, improving quality, increasing productivity, and decreasing cost (FRAKES; KANG, 2005). As introduced in Chapter 2 (Section 2.2.1), this proposal relies on three expected outcomes of the Reuse Asset Management Process defined within the IEEE Std 12207-2008. In essence, this standard describes the expected outcomes of the Reuse Asset Management Process in a general manner, *i.e.*, there is no low-level specification concerning mechanisms, techniques, or technologies to achieve such outcomes. Thus, the IEEE Std 12207-2008 is flexible enough to be adapted to needs of the proposal presented in this thesis.

First of all, this section starts by introducing the definition of the term $asset$. In the system and software domain, the term $asset$ is defined as "*an item such as design, specification, source code, documentation, test suites, manual procedures, etc., that has been designed for use in multiple contexts*" (IEEE..., 2010). In this thesis, the term $asset$ is defined as *a software component designed and developed to enable a black-box reuse, and with the goal of promoting the use/reuse of InfoVis techniques for network and service management*. In this way, there are two types of reusable assets in the context of the proposed approach as follows:

(a) **Data Wrapper -** It is a software component designed and developed to gather data from a given management dataset, process such data, and generate an output from the processed information in a well-defined format. In summary, *Data Wrappers* operate as gateways between the management dataset and the *Visualization* component.

This approach enables *Visualization* components to be agnostic from the management dataset, since the particular aspects (*e.g.*, the API to gather data) are handled by the *Data Wrapper*. At the same time, one or more *Visualization* components can be used over the same *Data Wrapper* output, which allows displaying the same information in different views. For instance, a *Data Wrapper* is able to gather data from am SNMP Management Information Base (MIB), and to output information about `ifTable` entries (*e.g.*, `ifinoctets` and `ifoutoctets`). Such output can be formatted to a *Visualization* component that displays a bar chart (*e.g.*, with the total of `ifinoctets` and `ifoutoctets` for each port of a switch) or for another one that displays a line chart (*e.g.*, with the distribution of `ifinoctets` and `ifoutoctets` on a given switch port in specific time range).

(b) **Visualization -** It is a software component designed and developed with a well-defined input interface to receive data and, then, processing such data in order to display the visualization. As above-mentioned, *Visualization* components are agnostic in relation to the data source. For instance, let's suppose a *Visualization* component developed to display the network topology through a link-node representation. This component can be fed with data from a *Data Wrapper* component that gathers data from a network discovery tool, or from a *Data Wrapper* that queries the northbound API of an SDN controller to retrieving topology info. In both cases, the only requirement is the *Data Wrapper* output be compatible with the *Visualization* input.

Essentially, based on the explanation regarding the *Data Wrapper* and *Visualization* components, the most usual way to launch a visualization in the proposed solution is done by composing a *Data Wrapper* with a *Visualization* component. In next subsections, the explanation about how the selected outcomes of the Reuse Asset Management Process are materialized in the proposed approach. Section 3.2.1 presents the asset classification scheme whereas Section 3.2.2 introduces the asset storage and retrieval mechanism. Finally, Section 3.2.3 describes the asset usage recording.

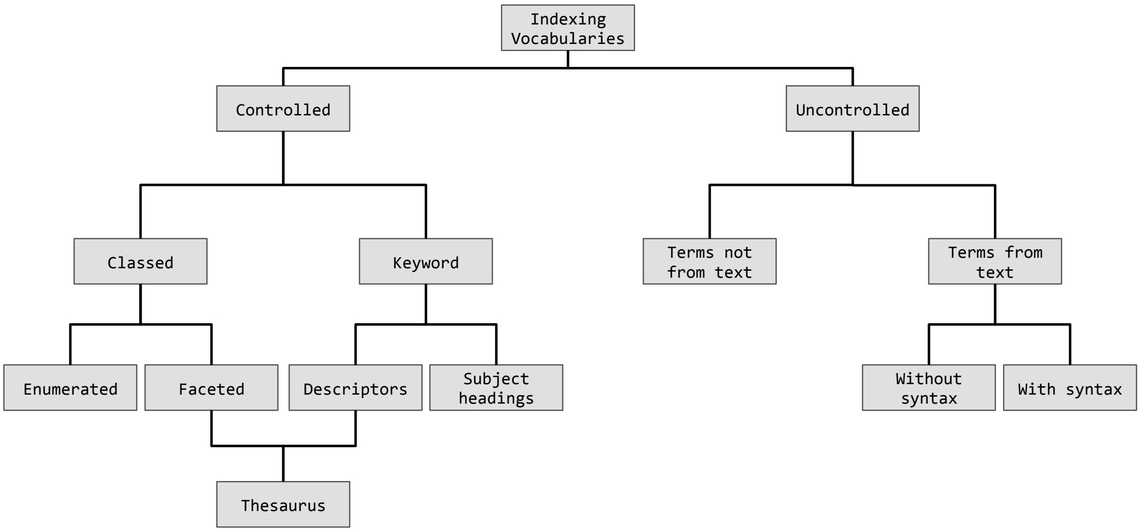
3.2.1 The Asset Classification Scheme

In the context of software reuse, the asset classification scheme is closely related to a Reusable Software Library (RSL). RSL is central to the practice of reuse by providing features such as storing, searching, and retrieving of reusable assets. This way, an RSL shall use a classification scheme mainly to support the searching of reusable assets. Frakes and Gandel (FRAKES; GANDEL, 1990) presented a taxonomy of library science that shows different methods investigated to classify assets for software reuse throughout the years (see Figure 3.3). Poulin and Yglesias (POULIN; YGLESIAS, 1993) describe the four major categories of asset classification scheme and highlight its trade-offs as follows:

- **Free-text -** It is the most simple method of classification, in which assets are described by individuals through free words or sentences. Although this approach can be partially automated by extracting, for example, text from the documentation, it has significant disadvantages such as ambiguity, hard to perform a search, and error-prone.
- **Enumerated -** In this type of classification, assets are organized into classes and hierarchically arranged through predefined lists (*e.g.*, like a table of contents in a book). This classification has the advantage that it is well-known for most people and then easy to adopt and use. The main disadvantages are: (*i*) indexes are built manually (*i.e.*, expensive and error-prone), (*ii*) it is susceptible to ambiguity, which makes difficult to locate assets; and (*iii*) the structure is typically not balanced (*i.e.*, there are many parts of one type and few of another).
- **Attribute-value -** In this classification scheme, assets are described by a pair $< attribute : value >$ where the attribute can be assigned with any value by the individual responsible for the classification. This approach has two advantages: (*i*) it is easy to use; and (*ii*) it can be partially automated. The main disadvantages are: (*i*) it requires more sophisticated search mechanism to avoid performance issues; and (*ii*) it also suffers from ambiguity problem.
- **Facets -** In a faceted classification, assets are described by a set of facets (*i.e.*, terms) and facet values, which makes this classification similar to the attribute-value method. However, the choice of facet values is not free, *i.e.*, the values are limited to a predefined range of allowable values. The main advantage of faceted classification is that it eliminates the ambiguity problem. The major disadvantage of this approach is that it is not designed to support semantic capabilities.

Throughout the years, it is possible to observe that the faceted classification scheme was more investigated and adopted to support RSLs than the other three methods presented above (FRAKES; POLE, 1994)(PRIETO-DÍAZ, 1991)(JONES; PRIETO-DIAZ, 1988). However, the faceted classification has an important disadvantage that it does not

Figure 3.3: Taxonomy of library science indexing methods that were investigated to classify assets for software reuse throughout the years



Source: (FRAKES; GANDEL, 1990).

support semantic capabilities. In essence, the lack of semantic takes to the following problem (HAPPEL et al., 2006). The relevant information of assets is isolated in multiple heterogeneous descriptions, covering different aspects. This way, such descriptions are less powerful because they are structured in a non-integrated manner that makes difficult, for example, the searching of assets.

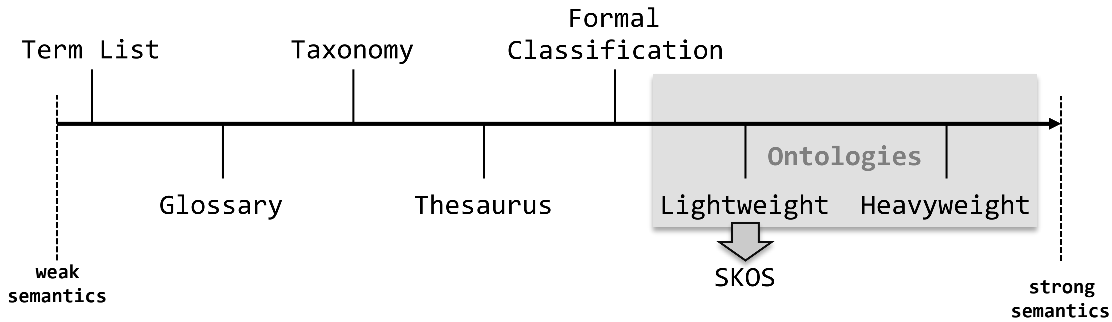
To cope with this issue, the proposed solution relies on Semantic Web concepts and uses the Simple Knowledge Organization System (SKOS) to codifying knowledge regarding reusable *assets* which also provides the classification scheme. SKOS is an official W3C recommendation that comprises a data-sharing standard, bridging several different fields of knowledge, technology, and practice (W3C, 2009). The motivation behind the SKOS is to provide a widely deployed standard for representing data of different families of Knowledge Organization Systems (KOS)[1] (*e.g.*, thesauri, classification schemes, subject heading systems, and taxonomies) and exchanging these data between computer systems (*i.e.*, machine-readable format).

Essentially, SKOS can be classified as a lightweight ontology in the semantic spectrum (see Figure 3.4) (DAVIES, 2010) because it provides softer semantic capabilities such as semantic search and annotation. Thus, SKOS lacks some of the benefits granted by heavyweight ontologies like formal axioms (*i.e.*, reasoning) and cleaning data (*i.e.*, high precision). On the other hand, SKOS is a way to allow a straightforward conversion

[1]Examples of KOS tools are common in the library and information sciences to organizing large collections of objects such as books or museum artifacts.

and reuse of existing knowledge to the Semantic Web technology context. In short, SKOS can be seen as a stepping-stone into Semantic Web and Linked Data.

Figure 3.4: Semantic spectrum adapted from Davies (DAVIES, 2010)



Source: by author (2016).

In the proposed solution, the SKOS data model to build up the asset classification scheme is primarily based on two taxonomies (see Appendix A): (*i*) a network and service management taxonomy defined in (CNOM, 2013); and (*ii*) an InfoVis taxonomy defined in (GUIMARÃES et al., 2016). These taxonomies provide the first-order topics for both domains (*i.e.*, network and service management and InfoVis). Moreover, they can be used as a starting point to refine concepts in more sub-levels. For example, the topic "*Internet Performance Measurement*" (which is analyzed in the first evaluation scenario, see Section 4.4) is an expansion of the concept "*Monitoring & measurements*" of the topic "*Methods*" in the network and service management taxonomy (see Table A.2). Likewise, the "*Star Plot*" is part of the concept "*Standard 2D/3D displays*" in the topic "*Techniques*" of the InfoVis taxonomy (see Table A.1).

3.2.2 The Asset Storage and Retrieval Mechanism

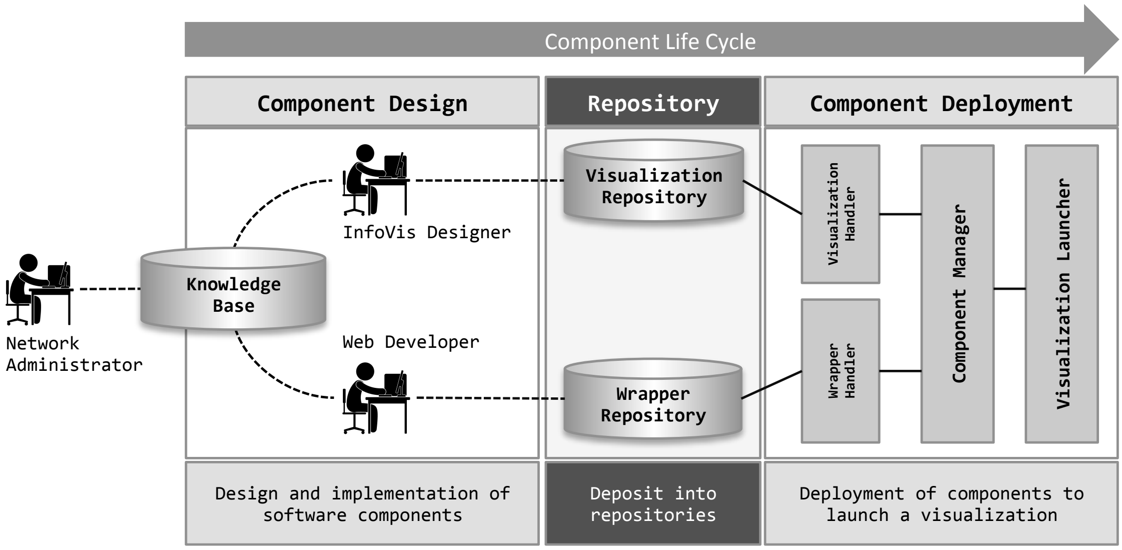
As previously introduced in Sections 2.2.2 and 2.2.3, the proposed approach relies on the CBD paradigm and ROA software architecture style. In this sense, the storage and retrieval mechanism of reusable assets (*i.e.*, *Data Wrapper* and *Visualization*) merges such concepts.

Essentially, reusable assets are software components implemented through RESTful Web services (*i.e.*, resources in the ROA architectural style). The development of such software components follows the component life cycle idealized by Lau *et al.* (LAU et al., 2013) (see Figure 3.5). The main strength of this approach, when compared with a standard CBD life cycle (KAUR; SINGH, 2010), is that the component development (*i.e.*, also known as "development for reuse") is treated as a separate process of the component-based system development (*i.e.*, also known as "development with reuse"). So, the component development life cycle is monolithic, more complete, and dissociated of the life cycle of a specific system. Figure 3.5 depicts the adaptation of component development life cycle idealized by Lau *et al.* (LAU et al., 2013) in the proposed approach. This life cycle consists of two phases named as designing and deployment phase. Here, these two phases are adapted to the needs of the proposed approach as follows:

- **Designing phase -** In this phase, components are designed and built in the form of a RESTFul Web Service and according to the domain requirements or knowledge and, then, recorded into a repository. For example, a *Data Wrapper* component is designed and developed by a *Web Developer* (*i.e.*, a stakeholder) to query the northbound API of an SDN controller in order to gather data about flow statistics. Once developed, this component is deposited into the *Wrapper Repository* (see Figure 3.1). In the same way, a *Visualization* component is designed and developed by a *InfoVis Designer* and deposited into the *Visualization Repository*.
- **Deployment phase -** In the component development life cycle idealized by Lau *et al.* (LAU et al., 2013), the deployment phase involves retrieving components from the repository and use them into a specific system under construction. Here, as highlighted in Section 2.2.2, components are developed in a black-box reuse fashion. Thus, the deployment phase retrieves components deposited into the repositories (through the *Component Manager* that invokes the *Visualization Handler* and *Wrapper Handler* components) in order to retrieve components and to build up and display a visualization for network and service management.

Based on the presented component development life cycle, the storage and retrieval mechanism works as follows. Reusable software components are identified and accessed through URIs, and their respective metadata and classification scheme are stored in a database (*i.e.*, into the *Visualization* and *Wrapper* repositories). Therefore, to retrieve a reusable software component, the client makes an HTTP request to the server by using a URI that identifies the Web Service of such a component. Afterward, the Web Service is responsible for interact with the *Component Manager*. According to the type of request (*e.g.*, to retrieve a *Data Wrapper* component), the *Component Manager* invokes the *Visualization Handler* or *Wrapper handler* module. These handlers are in charge of querying the metadata stored in the repository, and retrieving the requested reusable software com-

Figure 3.5: Adaptation of component development life cycle idealized by Lau *et al.* (LAU et al., 2013)

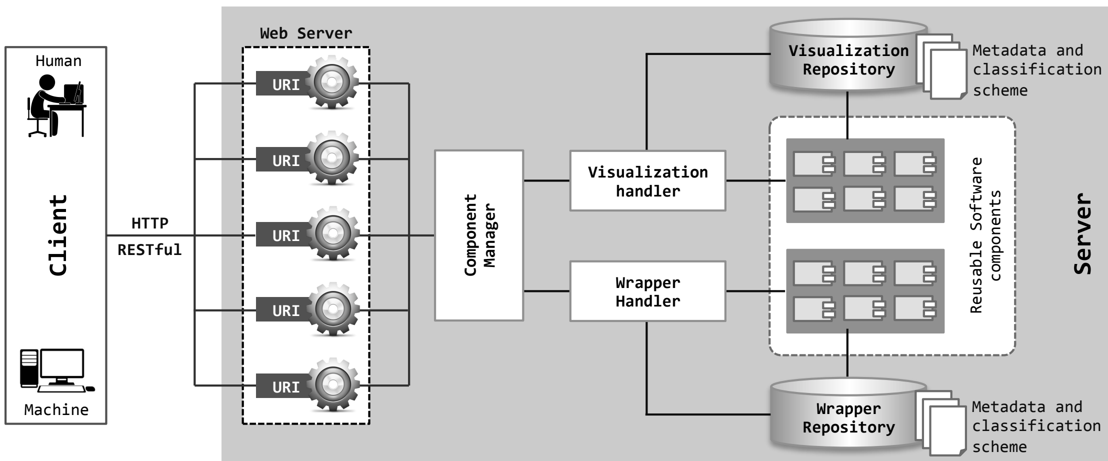


Source: by author (2016).

ponent. Finally, the software component is delivered to the client (*i.e.*, to the *Visualization Launcher* within the proposed architecture or an external client).

The mechanism described above is depicted in Figure 3.6. In summary, it enables the proposed approach for using a traditional Web server both for storing reusable software components as well as for implementing the ROA environment. Also, this ecosystem for storing and retrieving reusable components is useful in twofold: (*i*) it enables platform-independent access for humans through a Web browser (*e.g.*, when network administrators access the solution via Web Graphical User Interface (GUI) - see Section 3.3.2); and (*ii*) it is also machine-readable, allowing interaction with other systems (*e.g.*, to retrieve and display reusable visualization through mashups systems) in an easier way.

Figure 3.6: The storage and retrieval mechanism



Source: by author (2016).

3.2.3 Asset Usage Recording

To achieve a successful implementation of the Reuse Asset Management Process, the IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes (ISO/IEC/IEEE..., 2008) defines a set of activities and tasks. In this sense, the main task regarding asset usage recording is defined as follows: "*the asset manager shall keep track of each reuse of the asset and report to the domain engineer information about actual reuses of the asset*".

In the proposed approach, the usage of a reusable asset is recorded with the primary goal of supporting and facilitating the employment of visualizations by network administrators. For instance, a list of the most accessed reusable assets can help the network administrator to find a reusable asset promptly. In essence, there are four mechanisms to record the usage of reusable assets as shown below. The information provided for such mechanisms also aid in improving the knowledge base (see Figure 3.1).

1. **Users' log -** This mechanism refers mainly to logging the usage of reusable assets through the storage of users' activities. The information extracted from such log can help the network administrator by providing historical data regarding his/her individual usage of reusable assets as well as the usage of reusable assets by other network administrators. For instance, the network administrator can check the latest reusable assets accessed by him/herself, or the most accessed reusable assets comprehending all network administrators.

2. **Tagging -** It is a folksonomy-based mechanism (SEN; RIEDL, 2011) where network administrators can provide annotation related to the usage of reusable assets by means of unstructured textual labels, or simply tags. Tagging is an active research line in Human-Computer Interaction (HCI) field (*e.g.*, tagging of photo albums (AMES; NAAMAN, 2007)) and it facilitates many manipulations over tagged resources (here, reusable assets) such as indexing, browsing, and search (LIU et al., 2011). For instance, the network administrator can assign expressive terms to the reusable assets by using tags and then facilitate the discovery of assets. The tagging mechanism is also important to improve and update the vocabulary of the asset classification scheme (Section 3.2.1).

3. **Free-text annotations -** This mechanism allows network administrators to write free-text annotations by describing opinion/insight/advice about the reusable as-

sets. For instance, the network administrator uses free-text annotations to describe his/her insights about how a special reusable visualization help in his/her daily task. Although free-text annotations can be typically unstructured and semantically complex, there are current approaches (MOUSAVI et al., 2014) for text mining that allow to extracting rich information from free-text.

4. **Ranking -** This mechanism regards to a basic ranking system where the network administrator can give a grade that represents his/her level of satisfaction about the usage of a reusable asset. For example, after using a reusable asset, the network administrator can assign a quantitative grade (*e.g.*, ranging from 1 to 10) in which lowest values mean a lower level of satisfaction. Although straightforward, the metrics provided by such mechanism (*e.g.*, the average grade of a reusable asset) are helpful to facilitate, for example, the choice of a reusable asset by network administrators (as well as the ranking mechanism of mobile app stores helps users to choose an app).

3.3 Conceptual Architecture, Elements & Interactions

This section outlines the conceptual architecture and the interactions among its components, relying on the overview presented in Section 3.2 (see also Figure 3.1). Initially, Subsection 3.3.1 introduces the architecture tiers. Afterward, Subsection 3.3.2 describes the interactions among the architecture components.

3.3.1 Tiers

The conceptual architecture of the proposed approach is based on the three-tier Web application architecture, which consists of the presentation, logic and data tiers (see Figure 3.1). According to Voth *et al.* (VOTH; KINDEL; FUJIOKA, 1998), an architecture partitioned into these three tiers enhances scalability, reusability, security, and manageability. Here, this architectural style is implemented through the MVC design pattern, which has become a widely-adopted pattern for designing Web applications. Each tier is described below according to the functions presented by Huang *et al.* (HUANG; HE; MIAO, 2014).

- **Presentation Tier -** This tier has three primary functions: (*i*) admitting/denying requests performed by users (*e.g.*, network administrators) via a Web GUI (*e.g.*, to search for reusable software components in order to launch and display a visualization for network and service management), (*ii*) forwarding requests performed by users to the *Logic Tier* ; and (*iii*) receiving response from the *Logic Tier* and sends the response back to the user.
- **Logic Tier -** This tier orchestrates the business logic processing, and it also has three mainly functions, which include: (*i*) receiving requests from the *Presentation Tier*, (*ii*) processing the information received from the *Presentation Tier* by retrieving data from the *Data Tier* and applying proper business logic; and (*iii*) returning the processed information back to the *Presentation Tier*. In the proposed solution, the *Component Manager* and *Orchestrator* (see Figure 3.1) are examples of components placed in this tier.
- **Data Tier -** In essence, this tier handles the storage and management of persistent data (*e.g.*, the *Visualization Repository*, *Wrapper Repository*, *Knowledge Base*, and *Settings*). It is noteworthy that the *Data Tier* is not composed only of database systems (*i.e.*, Database Management Systems (DBMS)). For instance, as presented in Subsection 3.2.2, reusable software components (*e.g.*, *Visualization* components) are implemented via RESTful Web services. Thus, the *Visualization Repository* is divided into a DBMS that stores the metadata and the classification scheme and the Web server that stores the RESTful Web services.

3.3.2 Elements & Interactions

This subsection presents the explanation regarding components and their interactions in the proposed solution. In essence, it is used a top-down approach (*i.e.*, starting at the *Presentation Tier*) that is based on the role of the network administrator to conduct the explanation.

As a first step to reuse software components (*i.e.*, *Data Wrapper* and *Visualization*) for launching and displaying a visualization, the network administrator is encouraged to seek into the repositories (*i.e.*, *Visualization Repository* and *Wrapper Repository*) the components that fit with his/her management dataset. Here, it will be used as an example, the dataset investigated in the second case study presented in Chapter 4, Section 4.5. Then, let's suppose that the network administrator wants to visualize the traffic load on each port of the OpenFlow®[2] switches in an SDN-based network. To do that, the northbound API[3] of a Ryu[4] controller (in a SDN-based network, the controller is a logically centralized and software-based entity that controls the network) must be accessed to retrieve the stats of traffic load.

Initially, the network administrator interacts with Web GUI to enter login information and accessing the system. Once logged on, he/she is able to use the *Search* component (*i.e.*, in the *Presentation Tier*) by typing keywords of interest. Following the example above introduced, the network administrator could typing keywords like `SDN`, `Ryu`, `port`, `traffic load`, `switch` and so on. After the search query be submitted, the *Search* component calls the *Orchestrator* component (*i.e.*, in the *Logic Tier*), which triggers the *Component Manager* and *Data Loader*. In a nutshell, the *Component Manager* invokes the *Visualization Handler* and *Wrapper Handler* which consult, respectively, the *Visualization* and *Wrapper* repositories in order to retrieve information from the classification scheme. In its turn, the *Data Loader* queries the knowledge database by looking for information related to typed keywords.

Based on the data returned by the *Component Manager* and *Data Loader*, the *Orchestrator* creates a list of the available reusable software components that match the search query. In the proposed approach, the *Orchestrator* uses a simple content-based (*i.e.*, based on comparing terms in the classification scheme and annotations) and collaborative (*i.e.*, it checks the users' choices made in the past and the associated ratings) filtering strategy. Recent studies point out to use modern algorithms of recommender systems (BOBADILLA et al., 2013), but it is out of the scope of this thesis.

Resuming to the solution workflow, the list of the available reusable software components is returned by the *Orchestrator* to the *Search* component. Then, the *Search* component shows such list within the Web GUI. Thus, the network administrator is able to select the reusable software components that fit his/her requirements. Together with the list, there is a set of additional information (*e.g.*, ranking, free-text annotations, and tags) that aims at supporting the network administrator's choice. It is also worth noting that the list of reusable software components ensures that the suggested *Data Wrapper* and *Visu-*

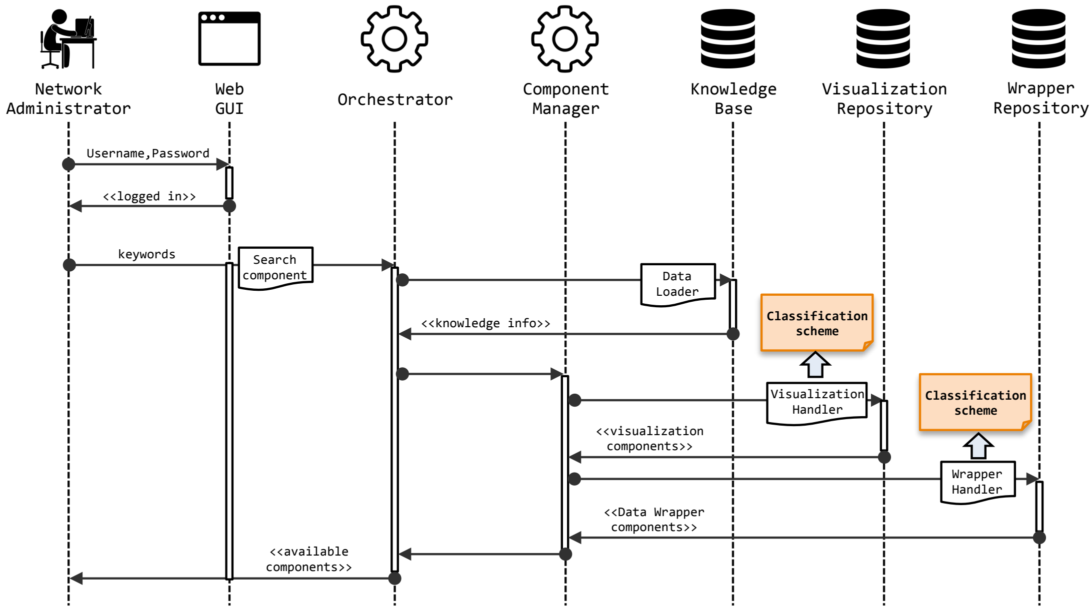
[2]OpenFlow®(https://www.opennetworking.org/sdn-resources/openflow) is the first standard communications interface defined between the control and forwarding layers of an SDN architecture.

[3]Currently, in an SDN-based network, the northbound API is an interface to access the controller.

[4]https://osrg.github.io/ryu/

alization components are compatible. Figure 3.7 depicts a sequence diagram that details the interactions explained until this point.

Figure 3.7: Sequence diagram for seeking to reusable software components

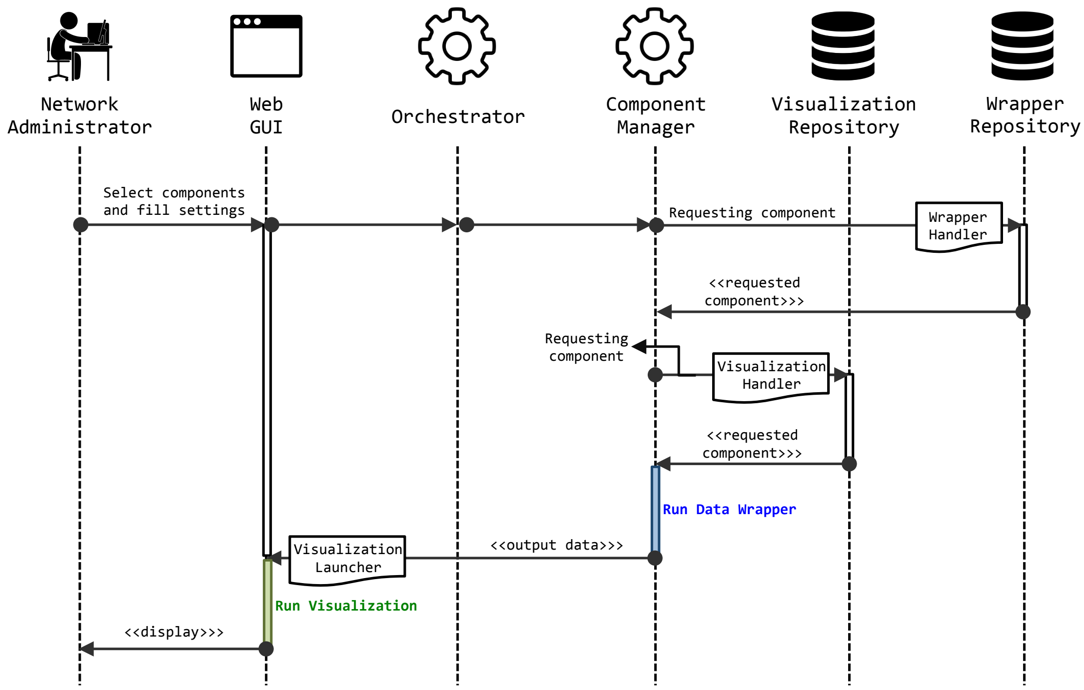


Source: by author (2016).

Once the reusable software components are selected by the network administrator, he/she can also set up some parameters. By using the proposed example, the *Data Wrapper* requires that the administrator inserts the address for accessing the northbound API of the Ryu Controller. Besides, let's suppose that the network administrator has selected a treemap view to display the traffic load of each port of OpenFlow®switches. Such visualization could allow, for example, configuring the canvas size (*e.g.*, by providing the width and height). Finished that configuration step, the *Orchestrator* component is invoked and it calls the *Component Manager* by passing the configuration settings provided by the administrator. After, the *Component Manager* is in charge of running the *Visualization Handler* and *Wrapper Handler* in order to load the selected components. In this point, the *Component Manager*, *Visualization Handler*, and *Wrapper Handler* work as presented in Section 3.2.2.

After the *Visualization* and *Data Wrapper* components have been properly loaded from their respective repositories, the *Component Manager* runs first the *Data Wrapper* component. It works like that because, in most cases, *Data Wrappers* requires low-level features such as system calls or database drivers. Thus, the *Data Wrapper* must be executed in the server-side to take advantage of the operational system capabilities and to avoid the exposure of the *Data Wrapper* source-code in the client (it is important to highlight that, although important, security issues are also out of the scope of this thesis). Upon the execution of the *Data Wrapper*, the *Component Manager* gathers the output data from the *Data Wrapper* and deliver it to the *Visualization Launcher* (in the *Presentation Tier*) together with the *Visualization* component (in the proposed approach *Visualization* components are designed and developed as Web-based apps, *i.e.*, they are performed in the client-side). The *Visualization Launcher*, in its turn, processes the data to launch and display the visualization itself in the Web GUI. Figure 3.8 depicts this pipeline.

Figure 3.8: Sequence diagram to select, configure, and run reusable components



Source: by author (2016).

In the Web GUI, together with the displayed visualization, there is the *Feedback* component. Through the *Feedback* component, the network administrator can tag, rank, and provide free-text annotation concerning visualizations (according to the concepts presented in Section 3.2.3). Information gathered by the *Feedback* component are sent to the *Orchestrator* component that calls the *Data Recorder* that updates the *Knowledge Base*. The *Settings* component performs basic functions such as the configuration of user profile and preferences and the storage of access log and users actions. Such information is sent to the *Backend* component that is responsible for the business logic and the interaction with the *Settings* database in the *Data Tier*. The *Settings* database also stores permissions and privileges to access the *Knowledge Base*, *Visualization Wrapper* repositories by the InfoVis Designer and Web Developer stakeholders.

3.4 Prototype Implementation

This section describes the prototype implementation of the proposed approach. Such a prototype has been implemented as a proof of concept and follows the design aspects of the conceptual architecture. In essence, the goal here is to present a brief overview of the design principles and the technology stack that drive the prototype implementation.

Most of the implementation of the prototype uses Web technologies and follows the MVC design pattern (as previously introduced) in a client-server communication model. The client-side interfaces (*i.e.*, the *Presentation* tier in the architecture and the *View* in the MVC pattern) uses a set of technologies related to the "Open Web Platform" (JACOBS; JAFFE; HEGARET, 2012). The primary goal was to build up user-friendly Web interfaces by the W3C standards. In this sense, the *View* layer is composed of a mix of patterns, technologies, and languages that form the myriad of new Web application trends. HTML and CSS have been used for structuring and formatting the interfaces. These technologies also enable the development of responsive interfaces to support modern mobile devices, such as smartphones and tablets. Specifically, the prototype uses the Bootstrap front-end framework. The JavaScript language has been used (in particular, the JQuery framework) to produce dynamic and interactive interfaces on the client-side. Using JQuery capabilities, Web server requests are performed using asynchronous requests, improving the user experience. Additionally, JQueryUI (the acronym for JQuery User Interface) provides a set of functionalities, such as interactions (*e.g.*, draggable, resizable), widgets, and effects.

Also in the client-side interfaces, the Data-Driven Documents (D3) visualization framework (BOSTOCK; OGIEVETSKY; HEER, 2011) has been used to implement *Visualization* components. D3 is a JavaScript library for manipulating documents based on data and provides a large set of visualization techniques (*e.g.*, parallel coordinates, force-directed graph, star plot, treemap, node-link tree). Additionally, D3 enables the development of interactive features (*e.g.*, zooming and brushing). JSON has been used to interchange data between the client-side and the server-side.

The server-side is implemented through a Linux-based distribution (CentOS) that runs an Apache Web server. The PHP language has been used to implement the business logic of the *Controller* and data retrieval performed by the *Model*. The *Visualization Repository* and *Wrapper Repository* are accessed through RESTful Web services by using the Flight micro-framework for PHP (a lightweight approach to building RESTful Web services). *Data Wrapper* components are developed in PHP and Shell Script. The data model that defines the asset classification scheme was written using the JSKOS, a JSON structure to encode SKOS.

3.5 Summary

This chapter presented an overview of the approach proposed in this thesis. First, it is worth noting that the proposed approach relies on three stakeholders. Such actors are encouraged to improving the repositories of *Visualization* and *Data Wrapper* with reusable software components and, then, to achieve the primary goal of this thesis: demonstrate that the reuse can improve productivity and decrease the cost when network administrators are adopting visualizations to help in everyday management tasks. Afterward, it was introduced the adaptation of three expected outcomes of the Reuse Asset Management Process defined in the international standard IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes. It is worth noting that the standard IEEE Std 12207-2008 does not provide low-level details concerning mechanisms, techniques, and technologies to achieve the expected outcomes of the Reuse Asset Management Process. Thus, this standard is generic enough to be adapted both by practitioners of software reuse as for other domains such as the proposed here.

As the last step, this chapter described the conceptual architecture that is based on a traditional three-tier Web application architecture. It is worthy of note that the use of a Web-based architecture is suitable because it is fully compliant with Semantic Web concepts and its technologies. Moreover, the Web environment is well-known and easy to use, since it requires only a Web browser and Internet connectivity. Such characteristics are also important for the stakeholders once the share of knowledge and consensus building is easier in a distributed environment like the Web.

4 EVALUATION

To show the feasibility of the reuse-based approach proposed in this thesis (*i.e.*, in terms of reducing cost and improving productivity), this chapter presents an evaluation that comprises two network management scenarios. First, Section 4.1 introduces the goal and an overview of the proposed evaluation. Next, Section 4.2 describes the methodologies used to estimate productivity (*i.e.*, through effort) and cost. Afterward, Section 4.3 details the four *Visualization* reusable software components designed and developed for this evaluation. Section 4.3 also shows the estimation of effort and cost of each *Visualization* component. Section 4.4, in its turn, presents the first assessment scenario, which relies on Internet performance measurements. Finally, Section 4.5 describes the second evaluation scenario, which uses a simulated SDN-based network.

4.1 Goal & Overview

The main goal of the evaluation presented in this chapter is to demonstrate that the reuse-based approach proposed in this thesis can significantly decrease cost (through saving time) and improve the productivity (in terms of consuming of time) of network administrators when adopting InfoVis techniques in daily tasks. To achieve this goal, two evaluation scenarios are defined. The first one uses Internet performance measurements datasets provided by FCC, the national regulator in the United States (US), and the Ofcom, the national regulator in the United Kingdom (UK) (see Section 4.4). The second explores a modern scenario of network management, specifically, an SDN-based environment (see Section 4.5).

Both evaluation scenarios employ a set of *Data Wrappers* and *Visualization* reusable software components (see Section 3.2) in order to display a visualization from a given management dataset. In this sense, it is possible to state that each assessment scenario is composed of a triple $[D_n, W_n, V_n]$ where: (*i*) D_n is the dataset (*i.e.*, the source of management data), (*ii*) W_n is the *Data Wrapper* component in charge of gather and process data from the dataset D_n and deliver the resulting information to the *Visualization* component V_n; and (*iii*) V_n is the visualization that displays the data delivered by the *Data Wrapper*

W_n. For each scenario, there was estimated the effort ($Effort$) in terms of consuming of time (in work hours) to build up a visualization VIS_n as depicted in Equation 4.1.

$$Effort(VIS_n) = Effort(W_n) + Effort(V_n) \tag{4.1}$$

In essence, Equation 4.1 corresponds to summing up the effort $Effort(W_n)$ to develop a *Data Wrapper* component W_n, and the effort $Effort(V_n)$ to develop the *Visualization* component V_n. As previously introduced, reusable software components are designed and developed for black-box reuse. Thus, the effort estimation can be done in isolation for each type of reusable software component (*i.e.*, *Data Wrapper* or *Visualization*). For this reason, the total effort $Effort(VIS_n)$ to build up a visualization VIS_n is estimated as described in Equation 4.1. Based on the effort, the cost is estimated (see Section 4.2.2).

4.2 Estimation Methodology

This section presents the estimation methodology employed in this evaluation. First, Section 4.2.1 shows the effort and productivity estimation to build up a visualization VIS_n, whereas Section 4.2.2 introduces the cost estimation.

4.2.1 Effort & Productivity Estimation

This thesis argues that building up a visualization VIS_n is primarily a software development task. In this sense, to estimate the effort and productivity regarding the building up of a visualization VIS_n, it is first necessary to estimate the size of a VIS_n using a software sizing estimation approach. There are different ways to estimating software size such as the analogy-based (*i.e.*, by comparing with a similar or of known size), regression-based (*i.e.*, by using regression analysis on historical data and input factors), expert judgment (*i.e.*, experts are consulted to estimate the software size), based on function points (*i.e.*, by examining the software characteristics), and parametric-based (*i.e.*, by using input values for parameters of selected software attributes) (FERENS, 1988).

To estimate software size, this thesis relies on the common principles of Functional Size Measurement (FSM) which is a step further of the Function Points (FP) and Function Points Analysis (FPA) methods originally designed by Albrecht (ALBRECHT, 1979).

FSM is a well-established approach in the software engineering field and widely used by the software industry (GENCEL; DEMIRORS, 2008). Moreover, the FSM methods are led by ISO/IEC (ISO/IEC, 2007) that standardizes, among other things, the fundamental concepts of FSM such as Functional User Requirements (FURs), Functional Size, Base Functional Component (BFC), BFC Type, and the FSM requirements that should be met by a candidate FSM method in order to be in accordance.

Nowadays, five FSM methods are certified by ISO/IEC as international standards as follows: (*i*) the International Function Points Users Group (IFPUG) (ISO/IEC, 2009), (*ii*) the Mark II (ISO/IEC, 2002), (*iii*) the Netherlands Software Metrics Association (NESMA) (ISO/IEC, 2005), (*iv*) the Common Software Metrics International Consortium (COSMIC) (ISO/IEC, 2011); and (*v*) the Finnish Software Metrics Association (FiSMA) (ISO/IEC, 2010). This evaluation adopts the COSMIC method for three main reasons: (*i*) it has generic applicability that comprehends software from different domains such as *data-rich* software (*e.g.*, software that needs to manage large amount of data), real-time software, and infrastructure software (*e.g.*, to support reusable components), (*ii*) it is used worldwide in various industries (EBERT; SOUBRA, 2014); and (*iii*) it is entirely open, *i.e.*, all method documentation is available in the public domain for free download.

The COSMIC measurement manual version 4.1.0 (COSMIC, 2015) was used to conduct the software size estimation. This manual is the COSMIC implementation guide for the standards defined in the ISO/IEC 19761:2011 (ISO/IEC, 2011). In essence, the COSMIC method measures software's FUR that results in a numerical "value of quantity" representing the functional size of software. It is important to highlight that the functional size measured through COSMIC method is independent of any implementation decision or operational artifacts. Such a characteristic allows, for example, to measure the software size before its implementation. So, this section introduces the main concepts of the COSMIC method briefly. The COSMIC measurement process consists of three phases as follows:

- The **Measurement Strategy** phase defines the purpose and scope of the measurement in order to enable future users of a measurement to decide how to interpret the measurement (*i.e.*, by specifying parameters of the measurement such as purpose, overall scope, functional users, and level of granularity).
- The **Mapping** phase aims at translating the FUR of the software to be measured into the key concepts of the COSMIC Generic Software Model (*e.g.*, *functional*

process, *data movement*, *data group*, *entry*, and *exit*). This translation produces the COSMIC model of the software that can be measured.

- The **Measurement** phase is the final step of the measurement process where the COSMIC unit of measurement is applied for assigning a size to the FUR of the software being measured.

The COSMIC unit of measurement is defined as 1 CFP (**C**osmic **F**unction **P**oint), which is the size of one *data movement*. A *data movement* is a base functional component (also named as sub-process) which moves a single *data group* (a unique set of data attributes that describe a single object of interest). A set of *data movements* is defined as a *functional process* that represents an elementary part of the FUR for the software being measured. A *functional process* shall comprise at least two *data movements*, and there is no upper limit to the number of *data movements* in a functional process. Moreover, there are four types of *data movement* defined by the COSMIC method which are described as follows:

- **Entry (E) -** A *data movement* that moves a data group from a functional user across the boundary into the functional process where it is required.
- **Exit (X) -** A *data movement* that moves a data group from a functional process across the boundary to the functional user that requires it.
- **Read (R) -** A *data movement* that moves a data group from persistent storage into the functional process that requires it.
- **Write (W) -** A *data movement* that moves a data group lying inside a functional process to persistent storage.

The COSMIC method defines the following principles to measure software size.

Principle 1. The size of a *functional process* FP is equal to the sum of its *data movements* DM as shown in Equation 4.2.

$$Size(FP) = \sum_{i=1}^{n} DM_i \tag{4.2}$$

Principle 2. The functional size FS of a part of software is equal to the sum of the sizes of its functional processes as shown in Equation 4.3.

$$Size(FS) = \sum_{i=1}^{n} Size(FP_i) \tag{4.3}$$

Principle 3. The total size TS of software in CFPs is equal to the sum of the functional size FS of each part of it, as shown in Equation 4.4.

$$TS = \sum_{i=1}^{n} FS_i \tag{4.4}$$

Based on these principles, the estimation of the total size TS of a visualization VIS_n in CFPs is defined by Equation 4.5.

$$TS(VIS_n) = TS(V_n) + TS(W_n) \tag{4.5}$$

Since the total size of a visualization is estimated, Equation 4.6 provides the total effort in work hours to build up VIS_n. In Equation 4.6 PDR means the Productivity Delivery Rate. PDR is defined in terms of hours per CFP, *i.e.*, the average number of hours spent to produce a CFP. In this evaluation, $PDR = 11$ hours per function point that corresponds to the average PDR for development team effort (MORRIS, 2016).

$$Effort(VIS_n) = TS(VIS_n) * PDR \tag{4.6}$$

4.2.2 Cost Estimation

Cost estimation for software development is not an easy task mainly because it meets a set of intangible aspects such as the application domain, the experience of developers, the complexity of the problem and its possible solution, the interaction among different sites, and so on. According to Pfleeger *et al.* (PFLEEGER; WU; LEWIS, 2005), cost estimation for software development is associated with decisions about: (*i*) *affordability* that includes the costs necessary to accomplish the development itself and other costs like training, repair, and upgrades over the software life cycle, (*ii*) *investment* that considers associated costs of a specific capability within the time and resources available; and (*iii*) *value* that takes into account other options to provide a more affordable or less risky investment to achieve the desired capability. Regarding cost-estimation approaches, there are different methods as shown below (PFLEEGER; WU; LEWIS, 2005).

- **Expert Judgment method -** In this method, human experts provide the cost estimation based on their experience. The main advantage of this approach is that experts can calibrate previous experience and data by observing the differences between previous projects and the current project. The disadvantage is that it may be very subjective because its heavy dependence on the judgment of the experts.

- **Analogy method -** It relies on a comparison and adjustment between data of a previous project and the current project. It also requires the expert opinion to adjust the costs of the previous project according to the requirements of the current project. The advantage of this method is that it uses historical data and does not rely only on the experts' judgment. The dependence of the expert judgment can be challenging and subjective placing the main disadvantage of this method.
- **Parametric/Algorithmic method -** These methods are based on research and analysis of historical data that uses some type of regression analysis to determine relationships between effort and some system attribute, such as the number of users, number of transactions, or required reliability. Accuracy, repeatability, and low subjectivity are the advantages of this method. The main disadvantages are: (*i*) the need of track a large amount of data to supply the models, (*ii*) the analyst must understand all attributes and their interdependence; and (*iii*) the calibration involves significant effort.
- **Bottom-Up/Work Breakdown structure method-** This method decomposes a project into smaller elements and estimates the element separately. The steps of decomposition and integration are the main advantages of this method since the estimation comprises many system-level tasks such as integration, documentation, project control, and configuration management that are often ignored by other methods. The disadvantage of this method is that it requires an extensive knowledge of what staff roles will be assigned and what management approaches will be used.
- **Top-Down method -** Unlike the bottom-up method, this method is based on general project features rather than on detailed functional or design characteristics. The advantages are the same of the bottom-up method. The disadvantage of this method is that it may be less accurate because the analyst can easily overlook important details (*e.g.*, low-level details and cost driver) that are clear in other methods.

Although the methods presented above take into account other aspects beyond the development effort in the software cost estimation, this evaluation estimates cost in isolation (*i.e.*, by using the development effort as the input parameter to determine cost). It is understood that for the purpose of the proposed assessment, the estimation of software size through the COSMIC method and the average value of PDR are enough to provide a suitable cost sizing. In this sense, uncertainties and risks associated with software costs (*e.g.*, costs of hardware, integration, and maintenance) are not addressed.

Equation 4.7 shows the formula used to estimate costs. The total cost $TCost(VIS_n)$ to develop a visualization VIS_n is given by multiplying the estimated effort $Effort(VIS_n)$ (see Equation 4.6) and the Person-Hour Value PHV. The value of PHV adopted here is the mid-range wage (in U$) of network administrators provided by the 2016 career reviews (see Table 4.1) (REVIEWS, 2016). Despite the fact that the proposed approach provides three types of stakeholders (see Section 3.1), the network administrator is the primary one. Thus, it is argued that the PHV of network administrators is enough to base the cost estimation proposed in this evaluation.

$$TCost(VIS_n) = Effort(VIS_n) * PHV \tag{4.7}$$

Table 4.1: The salary of network administrators according to the Top Ten carriers review 2016 (REVIEWS, 2016). This evaluation uses $PHV = U\$37.79$, *i.e.*, the mid-range salary.

	Per Year	**Per Month**	PHV
Highest salary (90th percentile)	U$ 115,180.00	U$ 9,598.33	U$ 59.98
Mid-range salary (50th percentile)	U$ 72,560.00	U$ 6,046.66	**U$ 37.79**
Lowest salary (10th percentile)	U$ 44,330.00	U$ 3,694.16	U$ 23.09

4.3 *Visualization* Components

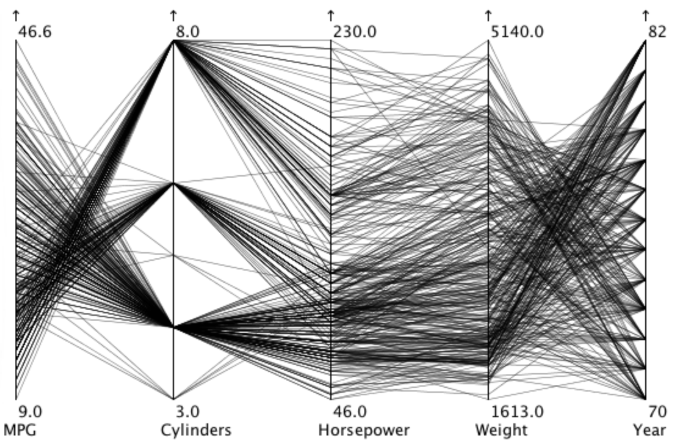
This section introduces the *Visualization* components used in the assessment scenarios (see Section 4.4 and 4.5). First, each employed InfoVis technique is briefly presented in the following order: parallel coordinates (Section 4.3.1), treemap (Section 4.3.2), star plot (Section 4.3.3), and network topology view (Section 4.3.4). After, this section summarizes the effort and cost estimation for each *Visualization* component accordingly to the methodology presented in Section 4.2.

4.3.1 Parallel Coordinates

Parallel coordinates (from now on referred to as V_1) (INSELBERG, 1985) is a well-known InfoVis technique that represents multiple dimensions without using orthogonal axes in Cartesian coordinates. In essence, parallel coordinates are constructed by placing axes in parallel on the embedding 2D Cartesian coordinate system in the plane (the parallel-coordinates domain) (HEINRICH; WEISKOPF, 2013). In this sense, each parallel axis represents an attribute and is linearly scaled within its data range. Each data item is represented by a polygonal line that intersects each parallel axes at the respective data value (see Figure 4.1). Although the orientation of the parallel axes is free, there are two common implementations that use either horizontal (*i.e.*, the axes are parallel to the x-axis) or vertical (*i.e.*, the axes are parallel to the y-axis) layout.

Figure 4.1 shows an application example of such a technique extracted from the eagereyes Website (KOSARA, 2016). This example uses a vertical layout where each parallel axis represents an attribute of car models released from 1970 to 1982 as follows (from the left to right): mileage (MPG), the number of cylinders (Cylinders), horsepower, weight, and year that the car was released. From this representation, it is possible to observe, for example, the indication of an inverse relationship between the mileage and number of cylinders. By analyzing the lines crossing the MPG and cylinders axes one can check that eight-cylinder cars usually have lower mileage than the six-cylinder and four-cylinder.

Figure 4.1: Example of parallel coordinates technique



Source: https://eagereyes.org/techniques/parallel-coordinates (2016).

Here, V_1 combines the parallel coordinates technique and a grid panel. In essence, the screen is horizontally divided into two parts as follows: (*i*) at the top, V_1 displays the interactive parallel coordinate view in a vertical layout; and (*ii*) at the bottom, V_1 displays the grid panel view. Each row of the grid panel details the data points by showing the value of their attributes in columns. Moreover, such views are integrated as described below:

- The parallel coordinates view provides a brushing interactive feature for each parallel axis. In essence, this interactive feature enables users to select a range of values of interest by using the mouse. Once the parallel coordinates view is updated according to the data selected by the user, the grid panel is also updated to display only the data range selected.
- When the user places the mouse over a row of the grid panel, the correspondent polygonal line in the parallel coordinates view is highlighted. This interactive feature allows users to analyze a particular polygonal line in more detail, in special for a large dataset in which the number of polygonal lines can clutter the view.

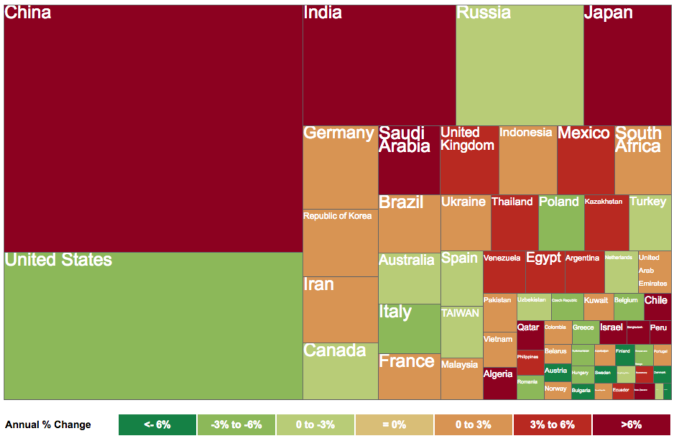
4.3.2 Treemap

The treemap visualization (from now on referred to as V_2) was proposed by Johnson and Shneiderman (JOHNSON; SHNEIDERMAN, 1991) for presenting hierarchical information. In short, they relied on the fact that there are several categories of information hierarchically structured such as directory structures, family trees, Internet addresses, and so on. Moreover, in most cases, these structures are large, which makes hard the understanding by humans when the data are not displayed in a pleasant visual representation (*e.g.*, humans have more difficulty in recognizing the spatial configuration of elements and the relationship between them in a plain text).

Essentially, a treemap view maps hierarchical information to a rectangular 2D display in a space-filling manner (JOHNSON; SHNEIDERMAN, 1991). In this sense, the tree nodes are encapsulated into the area of their parent node, and the size of a single node is determined proportionally about all other nodes of the hierarchy by a selected attribute (WIKI, 2012). Figure 4.2 shows a usage example of the treemap visualization technique. This treemap depicts the main responsible countries for emissions of CO2 fossil fuels in a worldwide context, in 2012. Each rectangle represents a country, and the size of rectangles means the amount of CO2 emission in million tons. For example, China is the number one in carbon dioxide emissions. Moreover, the color of each rectangle denotes the annual percent change accordingly to the caption at the bottom.

In this evaluation, V_2 divides the area into rectangles where the rectangles with the same color belong to the same parent (*i.e.*, the same category). The size of rectangles is calculated based on its values. For example, a rectangle represents a port of a given network switch and its size depicts the total traffic load (*e.g.*, the amount of transmitted

Figure 4.2: Example of a treemap view



Source: http://co2scorecard.org/countrydata/treemaps (2016).

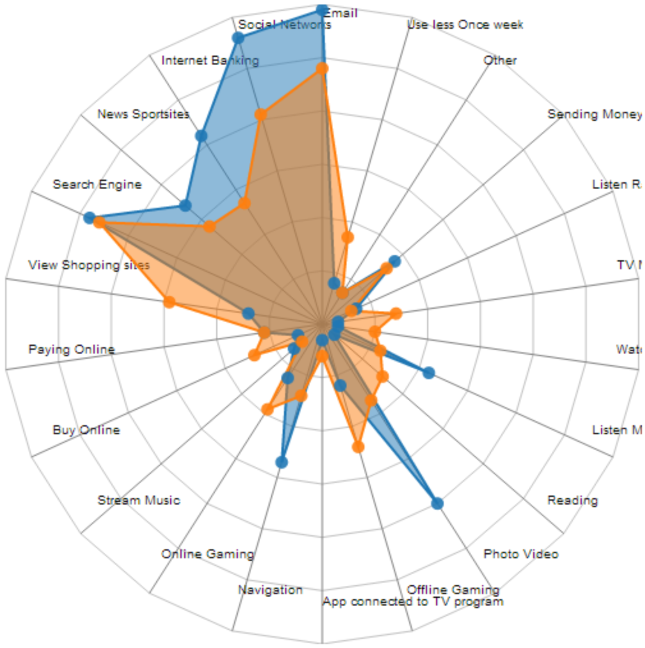
and received bytes). In addition, V_2 allows the user to see detailed information (*i.e.*, through an overlapped tooltip) when the mouse is over a rectangle. Also, a caption is presented above the view with the goal of differentiating each category.

4.3.3 Star Plot

Star plots (also known as radar chart or spider chart) (CHAMBERS et al., 1983) are a useful way to display multivariate data in the form of a two-dimensional chart with an arbitrary number of variables (*i.e.*, three or more quantitative variables). In summary, the star plot consists of a sequence of equiangular spokes where each spoke represents one of the variables. The data length of a spoke is proportional to the magnitude of the variable for the data point about the maximum magnitude of the variable across all data points. A line connects the data values for each spoke that gives the plot a star-like appearance (WIKIPEDIA, 2016). Figure 4.3 shows an example of the star plot where each spoke represents a given service (*e.g.*, email and social network) and its use by owners of smartphone and tablet. There are two data series represented in the chart, blue and orange. The blue one refers to the percentage of users that use smartphone whereas the orange is the percentage of users that use a tablet.

Here, visualization V_3 is a star plot where each data series is painted with a particular color (as presented in the example). There is also a legend within the chart that entitles each data series and shows its respective color to facilitate the readability by users. V_2 has two interactive features as follows: (*i*) the data series is highlighted when the mouse is over it (both when the mouse is over a particular point of the data series or the painted

Figure 4.3: Example of a star plot



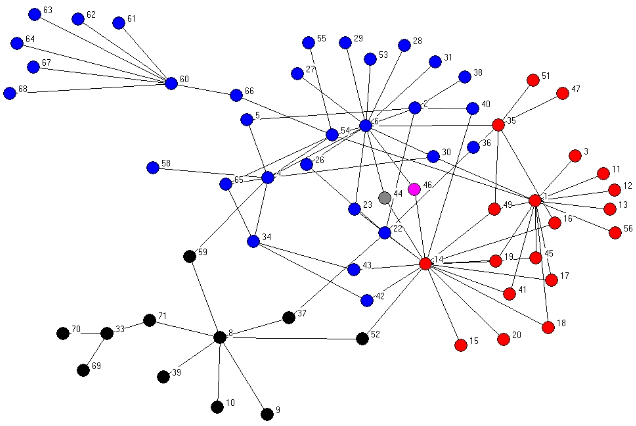
Source: http://www.visualcinnamon.com/2013/09/making-d3-radar-chart-look-bit-better.html (2013).

area); and (*ii*) a tooltip appears when the mouse is over a specific data point to show detailed information about it.

4.3.4 Network Topology View

In computer networks, the network topology can be defined as the representation and arrangement of elements that compose the network (*e.g.*, devices, links, services, and so on). In general, there are two main types of network topology: physical and logical topology. The physical topology represents the real placement of the network elements into the infrastructure (*e.g.*, the physical location of a switch or access point, cable installation, etc.). The logical topology, in its turn, depicts how the network elements are connected in order to provide the desired network capabilities (*e.g.*, packet forwarding, security and fault tolerance mechanisms, services, etc.) regardless of the physical design. Regarding InfoVis techniques for representing network topology, the node-link diagram (*i.e.*, graphs) is the most used technique. In this representation, nodes are the network elements (*e.g.*, devices) whereas the connections among such elements are the edges of a graph. Figure 4.4 depicts a general purpose example of a node-link diagram.

Figure 4.4: Example of a node-link diagram



Source: https://upload.wikimedia.org/wikipedia/commons/5/5a (2016).

In this evaluation, the node-link diagram technique (from now on referred to as V_4) was implemented for displaying the logical topology of computer networks. This visualization uses glyphs for representing the different types of devices (*i.e.*, nodes) that comprise the network (*e.g.*, switches, hosts, and servers). Nodes can be distinguished by color (*e.g.*, devices painted in yellow means a warning of the increase of incoming traffic). The color of links and their thickness are also customizable (*e.g.*, to highlight a broken connection). The floating menu provides a feature for calling other visualizations by clicking on an element of interest. For example, when such a feature is enabled, it is possible to click on a switch and visualize a simple dashboard with stats (*e.g.*, byte and packet count) of a specific port. Besides, it has interactive features such as add, edit and delete nodes and links, zooming, panning, and moving elements.

4.3.5 Effort and Cost Estimation

Based on the four InfoVis techniques previously described, this section introduces the effort and cost estimation for each one of them. Initially, accordingly to the COSMIC measurement manual, each developed *Visualization* reusable software component was decomposed into functional processes. Afterward, the *data groups* were identified, and the *data movements* accounted and distributed among *entries*, *exits*, *reads*, and *writes* (see Section 4.2.1). As the last step, the total size of each visualization ($TS(V_n)$) in CFPs was calculated through Equation 4.4. The documentation regarding COSMIC size measurement of each *Visualization* reusable software component is presented in Appendix C.

With the size estimation of *Visualization* Components in hands, the total effort was calculated based on Equation 4.6, *i.e.*, $Effort(V_n) = TS(V_n) * PDR$. Then, the total cost of each *Visualization* reusable software component was estimated through Equation 4.7. Table 4.2 summarizes the results of effort and cost estimation of each *Visualization* component.

Table 4.2: Results for effort and cost estimation of each *Visualization* component

V_n	**CFP**	$Effort(V_n)$ (hours)	$TCost(V_n) = Effort(V_n) * PHV$ (U$)
V_1	15	165	6,235.35
V_2	6	66	2,494.14
V_3	6	66	2,494.14
V_4	29	319	12,055.01

4.4 Scenario 1: Internet Performance Measurements

This section describes the first scenario analyzed in this evaluation. Essentially, this scenario relies on two management datasets (from now on D_1 and D_2) which are the result of Internet performance measurements performed in the United States (US) and United Kingdom (UK) to evaluate multiple broadband providers (fixed-line access). These measurements are obtained by using the SamKnows platform. In essence, the SamKnows company is specialized in the deployment of hardware-based probes (called as Whitebox) that are capable of running continuous measurements to assess broadband performance. The hardware-based probes are strategically placed (*i.e.*, in points of interest) within access networks and behind residential gateways. It is understood that such datasets fit with the proposed evaluation because they are plenty of management data (*e.g.*, download and upload speed, RTT, latency, and packet loss) that can be better analyzed through InfoVis techniques. Moreover, these publicly available datasets have actively been utilized in multiple studies (BAJPAI; SCHONWALDER, 2015).

First, Section 4.4.1 and 4.4.2 provide, respectively, a brief description of data provided by FCC (D_1), the national regulator in the US, and Ofcom (D_2), the national regulator in the UK. Next, Section 4.4.3 describes the four developed usage scenarios, highlighting the *Data Wrapper* components. Finally, the results and discussion regarding scenario are presented in Section 4.4.4.

4.4.1 Data from FCC

The data provided by FCC are the result of an ongoing nationwide performance study of broadband service in the United States, named as Measuring Broadband America (MBA), which is developed out of a recommendation by the National Broadband Plan to improve the availability of information for consumers about their broadband service (FCC, 2016). The MBA program is divided into two branches: (*i*) measuring fixed broadband; and (*ii*) measuring mobile broadband. This evaluation scenario uses data of the first one, *i.e.*, data from the Internet performance measurements of fixed broadband.

The FCC's MBA program makes available to stakeholders and the general public, among other things, the raw data collected from the measuring fixed broadband. To gather such data, after a call for volunteers for testing, SamKnows has selected 12,000 participants among the Internet Service Providers (ISP) of fixed broadband. Each selected participant receives a Whitebox that performs the test and collects the metrics presented in Table B.1 (see Appendix B). The FCC highlights that to maintain the integrity of raw collected data, measurements with invalid characteristics are discarded (*e.g.*, measurements outside the testing period and measurements in which the ISP changed).

All raw data collected through the MBA program are released by FCC periodically as part of a raw unaudited dataset. In essence, the FCC recognizes that this data can have value, for example, to the research community. Such data is divided into *tar.gz* files (one file per month) that can be directly downloaded from the FCC Website. Each *tar.gz* file contains a pair of files for each metric presented in Table B.1. For example, for the ICMP latency metric, there are two files as follows: <*curr_ping _2014_02.csv,curr_ping_2014_02.sql*>. The first one is a plain text file that contains the measurement data in a comma-separated format (*i.e.*, each value is separated by comma), where each line represents one measurement. The second is the data description (written in Structured Query Language (SQL)) of the data within the first file.

For this assessment scenario, raw data were parsed from the text file in a comma-separated format (*e.g.*, *curr_ping_2014_02.csv*) and stored in a relational database by using the data description written in SQL (*e.g.*, *curr_ping_2014_02.sql*). This approach enables to manipulate the raw data through SQL queries and then to build a *Data Wrapper* reusable software component that pulls data from a remote database.

4.4.2 Data from Ofcom

In the same way of FCC, Ofcom maintains a panel of residential fixed broadband users to enable it to monitor the performance of UK fixed-line broadband connections by using the SamKnows platform (OFCOM, 2016). Differently from FCC, Ofcom releases raw data of two measurement periods each year, which are based on tests run in the months of August and February. The collection methodology of raw data is described as follows:

- The averages are based on those respondents who have achieved five valid tests for each metric in each time period within the month.
- The ISP averages contain only respondents who live within specific geographic markets or in the Kingston-upon-Hull area, and within 5,000m from the local exchange.
- For the ISP averages, the only weights applied are distance related and apply to ADSL connections.

Ofcom's datasets provide the metrics presented in Table B.2 (see Appendix B). A dataset is released through a plain text file that contains the measurement data in a comma-separated format (*i.e.*, each value is separated by comma), where each line represents one measurement. Ofcom also provides additional information about the measurements in its raw data such as the name of the ISP, headline speed, and technology (this information are not present in the FCC's raw data). In this assessment, the datasets were kept in comma-separated format to enable the development of a *Data Wrapper* reusable software component that handles data from a static file in a pre-defined format.

4.4.3 Usage Scenarios

In this evaluation scenario, four usage scenarios were developed that use the datasets D_1 and D_2, comprehending two *Data Wrapper* reusable software components (W_1 and W_2) combined with two *Visualization* components, V_1 and V_3. Moreover, the four scenarios use measurement data taken along an entire month, specifically, in February 2014. This month was chosen because it is the period that raw data are available for the two datasets D_1 and D_2. Table 4.3 provides a brief description of each usage scenario (see also Figure 4.5).

Table 4.3: Usage scenarios

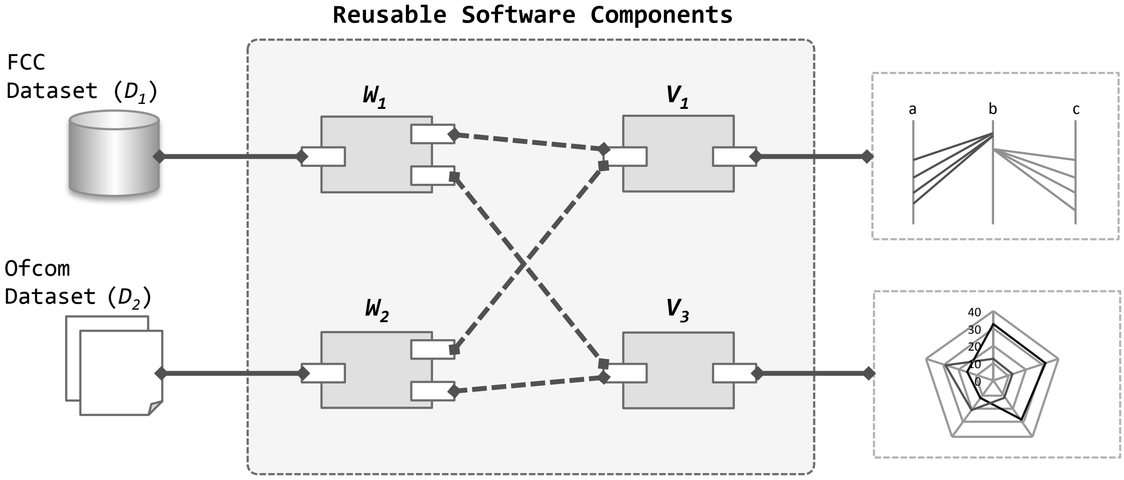
Scenario	Dataset	Data Wrapper	Visualization	Description
S_1	D_1	W_1	V_1	Using V_1 to display the average RTT, min RTT, max RTT, successes, and failures of the location with the highest average RTT.
S_2	D_1	W_1	V_3	Using V_3 to display the mean of the average RTT, min RTT, and max RTT for the top ten measurement points (*i.e.*, the ten locations with the highest mean of average RTT).
S_3	D_2	W_2	V_1	Using V_1 to display information collected from different measurement points as follows: the ISP, headline speed, download speed for 24h, max download speed, download speed between 8-10pm in weekdays, upload speed for 24h, max upload speed, upload speed between 8-10pm in weekdays, and technology (*e.g.*, ADSL and ADSL2+).
S_4	D_2	W_2	V_3	Using V_3 to show the average of download speed for 24h, the average of max download speed, and the average of download speed between 8-10pm in weekdays, of the ISPs that provide a headline speed of 20Mbps.

As shown in Table 4.3, two *Data Wrappers* W_1 and W_2 were developed for each usage scenario. Essentially, each *Data Wrapper* has an input interface to gathering data of each management dataset D_1 and D_2. Also, W_1 and W_2 were designed and developed to generating output data compatible with V_1 and V_3. This approach enables illustrating the reuse of W_1 and W_2 by different Visualization components as well as the reuse of V_1 and V_3 to visualize data from different datasets. An overview of how *Data Wrappers* W_1 and W_2 works is presented below.

- W_1 accesses a relational database that stores the dataset D_1 to retrieve data. Afterward, it processes such data according to the chosen output (which is an input parameter of W_1). For example, the user sets up the output format parameter to generate data for V_1. Thus, W_1 formats each data point and its attributes in a predefined format that is loaded by V_1 to display the parallel coordinates view and the grid panel. In essence, this component can: (*i*) connecting to a remote database, (*ii*) perform SQL queries; and (*iii*) formatting the returned data according to the selected output. It is noteworthy that W_1 takes advantage of performing SQL queries (through a query handler), which allows enhancements on data filtering.
- W_2 processes a static file (in comma-separated format) that stores the dataset D_2. In essence, W_2 is able to handle a plain text file where each line stores a data point,

and each attribute of the data point is comma-separated. After gather data from the file, this component formats the gathered data according to the chosen output (*i.e.*, parameterized to generate data for V_1 or V_2).

Figure 4.5: Combination of the reusable software components for the proposed usage scenarios



Source: by author (2016).

4.4.4 Results and Discussion

This section presents the effort and cost estimation for each usage scenario based on the methodology introduced in Section 4.2. First, it was carried out the estimation of the *Data Wrapper* components W_1 and W_2 (once the estimation of V_1 and V_3 was done and presented in Section 4.3.5). Accordingly to the COSMIC measurement manual, each designed and developed *Data Wrapper* reusable software component was decomposed into functional processes, and *data movements* accounted and distributed among *entries*, *exits*, *reads*, and *writes*. Thus, the total size of each *Data Wrapper* ($TS(W_n)$) in CFPs was calculated through Equation 4.4. Table 4.4 summarizes the results of effort and cost estimation of W_1 and W_2. Also, the documentation regarding COSMIC size measurement of W_1 and W_2 is presented in Appendix C.

Table 4.4: Results for effort and cost estimation of W_1 and W_2

W_n	**CFP**	$Effort(W_n)$ (hours)	$TCost(V_n) = Effort(W_n) * PHV$ (U$)
W_1	11	121	4,572.59
W_2	7	77	2,909.83

Figure 4.6 depicts the total effort of each scenario where *Data Wrappers* and *Visualization* software components are built up from scratch. In S_1, for example, the total effort corresponds to 286 hours. Considering that one network administrator works 160 hours per month, the effort spent in S_1 reaches more than one month of work (*i.e.*, more than one month and a half). The total effort estimated to S_3 is close to S_1, achieving 242 hours. S_2 and S_3 reached 187 and 143 hours respectively. Figure 4.7, in its turn, details the total cost accounting for each scenario. For example, from the cost accounting, one can infer that, in S_3 by reusing V_1, it is possible to save U$ 6,235.35 (*i.e.*, 165 working hours).

Figure 4.6: Effort estimation for each usage scenario

Source: by author (2016).

Here, the star plot of S_2 (see Figure 4.8) and the parallel coordinates view of S_3 (see Figure 4.9) are used to exemplify other benefits provided by a reuse-based approach. In both cases, the consensus building through knowledge sharing can facilitate not only the reuse but also the usage of visualizations by network administrators. The interpretation of the star plot and the parallel coordinates view is an example. The green and orange areas of the star plot (Figure 4.8) shows that the MAX RTT and MIN RTT of the top ten worst destinations have more variation than the AVG RTT. Such insight is obtained by analyzing the variation of the length of the spoke for each point in the same data group. Regarding the parallel coordinates view (Figure 4.9), the dark line in the view highlights an unexpected behavior when the coordinate "Technology" is selected on ADSL1 and ADSL2+. The measurements of download and upload speed exceed 65Mbps

Figure 4.7: Cost estimation for each usage scenario

Source: by author (2016).

and 18Mbps, respectively, whereas the headline speed is 8Mbps. In this view, the unexpected behavior can be promptly identified by analyzing the pattern of the other lines that are concentrated in the same range.

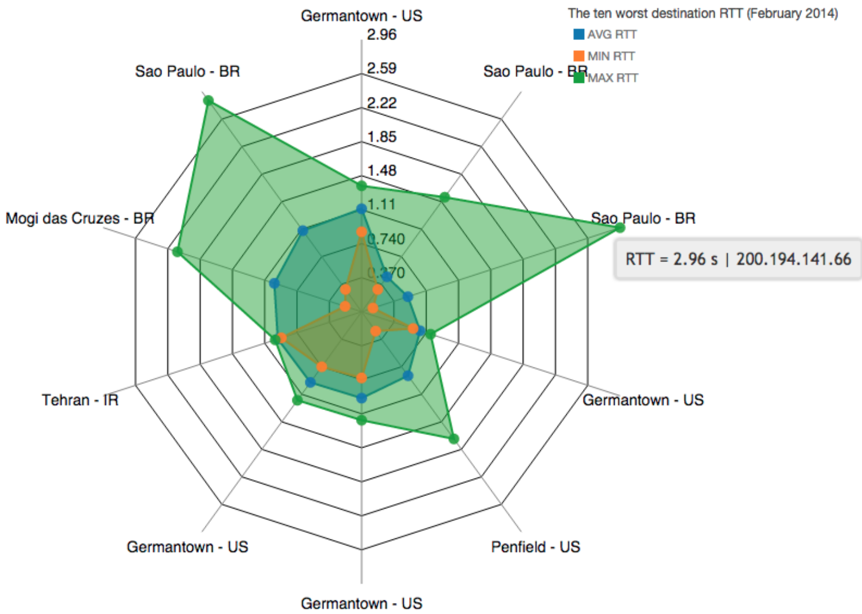
4.5 Scenario 2: SDN-based Network

This section describes the second scenario analyzed in this evaluation which relies on an SDN-based network. First, Section 4.5.1 provides an overview regarding the SDN paradigm and its major concepts. Next, Section 4.5.2 presents the SDN experimental environment that conducts to this evaluation scenario. Section 4.5.3 describes the usage scenarios. Finally, Section 4.5.4 introduces results and discussion.

4.5.1 Overview

Nowadays, the SDN paradigm has attracted attention from both academia and industry mainly because it is a way to overcome traditional network challenges such as the "Internet Ossification" (CHOWDHURY; BOUTABA, 2009). In SDN, the network control plane is decoupled from the forwarding plane (*i.e.*, data plane), making the control plane directly programmable (ONF White Paper, 2012). Thus, the network intelli-

Figure 4.8: Visualization V_3 is a star plot that displays data gathered from D_1 in S_2. The tooltip depicts the higher MAX RTT (2.96s) that was measured for the IP address 200.194.141.66.



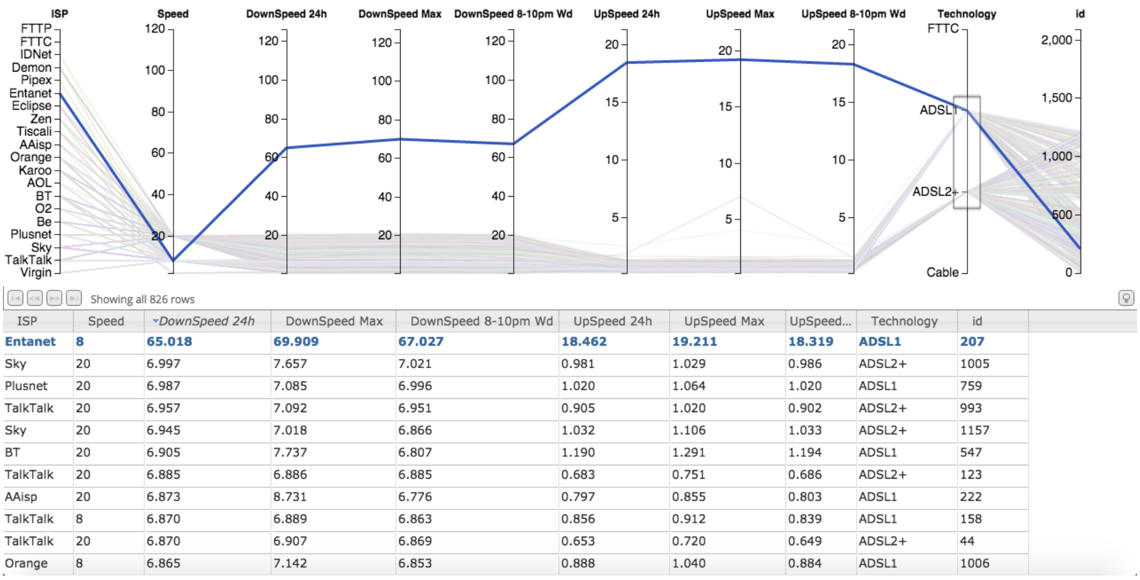
Source: by author (2016).

gence is logically centralized in software-based controllers, and the network devices (*e.g.*, switches) perform packet forwarding based on rules installed by the controllers (NUNES et al., 2014). In summary, it is possible to state that SDN has two major characteristics as follows (FEAMSTER; REXFORD; ZEGURA, 2014):

- The separation of the control plane (that defines how to handle the traffic) from the data plane (that forwards traffic according to the rules installed by the control plane).
- The consolidation of the control plane where a software-based program controls multiple network elements in the data plane through a predefined API.

The current literature has also introduced the concepts of *southbound* and *northbound* APIs (KIM; FEAMSTER, 2013). The first one refers to the interface between the SDN-capable devices (*e.g.*, switches) in the data plane, and the software controller in the control plane. The second defines how to describe network policies, and how to translate such policies into an understandable form to the control plane (*i.e.*, to the controller). Figure 4.10 presents a general architecture of an SDN-based network which is composed of four major planes (*i.e.*, data, control, application, and management planes) and the *southbound* and *northbound* APIs.

Figure 4.9: Visualization V_1 displays data collected from D_2 in S_3. The line highlighted in the parallel coordinates shows an unexpected behavior when the coordinate "Technology" is selected in ADSL1 and ADSL2+. The grid panel shows the nominal values of each variable (parallel axis) for each measured point (*e.g.*, the column "Speed" contains the headline speed of each measured point).



ISP	Speed	DownSpeed 24h	DownSpeed Max	DownSpeed 8-10pm Wd	UpSpeed 24h	UpSpeed Max	UpSpeed...	Technology	id
Entanet	**8**	**65.018**	**69.909**	**67.027**	**18.462**	**19.211**	**18.319**	**ADSL1**	**207**
Sky	20	6.997	7.657	7.021	0.981	1.029	0.986	ADSL2+	1005
Plusnet	20	6.987	7.085	6.996	1.020	1.064	1.020	ADSL1	759
TalkTalk	20	6.957	7.092	6.951	0.905	1.020	0.902	ADSL2+	993
Sky	20	6.945	7.018	6.866	1.032	1.106	1.033	ADSL2+	1157
BT	20	6.905	7.737	6.807	1.190	1.291	1.194	ADSL1	547
TalkTalk	20	6.885	6.886	6.885	0.683	0.751	0.686	ADSL2+	123
AAisp	20	6.873	8.731	6.776	0.797	0.855	0.803	ADSL1	222
TalkTalk	8	6.870	6.889	6.863	0.856	0.912	0.839	ADSL1	158
TalkTalk	20	6.870	6.907	6.869	0.653	0.720	0.649	ADSL2+	44
Orange	8	6.865	7.142	6.853	0.888	1.040	0.884	ADSL1	1006

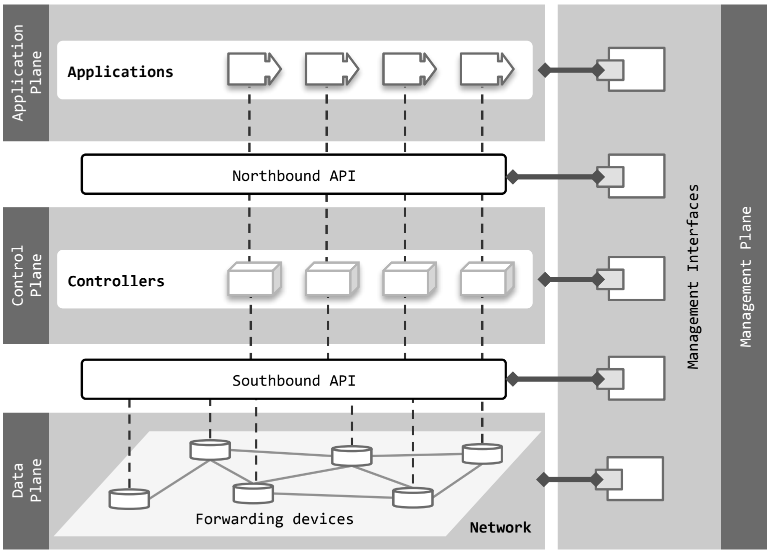
Source: by author (2016).

Given the above-presented characteristics, one of the central promises of SDN is to allow flexible and efficient network management via software-based programs. However, this promise is not a reality yet. In this sense, Wickboldt *et al.* (WICKBOLDT et al., 2015) introduce a set of requirements to foster the adoption and development of the SDN management. The requirements are bootstrap and configuration, availability and resilience, network programmability, performance and scalability, isolation and security, flexibility and decoupling, network planning; and monitoring and ***visualization***. Although most of such requirements are not new and have been investigated in the scope of traditional networks, they must be revisited to encompass the SDN paradigm and, then, to achieve the promise of flexible and efficient management.

In this evaluation scenario, this thesis takes into account the ***visualization*** requirement. The goal is to check how the proposed reuse-based approach can be effective to reduce cost and to enhance the productivity of network administrators when adopting of InfoVis techniques for SDN management. In essence, the northbound APIs of two current SDN controllers (Floodlight[1] and Ryu[2]) were investigated, and specific *Data Wrappers* components were designed and developed to gather and process data made available by SDN controllers.

[1]http://www.projectfloodlight.org/
[2]https://osrg.github.io/ryu/

Figure 4.10: General architecture of an SDN-based network



Source: by author (2016).

4.5.2 Experimental Environment

In order to conduct the evaluation of this scenario, it was designed and developed an experimental SDN environment which is based on a university campus network. The emulated network topology is depicted in Figure 4.11, and it is composed of:

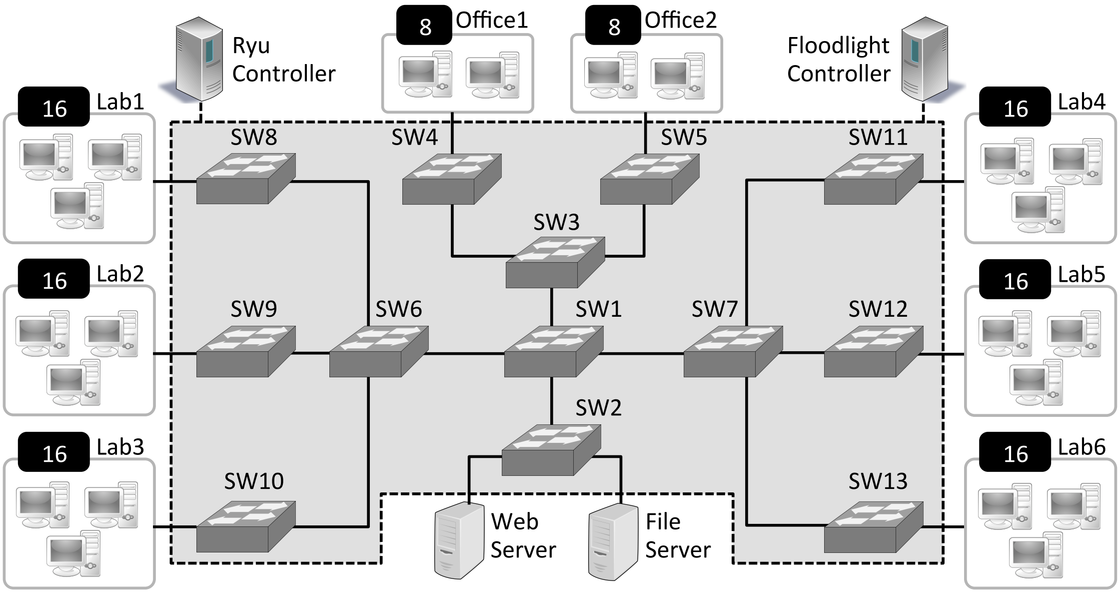
- 13 OpenFlow®switches that connect 112 hosts from 6 laboratories (each laboratory has 16 hosts) and two administration offices (each administration office has eight hosts).
- One Web server which is responsible for providing Web pages like the university website, virtual learning environment, and academic registration system.
- One FTP server that behaves like a repository of large files, making available to download, for example, iso of operating systems.
- One SDN controller. Although Figure 4.11 shows the two SDN controllers employed in this evaluation scenario, each usage scenario (see Section 4.5.3) uses only one of them (*i.e.*, the topology is the same changing only the controller).

In a nutshell, it was used a simulation of real traffic of HTTP and FTP services between hosts (within the labs and administration offices) and servers (*i.e.* the Web server and File server). The HTTP traffic is characterized by all hosts accessing the Web Server to request Web pages. Such Web pages vary in size according to the average page weight provided by the HTTP Archive Report[3] (*i.e.*, between 1.5MBytes and 2MBytes). The FTP

[3]http://httparchive.org/

service was used to generate traffic by transferring large files (*e.g.*, an iso with 8GBytes) between hosts of interest and the FTP server. The evaluation topology was created using the Mininet[4] emulator. The experiments itself were performed in one 2.6 GHz Intel Core i5 with 8 GB 1600 MHz DDR3 of RAM.

Figure 4.11: Experimental SDN-based network topology



Source: by author (2016).

4.5.3 Usage Scenarios

In this evaluation scenario, three usage scenarios were developed (S_5, S_6, and S_7), comprehending three *Data Wrapper* reusable software components (W_3, W_4, and W_5) combined with three *Visualization* components, V_4, V_2, and V_1, respectively. Here, the datasets are obtained through requests to the northbound API of the employed SDN controllers (Floodlight and Ryu). From now on, data from the Floodlight controller is referred as D_3 and data from the Ryu controller as D_4. Table 4.5 provides a brief description of each usage scenario (see also Figure 4.12).

Overall, the *Data Wrapper* developed for each usage scenario (*i.e.*, W_3, W_4, and W_5) has an input interface to gathering data from a specific SDN controller (*e.g.*, W_5 can handle the Floodlight controller) (see Figure 4.12). In short, each *Data Wrapper* reusable software component uses the REST API provided by the controllers to retrieve the desired data. For example, the URI `/wm/core/controller/switches/json`

[4]http://mininet.org/

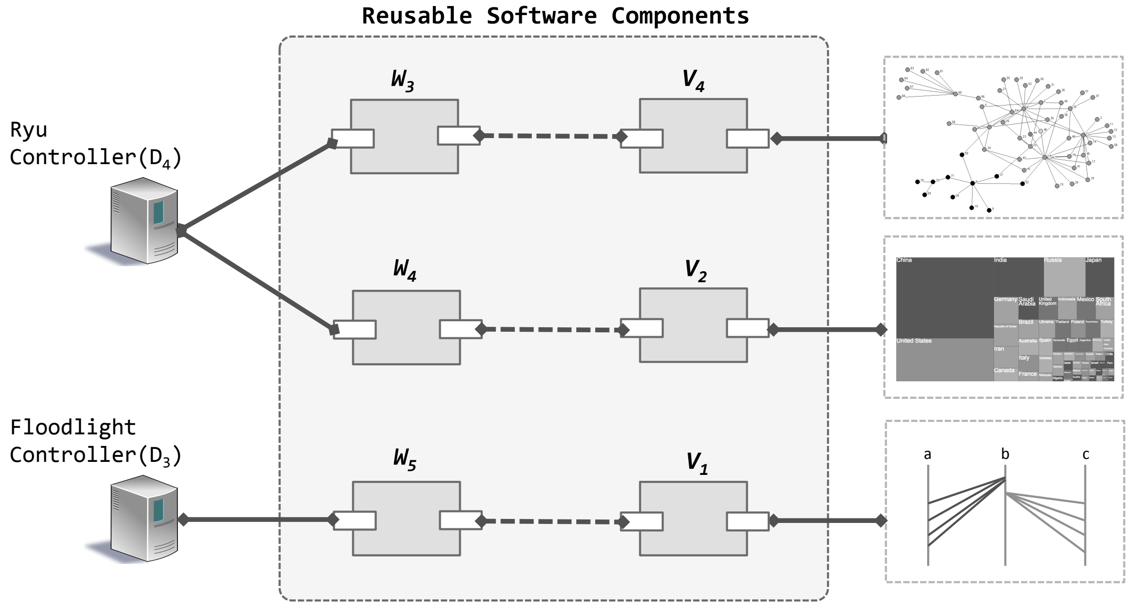
Table 4.5: Usage scenarios

Scenario	Dataset	Data Wrapper	Visualization	Description
S_5	D_4	W_3	V_4	Using V_4 to display the logical network topology and the simple dashboard with stats about a switch of interest.
S_6	D_4	W_4	V_2	Using V_2 to display the traffic load (*i.e.*, bytes and packets) in each port of the switches that compose the topology.
S_7	D_3	W_5	V_1	Using V_1 to display information about aggregate traffic (*i.e.*, bytes and packets) in each switch of the topology.

in the Floodlight REST API allows getting a list of all switch connected to the controller in JSON format. The same is possible by using the module `ryu.app.ofctl_rest` (*i.e.*, the REST API) of the Ryu controller.

It is also worth of noting that W_5 was designed and developed to generating output data compatible with V_1. This approach allows to illustrating, for example, the reuse of V_1 (that could be employed through an ideal black-box reuse, *i.e.*, without any code intervention) in a different scenario of management, and together with another *Data Wrapper* component.

Figure 4.12: Combination of the reusable software components for the proposed usage scenarios



Source: by author (2016).

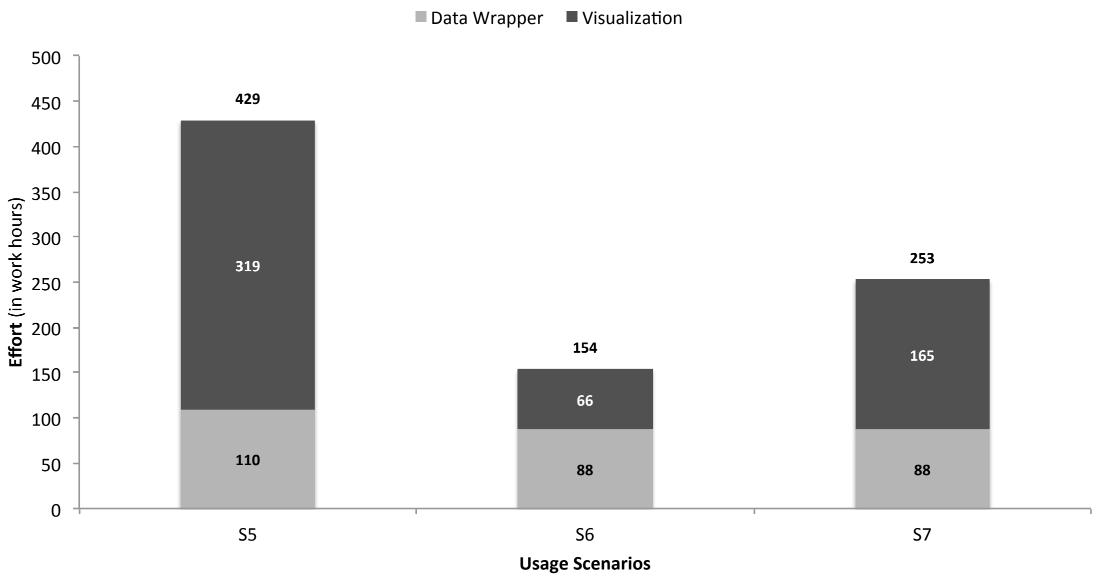
4.5.4 Results and Discussion

This section presents the effort and cost estimation for each usage scenario based on the methodology introduced in Section 4.2. First, it was carried out the estimation of the *Data Wrapper* components (W_3, W_4, and W_5) where the total size of each component ($TS(W_n)$) in CFPs was calculated through Equation 4.4. Table 4.6 summarizes the results of effort and cost estimation of W_3, W_4, and W_5. Also, the documentation regarding COSMIC size measurement of such components is presented in Appendix C.

Table 4.6: Results for effort and cost estimation of W_3, W_4, and W_5

W_n	**CFP**	$Effort(W_n)$ (hours)	$TCost(V_n) = Effort(W_n) * PHV$ (U$)
W_3	10	110	4,156.90
W_4	8	88	3,325.52
W_5	8	88	3,325.52

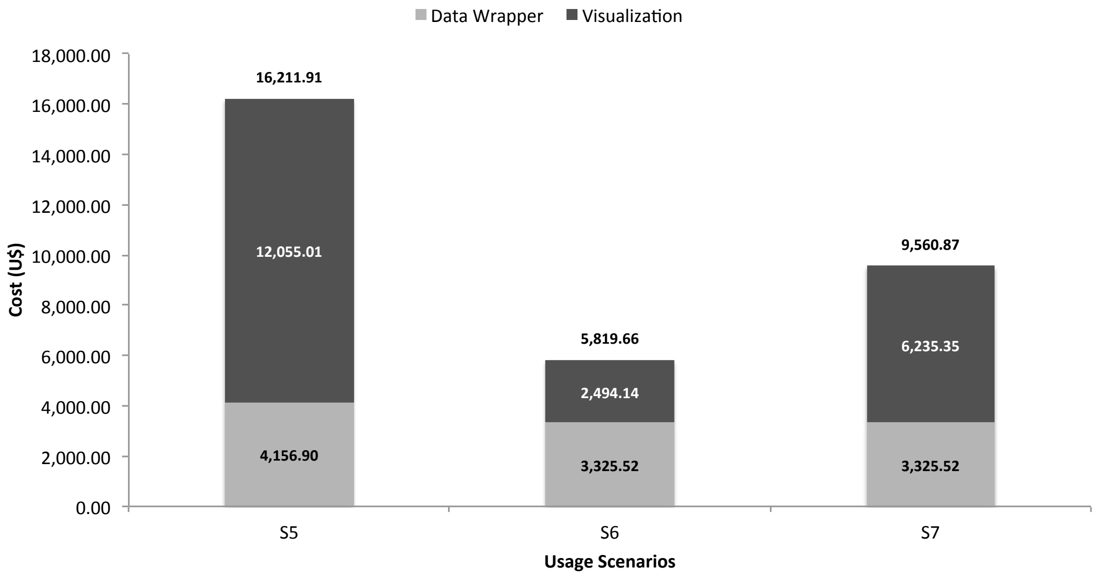
Figure 4.13: Effort estimation for each usage scenario



Source: by author (2016).

Figure 4.13 depicts the total effort whereas Figure 4.14 the total cost of each scenario taking into account that *Data Wrapper* and *Visualization* reusable software components are built up from scratch. The first aspect to be highlighted here is the result to S_7 concerning the effort and cost. The sum of the total effort and total cost in this usage scenario achieves 253 work hours and U$ 9,560.87, respectively (when developed from scratch). However, V_1 can be reused in S_7 which reduces the total effort and total cost in $\simeq 65\%$. In this case, as the same as occurs in S_3 (see Section 4.4.4), the effort reduction

Figure 4.14: Cost estimation for each usage scenario



Source: by author (2016).

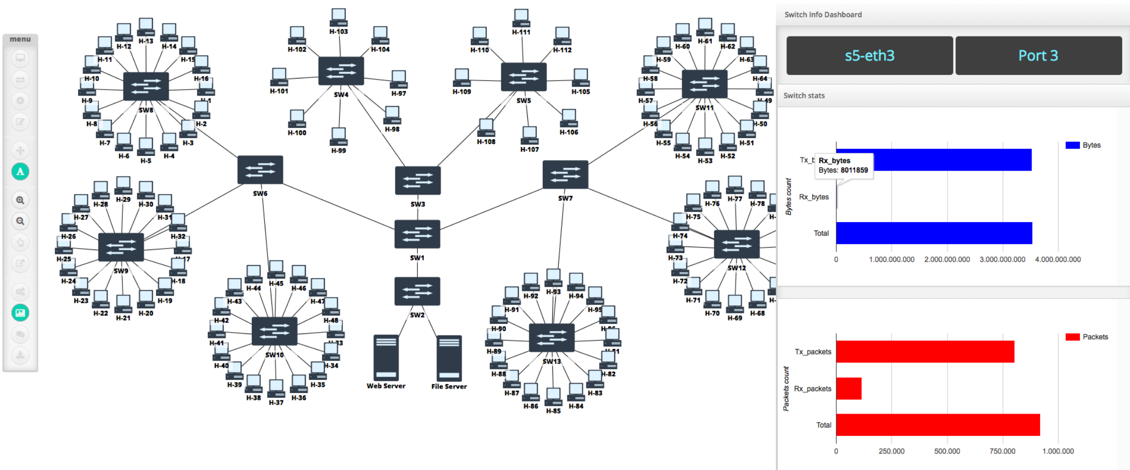
represents a month of work (considering that one network administrator works 160 hours per month).

Regarding the usage scenario S_6, it is possible to identify that the effort and cost for building up the *Data Wrapper* component are bigger than to the effort and cost for building up the *Visualization* one. On the other hand, the building of knowledge and consensus from the development of W_4 can generate indirect effort and cost savings. For instance, the simple dashboard of V_4 displays the traffic load of a switch port (*i.e.*, the amount of transmitted and received bytes and packets) through bar charts. Then, let's suppose that the bar chart is also a *Visualization* component. For retrieving the traffic load data and fill the *Visualization* bar chart component it is possible to envisage two approaches: (*i*) build up a new *Data Wrapper* component; or (*ii*) add an output interface in W_4 compatible with the input interface of the *Visualization* bar chart component. In the former, although built up from scratch, the developer can take advantages of the knowledge embedded in W_4 (*e.g.*, by analyzing the source code). In the second case, the effort and cost are reduced to the work of extending W_4 (*i.e.*, by adding an output interface).

The usage scenario S_5 is the most expensive since the *Data Wrapper* and *Visualization* component add up 429 work hours of effort, and U$ 16,211.91 of cost. It is mainly explained by the complexity of the *Visualization* component V_4. Nevertheless, once the W_3 and V_4 are designed and developed, it is possible to foresee opportunities of reuse that can also be effective to reduce cost and improve productivity. For example, the floating menu of V_4 can be extended to adding a button that makes an external call to the W_4 and V_2 components. In this way, V_4 could be easily integrated with the view presented in S_6 by reusing the W_4 and V_2 components. The same could be done with the view displayed in S_7 (*i.e.*, reusing W_5 and V_1) since it shows the aggregate traffic in each switch of the topology.

The InfoVis techniques of S_5 (see Figure 4.15), S_6 (see Figure 4.16), and S_7 (see Figure 4.17), are also used here to exemplify that the background knowledge can support network administrators when adopting visualizations for everyday tasks. The treemap view (S_6) enables administrators to visualize detailed information by clicking on a point of interest. Figure 4.16 highlights the comparison between the traffic load in s5-eth3 (Office2) and s9-eth4 (Lab2) because the experiment has simulated the download of a large file from the File server to the host on the port s5-eth3. Thus, the total traffic of s5-eth3 is bigger than the others while s9-eth4 follows the expected load for Labs and Offices. Such insight can be obtained by observing the size of the rectangles.

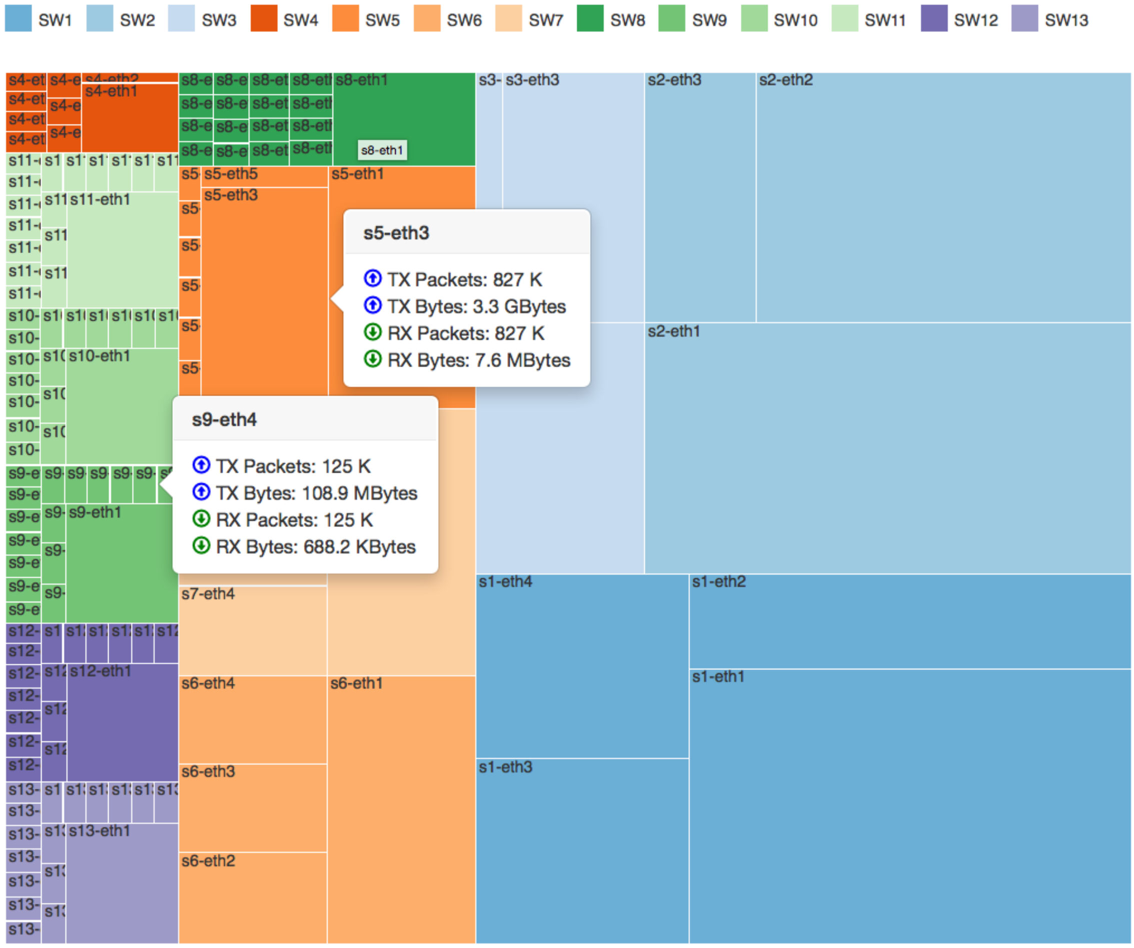
Figure 4.15: Visualization displayed in the usage scenario S_5



Source: by author (2016).

The parallel coordinate (S_7) shows information about aggregate traffic in each switch of the topology. Figure 4.17 shows only the switches of Labs. With this visualization, the network administrator can promptly identify that SW8 (Lab1) has approximately ten times more aggregate traffic than the others switches (see the dark line in the parallel coordinates system and the highlighted row in the data grid). In S_5, the topology view helps the network administrator to understand the network topology. Moreover, Figure 4.15 shows the topology and the simple dashboard that displays information about s5-eth3 that could be used to complement the visualization presented in S_6. This visualization uses different glyphs to display each type of device (*e.g.*, switch, host, and server).

Figure 4.16: Visualization displayed in the usage scenario S_6



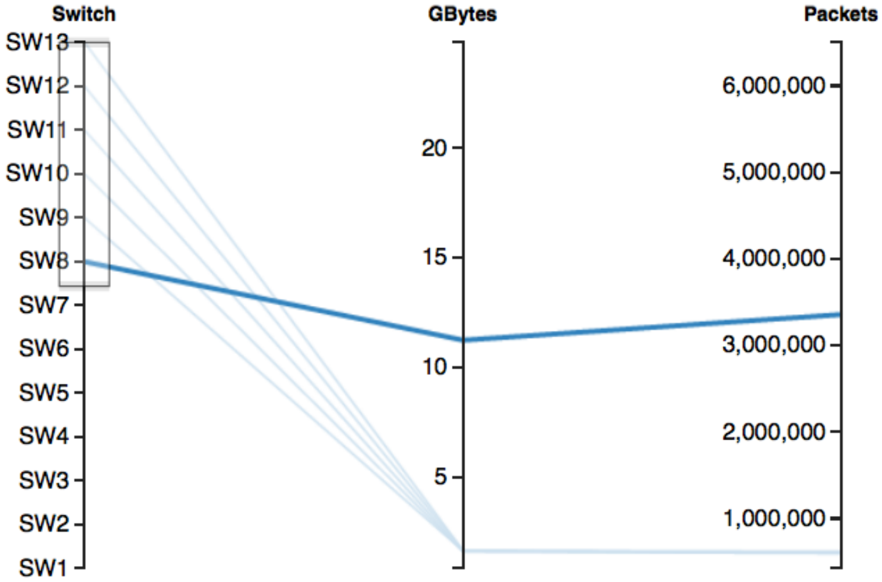
Source: by author (2016).

It also has interactive features like zooming and moving elements. Such features allow the network administrator, for example, to organize elements according to his/her way (unlike the topology view available in the Floodlight Web UI which is based on a simple force-directed graph, for example).

4.6 Summary

This chapter introduced an evaluation that demonstrates that the reuse-based approach proposed in this thesis can significantly decrease cost and improve the productivity of network administrators when adopting visualizations in daily tasks. In short, this evaluation advocates that building up a visualization VIS_n is a software development task. Based on that, the estimation of effort (in work hours) to build up a visualization VIS_n is achieved by applying the COSMIC software sizing method that relies on the common principles of FSM. In essence, a visualization VIS_n is divided into two reusable software

Figure 4.17: Visualization displayed in the usage scenario S_7



Showing all 6 rows

Switch	GBytes	Packets	id
SW8	**11.25**	**3357999**	**7**
SW9	1.67	610348	8
SW10	1.65	614618	9
SW11	1.64	600214	10
SW12	1.67	623622	11
SW13	1.69	618542	12

Source: by author (2016).

components: *Data Wrapper* and *Visualization*. The size of these elements was estimated in terms of CFPs, and the effort to develop each one is obtained by multiplying the number of CFPs and the PDR. The cost is calculated by multiplying the effort and the PHV.

The estimation of effort and cost was performed and analyzed for two assessment scenarios. The first one depicts results for datasets regarding Internet performance measurement of real world environments (multiple broadband providers). The raw data released by the FCC, for example, is a typical case of a large set of management data that can be better analyzed through visualization. Moreover, these datasets allow experimenting some challenges regarding visual analytics such as the volume of data. For example, the raw data of FCC for February 2014 (the period used in this case study) sums more than 15 GBytes of data. The second assessment scenario explores an SDN-based environment. Besides being considered a hot-topic into the network community, the literature shows that the employment of visualizations is an essential requirement to foster the adoption and development of the SDN management. Thus, an SDN experimental environment was designed and developed to carry out the evaluation. For both assessment scenarios, usage scenarios were defined, quantified, and discussed.

5 CONCLUSION

Based on a comprehensive survey (carried out through an SLR) on the literature regarding the employment of InfoVis for network and service management, this thesis addresses the field from another point of view. In essence, most of the works proposed in the literature have focused on the investigation of specific InfoVis techniques and its power for visual analytics in order to assist network administrators in different issues of network management (*e.g.*, security management). In fact, this investigation focus is very relevant and has provided a significant background in the field. However, up to now, no work has investigated how to promote the adoption of visualizations by network administrators (*i.e.*, to assist in everyday tasks), focusing on improving the productivity and decreasing cost. So, to achieve that, this thesis meets a set of concepts and standards concerning the software engineering field and proposes a reuse-based approach that aims at verifying the following hypothesis.

Hypothesis: **a reuse-based approach can be effective to decrease cost and improve the productivity of network administrators when adopting InfoVis techniques to assist in everyday tasks.**

The conducted investigations, the proposed approach, and the results presented in this thesis set a clear path towards supporting the proposed hypothesis. Such aspects are discussed and detailed over the answers to the three research questions (RQ) associated with the hypothesis, in Section 5.1. Afterward, Section 5.2 presents future work related to the results of this thesis.

5.1 Answers for the Fundamental Questions

In the light of the investigation presented in this thesis, it is possible to identify answers to the three research questions (RQ) associated with the hypothesis. Such answers are described below.

RQ I. *How a reuse-based approach can decreasing costs and increase the productivity of administrators when adopting InfoVis techniques?*

Answer: In analogy, this thesis indicates that the reuse of software components designed and developed to launching visualizations for network and service management can reduce cost and improve the productivity of network administrators in the same way that systematic software reuse in the software engineering domain. Essentially, build up from scratch is expensive and time-consuming. Moreover, in general, network administrators are unskilled in the InfoVis domain as well as they are non-experts in programming. In this sense, the proposed approach relies on the CBD and ROA architectural style which provide means to the black-box reuse. Thus, network administrators are able to reuse software components with no code modification, i.e., they can simply select the desired components and then launch the visualization. Furthermore, the proposal presented in this thesis envisages other two types of stakeholders: the InfoVis designer and the Web developer. It is understood that these stakeholders are fundamental to allow a systematic reuse. In short, they support and enhance the process of building, deploying, and maintenance of software components. Also, they can contribute together with network administrators to improve the knowledge base (through knowledge sharing and consensus building). By combining these aspects, the proposed reuse-based approach points a promising path to promote the adoption of visualizations by network administrators with the focus on reducing cost and improve the productivity.

RQ II. *What methods/mechanisms could be used in the design and development of a reuse-based approach?*

Answer: Software reuse was born in the second half of the 60s comprising an entire research line in the software engineering domain. Thus, throughout the years, the software reuse field has provided an appropriate scaffold in terms of concepts, standards, and best practices to the design and development of reuse-based solutions. In this sense, the proposal presented in this thesis leverages methods/mechanisms which coming from the legacy and maturity of the software reuse discipline.

As a starting point, it is was explored the guidelines provided by the Reuse Asset Management Processes defined in the IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes. This standard provides, among other things, seven expected outcomes as a result of a successful implementation of the Reuse Asset Management Process. From the seven expected outcomes, this thesis has investigated and employed three of them (since these three outcomes were considered key to the proposal): the asset classification scheme, the asset storage and retrieval mechanism, and the asset usage recording. Furthermore, in line with the IEEE Std 12207-2008, methods/mechanisms of CBD paradigm (*e.g.*, component lyfe cycle) and ROA architectural style (*e.g.*, software components as Web services resources) were used.

In order to design and implement the proposed solution, Web Semantic concepts and technologies (*e.g.*, SKOS, RESTful Web services, tagging, and ranking) were employed together with the software reuse principles above-mentioned. The key aspect here is that the proposed methods/mechanisms show a suitable path to support the systematic reuse in the proposed solution. Of course, other design decisions could be done from new or trending technologies. For example, JSON-LD[1] is a new format to link data on the Web, and it can be regarded as a lightweight ontology in the semantic spectrum (see Figure 3.4). In this sense, in the near future, SKOS could be replaced by JSON-LD to describe the asset classification scheme (although JSKOS is currently compatible with JSON-LD). However, this could be only an improvement (or an update) on the classification scheme once the fundamental concept remains the same.

RQ III. *What is the performance, in terms of cost and productivity, of a reuse-based approach?*

Answer: In order to answer this ask, it is understood that the first step is to place the answer to a predecessor question: how to measure cost and productivity in the software development process? In this point, the proposed evaluation relies again on the guidelines and procedures of the software engineering. Specifically, the COSMIC methodology was used to conduct the size estimation of the reusable software components developed for each evaluated network management scenario (*i.e.*, *Data Wrapper* and *Visualization*). With the size estimation of each software component, the effort and, consequently, the productivity and cost could be measured. Therefore, this was a fundamental stage to analyze and discuss the performance (in terms of cost and productivity) of the reuse-based approach proposed in this thesis.

Regarding the evaluation, in both network management scenarios, it was shown the estimated effort and cost to build reusable software components from scratch. Based on the obtained results it is possible to demonstrate the sub-

[1] http://json-ld.org/

stantial decrease in cost and effort with the reuse of at least one software component. For example, in S_1 (see Section 4.4), the total effort corresponds to 286 hours or U$ 10,807.94 (*i.e.*, to build W_1 and V_2), *i.e.*, more than one month and a half of work (considering that one person works 160 hours per month). If V_1 could be reused in S_1, this would represent 165 hours or U$ 6,235.35 of saving. In the second management scenario (Section 4.5), other aspects that can generate indirect savings of effort and cost are placed in the discussion. For instance, the possibility of extending the floating menu of V_4 by adding a button that makes an external call to the W_4 and V_2 components. This alternative makes possible that V_4 can be easily integrated (*i.e.*, with low effort and cost) with the view presented in S_6 by reusing the W_4 and V_2 components.
It is also noteworthy that the proposed reuse-based approach can provide indirect saving. Specifically, the consensus building through knowledge sharing can facilitate not only the reuse of software components but also the usage of visualizations by network administrators. An example is the background knowledge provided by feedbacks of how to get insights about information displayed in a given visualization. In this case, network administrators are able to, beyond the reuse, speed up their learning curve about a visualization of interest. Although hard to measure and quantify, this can be also an important outcome provided by the systematic reuse.

5.2 Future Work

This section describes further opportunities for research that were identified during the investigation developed in this thesis.

- **Version Control -** The integration of the proposed approach with widely-accepted version control systems (like Git). Essentially, this leads to achieving other two expected outcomes of the ISO/IEC 12207 / IEEE Std 12207-2008: (*i*) a set of criteria to the acceptance, certification, and retirement of assets; and (*ii*) a tool to notify users of assets regarding problems detected, modifications made, new versions created and deletion of assets.
- **Recommender system -** The knowledge base defined in the proposed approach enables a multitude of opportunities for the employment and evaluation of recom-

mender algorithms. A direct benefit of these algorithms can be the improvement of the seek process of reusable software components by stakeholders.

- **Security issues -** One can envisage that a key security issue is to keep the safety of the management datasets. In fact, network administrators could be reluctant to expose their management data to a third-party solution. In this sense, must be done an investigation of proper security mechanisms to dismiss such concerns and ensure, for example, the integrity and privacy of data.

REFERENCES

ALBRECHT, A. Measuring application development productivity. In: IBM APPLICATIONS DEVELOPMENT SYMPOSIUM, 1979, **Proceedings...**, [S.l.: s.n.], 1979. p. 83–92.

ALSALEH, M. et al. Visualizing PHPIDS log files for better understanding of web server attacks. In: WORKSHOP ON VISUALIZATION FOR CYBER SECURITY, 10, 2013. **Proceedings...** New York: ACM, 2013. p. 1–8. Available from Internet: http://doi.acm.org/10.1145/2517957.2517958.

AMES, M.; NAAMAN, M. Why we tag: Motivations for annotation in mobile and online media. In: THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2007. **Proceedings...** New York: ACM, 2007. p. 971–980. Available from Internet: http://doi.acm.org/10.1145/1240624.1240772.

BAJPAI, V.; SCHONWALDER, J. A survey on internet performance measurement platforms and related standardization efforts. **Communications Surveys Tutorials, IEEE**, vol. 17, no. 3, p. 1313–1341, thirdquarter 2015. ISSN 1553-877X.

BALDO JR., J.; MOORE, J.; RINE, D. Software reuse standards. **StandardView**, ACM, New York, NY, USA, vol. 5, no. 2, p. 50–57, jun. 1997. ISSN 1067-9936. Available from Internet: http://doi.acm.org/10.1145/260558.260559.

BARBOSA, P.; GRANVILLE, L. Interactive SNMP traffic analysis through information visualization. In: Network Operations and Management Symposium (NOMS), 2010 IEEE, [S.l.: s.n.], 2010. p. 73–79. ISSN 1542-1201.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. 2. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321154959.

BECKER, R. A.; EICK, S. G.; WILKS, A. R. Visualizing network data. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, vol. 1, no. 1, p. 16–28, mar. 1995. ISSN 1077-2626. Available from Internet: http://dx.doi.org/10.1109/2945.468391.

BELQASMI, F. et al. Soap-based vs. restful web services: A case study for multimedia conferencing. **Internet Computing, IEEE**, vol. 16, no. 4, p. 54–63, July 2012. ISSN 1089-7801.

BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. **Uniform Resource Identifier (URI): Generic Syntax**. 2005. RFC 3986.

BIERSACK, E. et al. Visual analytics for BGP monitoring and prefix hijacking identification. **Network, IEEE**, vol. 26, no. 6, p. 33–39, November 2012. ISSN 0890-8044.

BOBADILLA, J. et al. Recommender systems survey. **Knowledge-Based Systems**, vol. 46, p. 109 – 132, 2013. ISSN 0950-7051. Available from Internet: http://www.sciencedirect.com/science/article/pii/S0950705113001044.

BOKKELEN, J. V. **Responsibilities of Host and Network Managers. A Summary of the "Oral Tradition" of the Internet**. 1990. RFC 1173.

BOSTOCK, M.; OGIEVETSKY, V.; HEER, J. D3 - data-driven documents. **Visualization and Computer Graphics, IEEE Transactions on**, vol. 17, no. 12, p. 2301–2309, 2011. ISSN 1077-2626.

BRANCA, G.; ATZORI, L. A survey of SOA technologies in NGN network architectures. **IEEE Communications Surveys Tutorials**, vol. 14, no. 3, p. 644–661, Third 2012. ISSN 1553-877X.

BURCH, H.; CHESWICK, B. Mapping the internet. **Computer**, vol. 32, no. 4, p. 97–98, 102, Apr 1999. ISSN 0018-9162.

CARD, S. K.; MACKINLAY, J.; SHNEIDERMAN, B. **Readings in Information Visualization: Using Vision to Think**. 1st. ed. San Diego, CA, USA: Academic Press, 1999. ISBN 1558605339.

CEZZAR, J. **What is graphic design?** 2016. Available at: http://www.aiga.org/what-is-design/ . Acessed: August 2016.

CHAMBERS, J. et al. Graphical Methods for Data Analysis. **The Wadsworth Statistics**: probability series. Boston: Duxury, 1983.

CHEN, H.; WU, Z.; CUDRE-MAUROUX, P. Semantic web meets computational intelligence: State of the art and perspectives [review article]. **Computational Intelligence Magazine, IEEE**, vol. 7, no. 2, p. 67–74, May 2012. ISSN 1556-603X.

CHI, E. H. A taxonomy of visualization techniques using the data state reference model. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2000, **Proceedings...**, [S.l.: s.n.], 2000. p. 69–75.

CHOWDHURY, N.; BOUTABA, R. Network virtualization: state of the art and research challenges. **Communications Magazine, IEEE**, vol. 47, no. 7, p. 20–26, July 2009. ISSN 0163-6804.

CLEMM, A. **Network Management Fundamentals**. 1st. ed. Indianapolis, IN 46240 USA: Cisco Press, 2006. ISBN 1587201372.

CNOM. **Network and Services Management Taxonomy**. 2013. Available at: http://sites.ieee.org/tc-cnom/resources/taxonomy/ . Acessed: September 2015.

COLITTI, L. et al. Visualizing interdomain routing with BGPlay. In: JOURNAL ON GRAPH ALGORITHMS AND APPLICATIONS, [S.l.: s.n.], 2005. p. 2004–2005.

CORCHADO, E.; HERRERO, A. Neural visualization of network traffic data for intrusion detection. **Appl. Soft Comput.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, vol. 11, no. 2, p. 2042–2056, mar. 2011. ISSN 1568-4946. Available from Internet: http://dx.doi.org/10.1016/j.asoc.2010.07.002.

COSMIC. **The COSMIC Functional Size Measurement Method - Version 4.0.1**. 2015. Available at: http://cosmic-sizing.org/publications/measurement-manual-401/ . Acessed: September 2015.

DAVIES, J. Lightweight ontologies. In: POLI, R.; HEALY, M.; KAMEAS, A. (Ed.). Theory and Applications of Ontology: Computer Applications. Springer Netherlands, 2010. p. 197–229. ISBN 978-90-481-8846-8. Available from Internet: http://dx.doi.org/10.1007/978-90-481-8847-5_9.

DINH-DUC, A.-V.; DANG-HA, T.-H.; LAM, N.-A. Nviz - a general purpose visualization tool for wireless sensor networks. In: INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING/ELECTRONICS, COMPUTER, TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY, 9, 2012, **Proceedings...**, [S.l.: s.n.], 2012. p. 1–4.

EBERT, C.; SOUBRA, H. Functional size estimation technologies for software maintenance. **Software, IEEE**, vol. 31, no. 6, p. 24–29, Nov 2014. ISSN 0740-7459.

ERBACHER, R.; WALKER, K.; FRINCKE, D. Intrusion and misuse detection in large-scale systems. **Computer Graphics and Applications, IEEE**, vol. 22, no. 1, p. 38–47, Jan 2002. ISSN 0272-1716.

FANG, W.; MILLER, B. P.; KUPSCH, J. A. Automated tracing and visualization of software security structure and properties. In: INTERNATIONAL SYMPOSIUM ON VISUALIZATION FOR CYBER SECURITY, 10, 2012. **Proceedings...** New York: ACM, 2012. p. 9–16. Available from Internet: http://doi.acm.org/10.1145/2379690.2379692.

FCC. **Measuring Broadband America**. 2016. Available at: https://www.fcc.gov/measuring-broadband-america. Acessed: August 2016.

FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to sdn: An intellectual history of programmable networks. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, vol. 44, no. 2, p. 87–98, abr. 2014. ISSN 0146-4833.

FERENS, D. Software size estimation techniques. In: IEEE 1988 NATIONAL AEROSPACE AND ELECTRONICS CONFERENCE, 1988, **Proceedings...**, [S.l.: s.n.], 1988. p. 701–705.

FRAKES, W.; KANG, K. Software reuse research: status and future. **Software Engineering, IEEE Transactions on**, vol. 31, no. 7, p. 529–536, July 2005. ISSN 0098-5589.

FRAKES, W.; POLE, T. An empirical study of representation methods for reusable software components. **Software Engineering, IEEE Transactions on**, vol. 20, no. 8, p. 617–630, Aug 1994. ISSN 0098-5589.

FRAKES, W.; TERRY, C. Software reuse: Metrics and models. **ACM Comput. Surv.**, ACM, New York, NY, USA, vol. 28, no. 2, p. 415–435, jun. 1996. ISSN 0360-0300. Available from Internet: http://doi.acm.org/10.1145/234528.234531.

FRAKES, W. B.; GANDEL, P. B. Representing reusable software. **Inf. Softw. Technol.**, Butterworth-Heinemann, Newton, MA, USA, vol. 32, no. 10, p. 653–664, dez. 1990. ISSN 0950-5849. Available from Internet: http://dx.doi.org/10.1016/0950-5849(90)90098-C.

FU, C.; BELQASMI, F.; GLITHO, R. Restful web services for bridging presence service across technologies and domains: an early feasibility prototype. Communications Magazine, IEEE, vol. 48, no. 12, p. 92–100, December 2010. ISSN 0163-6804.

GENCEL, C.; DEMIRORS, O. Functional size measurement revisited. ACM Trans. Softw. Eng. Methodol., ACM, New York, NY, USA, vol. 17, no. 3, p. 15:1–15:36, jun. 2008. ISSN 1049-331X. Available from Internet: http://doi.acm.org/10.1145/1363102.1363106.

GOODALL, J. Visualization is better! A comparative evaluation. In: INTERNATIONAL WORKSHOP ON VISUALIZATION FOR CYBER SECURITY, 6, 2009, **Proceedings...**, [S.l.: s.n.], 2009. p. 57–68.

GRAND, B. L.; SOTO, M. Information management–topic maps visualization. In: XML EUROPE, 2000, **Proceedings...**, [S.l.: s.n.], 2000. p. 1–11.

GRIFFIN, D.; PESCH, D. A survey on web services in telecommunications. **IEEE Communications Magazine**, vol. 45, no. 7, p. 28–35, July 2007. ISSN 0163-6804.

GUIMARÃES, V. T. et al. A survey on information visualization for network and service management. **IEEE Communications Surveys Tutorials**, vol. 18, no. 1, p. 285–323, Firstquarter 2016. ISSN 1553-877X.

GUIMARÃES, V. T. et al. A collaborative solution for SNMP traces visualization. In: INTERNATIONAL CONFERENCE ON INFORMATION NETWORKING (ICOIN), 2014, **Proceedings...**, [S.l.: s.n.], 2014. p. 458–463.

GUINARD, D.; ION, I.; MAYER, S. In search of an internet of things service architecture: REST or WS-*?s a developers' perspective. In: INTERNATIONAL ICST CONFERENCE (MOBIQUITOUS), 8, 2011. **Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 326–337. Available from Internet: http://dx.doi.org/10.1007/978-3-642-30973-1_32.

HAINES, M. N.; ROTHENBERGER, M. A. How a service-oriented architecture may change the software development process. **Commun. ACM**, ACM, New York, NY, USA, vol. 53, no. 8, p. 135–140, ago. 2010. ISSN 0001-0782. Available from Internet: http://doi.acm.org/10.1145/1787234.1787269.

HAPPEL, H.-J. et al. KOntoR: An ontology-enabled approach to software reuse. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING, 18, 2006, **Proceedings...**, [S.l.: s.n.], 2006. p. 1–6.

HARRINGTON, D.; PRESUHN, R.; WIJNEN, B. **An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks**. 2002. RFC 3411, STD 62.

HEINRICH, J.; WEISKOPF, D. State of the Art of Parallel Coordinates. In: EUROGRAPHICS-STATE OF THE ART REPORTS, 2013, **Proceedings...**, [S.l.]: The Eurographics Association, 2013.

HOPKINS, J. Component primer. **Commun. ACM**, ACM, New York, NY, USA, vol. 43, no. 10, p. 27–30, out. 2000. ISSN 0001-0782. Available from Internet: http://doi.acm.org/10.1145/352183.352198.

HUANG, D.; HE, B.; MIAO, C. A survey of resource management in multi-tier web applications. **Communications Surveys Tutorials, IEEE**, vol. 16, no. 3, p. 1574–1590, Third 2014. ISSN 1553-877X.

IEEE Standard for Information Technology–System and Software Life Cycle Processes–Reuse Processes. IEEE Std 1517-2010 (Revision of IEEE Std 1517-1999), p. 1–51, Aug 2010.

INOUE, A.; TAKAHASHI, Y.; KOYAMA, A. MANET Viewer III: 3D visualization system for mobile ad hoc networks. In: INTERNATIONAL CONFERENCE ON NETWORK-BASED INFORMATION SYSTEMS (NBIS), 16, 2013, **Proceedings...**, [S.l.: s.n.], 2013. p. 178–185.

INOUE, D. et al. DAEDALUS-VIZ: Novel real-time 3D visualization for darknet monitoring-based alert system. In: INTERNATIONAL SYMPOSIUM ON VISUALIZATION FOR CYBER SECURITY, 9, 2012. **Proceedings...** New York: ACM, 2012. p. 72–79. Available from Internet: http://doi.acm.org/10.1145/2379690.2379700.

INSELBERG, A. The plane with parallel coordinates. **The Visual Computer**, Springer-Verlag, vol. 1, no. 2, p. 69–91, 1985. ISSN 0178-2789. Available from Internet: http://dx.doi.org/10.1007/BF01898350.

ISO/IEC. **ISO 7498-4: Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework**. 1989.

ISO/IEC. **ISO 10040: Information Processing Systems - Open Systems Interconnection - Systems Management Overview**. 1998.

ISO/IEC. **ISO/IEC 20968 - Software engineering - Mk II Function Point Analysis - Counting Practices Manual**. 2002.

ISO/IEC. **ISO/IEC 24570 - Software engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis**. 2005.

ISO/IEC. **ISO/IEC 14143-1 - Information technology - Software measurement - Functional size measurement - Part 1: Definition of concepts**. 2007.

ISO/IEC. **ISO/IEC 20926 - Software and systems engineering - Software measurement - IFPUG functional size measurement method 2009**. 2009.

ISO/IEC. **ISO/IEC 29881 - Information technology - Systems and software engineering - FiSMA 1.1 functional size measurement method**. 2010.

ISO/IEC. **ISO/IEC 19761 - Software engineering - COSMIC: a functional size measurement method**. 2011.

ISO/IEC/IEEE Standard for Systems and Software Engineering - Software Life Cycle Processes. IEEE STD 12207-2008, p. c1–138, Jan 2008.

ISOLANI, P. H. et al. Interactive monitoring, visualization, and configuration of openflow-based SDN. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT (IM), 2015, **Proceedings...**, [S.l.: s.n.], 2015. p. 207–215.

ITOH, T. et al. Hierarchical visualization of network intrusion detection data. **Computer Graphics and Applications, IEEE**, vol. 26, no. 2, p. 40–47, March 2006. ISSN 0272-1716.

JACOBS, I.; JAFFE, J.; HEGARET, P. How the open web platform is transforming industry. Internet Computing, IEEE, vol. 16, no. 6, p. 82–86, 2012. ISSN 1089-7801.

JOHNSON, B.; SHNEIDERMAN, B. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: IEEE CONFERENCE ON VISUALIZATION, 1991, **Proceedings...**, [S.l.: s.n.], 1991. p. 284–291.

JONES, G.; PRIETO-DIAZ, R. Building and managing software libraries. In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC), 12, 1988, **Proceedings...**, [S.l.: s.n.], 1988. p. 228–236.

KARAPISTOLI, E.; ECONOMIDES, A. Wireless sensor network security visualization. In: INTERNATIONAL CONGRESS ON ULTRA MODERN TELECOMMUNICATIONS AND CONTROL SYSTEMS AND WORKSHOPS (ICUMT), 4, 2012, **Proceedings...**, [S.l.: s.n.], 2012. p. 850–856.

KASTURE, A.; RAUT, A.; THOOL, S. Visualization of wireless sensor network by a java framework for security in defense surveillance. In: INTERNATIONAL CONFERENCE ON ELECTRONIC SYSTEMS, SIGNAL PROCESSING AND COMPUTING TECHNOLOGIES (ICESC), 2014, **Proceedings...**, [S.l.: s.n.], 2014. p. 256–261.

KAUR, K.; SINGH, H. Candidate process models for component based software development. **Journal of Software Engineering**, vol. 4, no. 1, p. 16–29, 2010.

KEELE, S. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.: s.n.], 2007.

KEIM, D. A. Information visualization and visual data mining. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, vol. 8, no. 1, p. 1–8, jan. 2002. ISSN 1077-2626. Available from Internet: http://dx.doi.org/10.1109/2945.981847.

KIM, H.; FEAMSTER, N. Improving network management with software defined networking. **Communications Magazine, IEEE**, vol. 51, no. 2, p. 114–119, February 2013. ISSN 0163-6804.

KOIKE, H.; OHNO, K.; KOIZUMI, K. Visualizing cyber attacks using IP matrix. In: IEEE WORKSHOP ON VISUALIZATION FOR COMPUTER SECURITY (VIZSEC 05), 2005, **Proceedings...**, [S.l.: s.n.], 2005. p. 91–98.

KOSARA, R. **eageareyes - Visualization and Visual Communication**. 2016. Available at: https://eagereyes.org/techniques/parallel-coordinates. Acessed: August 2016.

KOSARA, R.; MACKINLAY, J. Storytelling: The next step for visualization. **Computer**, vol. 46, no. 5, p. 44–50, May 2013. ISSN 0018-9162.

KOYAMA, A.; KAMAKURA, K.; BAROLLI, L. MANET-viewer: A visualization system for mobile ad-hoc networks. In: INTERNATIONAL CONFERENCE ON ADVANCES IN MOBILE COMPUTING AND MULTIMEDIA, 7, 2009. **Proceedings...** New York: ACM, 2009. p. 452–457. Available from Internet: http://doi.acm.org/10.1145/1821748.1821834.

KOYAMA, A. et al. An implementation of visualization system for visualizing network topology and packet flow in mobile ad hoc networks. In: INTERNATIONAL CONFERENCE ON NETWORK-BASED INFORMATION SYSTEMS (NBIS), 15, 2012, **Proceedings...**, [S.l.: s.n.], 2012. p. 148–155.

KRUEGER, C. W. Software reuse. **ACM Comput. Surv.**, ACM, New York, NY, USA, vol. 24, no. 2, p. 131–183, jun. 1992. ISSN 0360-0300. Available from Internet: http://doi.acm.org/10.1145/130844.130856.

LAI, Q. et al. Visualizing and characterizing DNS lookup behaviors via log-mining. Neurocomputing, vol. 169, p. 100–109, 2015. Available from Internet: http://dblp.uni-trier.de/db/journals/ijon/ijon169.html#LaiZMWC15.

LAKKARAJU, K.; YURCIK, W.; LEE, A. J. NVisionIP: Netflow visualizations of system state for security situational awareness. In: ACM WORKSHOP ON VISUALIZATION AND DATA MINING FOR COMPUTER SECURITY, 2004. **Proceedings...** New York: ACM, 2004. p. 65–72. Available from Internet: http://doi.acm.org/10.1145/1029208.1029219.

LAU, K.-K. et al. Component-based development. In: ____. CESAR - Cost-efficient Methods and Processes for Safety-relevant Embedded Systems. Vienna: Springer Vienna, 2013. p. 179–212. ISBN 978-3-7091-1387-5. Available from Internet: http://dx.doi.org/10.1007/978-3-7091-1387-5_5.

LEE, B. et al. Task taxonomy for graph visualization. In: WORKSHOP ON BEYOND TIME AND ERRORS: NOVEL EVALUATION METHODS FOR INFORMATION VISUALIZATION, 2006. **Proceedings...** New York: ACM, 2006. p. 1–5. Available from Internet: http://doi.acm.org/10.1145/1168149.1168168.

LIU, D. et al. Semi-automatic tagging of photo albums via exemplar selection and tag inference. **Multimedia, IEEE Transactions on**, vol. 13, no. 1, p. 82–91, Feb 2011. ISSN 1520-9210.

MAHMOOD, S.; LAI, R.; KIM, Y. Survey of component-based software development. **Software, IET**, vol. 1, no. 2, p. 57–66, April 2007. ISSN 1751-8806.

MCPHERSON, J. et al. PortVis: A tool for port-based detection of security events. In: ACM WORKSHOP ON VISUALIZATION AND DATA MINING FOR COMPUTER SECURITY, 2004. **Proceedings...** New York: ACM, 2004. p. 73–81. Available from Internet: http://doi.acm.org/10.1145/1029208.1029220.

MOERE, A. V.; PURCHASE, H. On the role of design in information visualization. **Information Visualization**, Sage Publications, vol. 10, no. 4, p. 356–371, 2011.

MOHAGHEGHI, P.; CONRADI, R. Quality, productivity and economic benefits of software reuse: A review of industrial studies. **Empirical Softw. Engg.**, Kluwer Academic Publishers, Hingham, MA, USA, vol. 12, no. 5, p. 471–516, out. 2007. ISSN 1382-3256. Available from Internet: http://dx.doi.org/10.1007/s10664-007-9040-x.

MORRIS, P. **The Cost of Speed**. 2016. Available at: http://www.totalmetrics.com/resources/software-measurement-articles/total-metrics-articles Acessed: August 2016.

MOUSAVI, H. et al. Mining semantic structures from syntactic structures in free text documents. In: IEEE INTERNATIONAL CONFERENCE ON SEMANTIC COMPUTING (ICSC), 2014, **Proceedings...**, [S.l.: s.n.], 2014. p. 84–91.

MUNZNER, T. **Visualization Analysis and Design**. [S.l.]: CRC Press, 2014. (AK Peters Visualization Series). ISBN 9781466508910.

NUNES, B. et al. A survey of software-defined networking: Past, present, and future of programmable networks. **Communications Surveys Tutorials, IEEE**, vol. 16, no. 3, p. 1617–1634, Third 2014. ISSN 1553-877X.

OFCOM. **Fixed-line broadband performance**. 2016. Available at: http://stakeholders.ofcom.org.uk/market-data-research/telecoms/broadband/. Acessed: August 2016.

ONF White Paper. **Software-Defined Networking: The New Norm for Networks**. [S.l.: s.n.], 2012.

PAPADOPOULOS, S.; MOUSTAKAS, K.; TZOVARAS, D. Hierarchical visualization of BGP routing changes using entropy measures. In: INTERNATIONAL SYMPOSIUM ON VISUAL COMPUTING, 2012, **Proceedings...**, Springer, 2012. (Lecture Notes in Computer Science, vol. 7432), p. 696–705. Available from Internet: http://dblp.uni-trier.de/db/conf/isvc/isvc2012-2.html#PapadopoulosMT12.

PAPADOPOULOS, S.; THEODORIDIS, G.; TZOVARAS, D. BGPfuse: Using visual feature fusion for the detection and attribution of BGP anomalies. In: Proceedings of the Tenth Workshop on Visualization for Cyber Security, New York, NY, USA: ACM, 2013. (VizSec '13), p. 57–64. ISBN 978-1-4503-2173-0. Available from Internet: http://doi.acm.org/10.1145/2517957.2517965.

PAPAZOGLOU, M. P.; GEORGAKOPOULOS, D. Introduction: Service-oriented computing. **Commun. ACM**, ACM, New York, NY, USA, vol. 46, no. 10, p. 24–28, out. 2003. ISSN 0001-0782. Available from Internet: http://doi.acm.org/10.1145/944217.944233.

PAUTASSO, C.; WILDE, E. Why is the web loosely coupled?: A multi-faceted metric for service design. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB (WWW), 18 ,2019. **Proceedings...** New York: ACM, 2009. p. 911–920. Available from Internet: http://doi.acm.org/10.1145/1526709.1526832.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. "big" web services: Making the right architectural decision. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB (WWW), 17, 2008. **Proceedings...** New

York: ACM, 2008. p. 805–814. Available from Internet: http://doi.acm.org/10.1145/1367497.1367606.

PFLEEGER, S.; WU, W.; LEWIS, R. 2005.

POULIN, J.; YGLESIAS, K. Experiences with a faceted classification scheme in a large reusable software library (RSL). In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC), 17, 1993, **Proceedings...**, [S.l.: s.n.], 1993. p. 90–99.

PRAS, A. **Network Management Architectures**. Tese (Doutorado) — University of Twente, Enschede, The Netherlands, 1995.

PRAS, A. et al. Key research challenges in network management. **IEEE Communications Magazine**, IEEE Communication Society, New York, vol. 45, no. 10, p. 104–110, October 2007.

PRICE, B. A.; BAECKER, R.; SMALL, I. S. A principled taxonomy of software visualization. **J. Vis. Lang. Comput.**, vol. 4, no. 3, p. 211–266, 1993. Available from Internet: http://dblp.uni-trier.de/db/journals/vlc/vlc4.html#PriceBS93.

PRIETO-DÍAZ, R. Implementing faceted classification for software reuse. **Commun. ACM**, ACM, New York, NY, USA, vol. 34, no. 5, p. 88–97, maio 1991. ISSN 0001-0782. Available from Internet: http://doi.acm.org/10.1145/103167.103176.

PROVAN, K. G.; KENIS, P. Modes of network governance: Structure, management, and effectiveness. **Journal of public administration research and theory**, PMRA, vol. 18, no. 2, p. 229–252, 2008.

RAVICHANDRAN, T.; ROTHENBERGER, M. A. Software reuse strategies and component markets. **Commun. ACM**, ACM, New York, NY, USA, vol. 46, no. 8, p. 109–114, ago. 2003. ISSN 0001-0782. Available from Internet: http://doi.acm.org/10.1145/859670.859678.

RENDON, O. C.; ESTRADA-SOLANO, F.; GRANVILLE, L. Z. A mashup ecosystem for network management situations. In: IEEE GLOBAL COMMUNICATIONS CONFERENCE (GLOBECOM), 2013, **Proceedings...**, [S.l.: s.n.], 2013. p. 2249–2255.

REVIEWS, T. T. **2016 Best Computer Career Reviews and Comparisons**. 2016. Available at: http://computer-careers-review.toptenreviews.com/network-administrator-review.html . Acessed: August 2016.

RICHARDSON, L.; RUBY, S. **Restful Web Services**. First. [S.l.]: O'Reilly, 2007. ISBN 9780596529260.

SALVADOR, E.; GRANVILLE, L. Using visualization techniques for SNMP traffic analyses. In: Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, [S.l.: s.n.], 2008. p. 806–811. ISSN 1530-1346.

SATO K., S. N.; SHIMADA, H. Visualization and management platform with augmented reality for wireless sensor networks. **Wireless Sensor Network**, vol. 7, no. 1, p. 1–11, January 2015.

SATO, S.; KOYAMA, A.; BAROLLI, L. MANET-Viewer II: A visualization system for visualizing packet flow in mobile ad-hoc networks. In: IEEE WORKSHOPS OF INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS (WAINA), 2011, **Proceedings...**, [S.l.: s.n.], 2011. p. 549–554.

SEN, S.; RIEDL, J. Folksonomy formation. Computer, vol. 44, no. 5, p. 97–101, May 2011. ISSN 0018-9162.

SHADBOLT, N.; HALL, W.; BERNERS-LEE, T. The semantic web revisited. **Intelligent Systems, IEEE**, vol. 21, no. 3, p. 96–101, Jan 2006. ISSN 1541-1672.

SHIRAVI, H.; SHIRAVI, A.; GHORBANI, A. A. A survey of visualization systems for network security. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, vol. 18, no. 8, p. 1313–1329, ago. 2012. ISSN 1077-2626. Available from Internet: http://dx.doi.org/10.1109/TVCG.2011.144.

SHNEIDERMAN, B. The eyes have it: A task by data type taxonomy for information visualizations. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES, 1996. **Proceedings...** Washington, DC, USA: IEEE Computer Society, 1996. p. 336–. Available from Internet: http://dl.acm.org/citation.cfm?id=832277.834354.

SINHA, A.; JAIN, H. Ease of reuse: An empirical comparison of components and objects. **Software, IEEE**, vol. 30, no. 5, p. 70–75, Sept 2013. ISSN 0740-7459.

STANGE, J.-E. et al. Visual filter: Graphical exploration of network security log files. In: WORKSHOP ON VISUALIZATION FOR CYBER SECURITY, 11, 2014. **Proceedings...** New York: ACM, 2014. p. 41–48. Available from Internet: http://doi.acm.org/10.1145/2671491.2671503.

STATISTICS Bureau of Labor. **Occupational Ooutlook Handbook**. 2016. Available at: http://www.bls.gov/ooh/computer-and-information-technology/web-developers.htm. Acessed: August 2016.

SZYPERSKI, C. **Component Software: Beyond Object-Oriented Programming**. 2nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0201745720.

TEOH, S. T. et al. A visual technique for internet anomaly detection. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND IMAGING (IASTED), 2002, **Proceedings...**, [S.l.]: ACTA Press, 2002. p. 1–8.

TORY, M.; MöLLER, T. Rethinking visualization: A high-level taxonomy. In: PROCEEDINGS OF THE IEEE SYMPOSIUM ON INFORMATION VISUALIZATION (INFOVIS), 2004. **Proceedings...** Washington, DC, USA: IEEE Computer Society, 2004. p. 151–158. Available from Internet: http://dx.doi.org/10.1109/INFOVIS.2004.59.

TSUTSUI, Y. et al. Development of a real-time system for visualizing network topology and packet flow in manet field experiments. In: INTERNATIONAL CONFERENCE ON MOBILE AD-HOC AND SENSOR NETWORKS (MSN), 10, 2014, **Proceedings...**, [S.l.: s.n.], 2014. p. 151–157.

VERAS, R.; THORPE, J.; COLLINS, C. Visualizing semantics in passwords: The role of dates. In: INTERNATIONAL SYMPOSIUM ON VISUALIZATION FOR CYBER SECURITY, 9, 2012. **Proceedings...** New York: ACM, 2012. p. 88–95. Available from Internet: http://doi.acm.org/10.1145/2379690.2379702.

VITHARANA, P. Risks and challenges of component-based software development. **Commun. ACM**, ACM, New York, NY, USA, vol. 46, no. 8, p. 67–72, ago. 2003. ISSN 0001-0782. Available from Internet: http://doi.acm.org/10.1145/859670.859671.

VOTH, G.; KINDEL, C.; FUJIOKA, J. Distributed application development for three-tier architectures: Microsoft on windows dna. **Internet Computing, IEEE**, vol. 2, no. 2, p. 41–45, Mar 1998. ISSN 1089-7801.

W3C. **SKOS Simple Knowledge Organization System Reference**. 2009. Available at: http://www.w3.org/TR/skos-reference/. Acessed: August 2016.

W3C. **World Wide Web Consortium (W3C) standards - Semantic Web**. 2016. Available at: http://www.w3.org/standards/semanticweb/. Acessed: August 2016.

WICKBOLDT, J. et al. Software-defined networking: management requirements and challenges. **Communications Magazine, IEEE**, vol. 53, no. 1, p. 278–285, January 2015. ISSN 0163-6804.

WIKI, I. **Treemap**. 2012. Available at: http://www.infovis-wiki.net/index.php?title=Treemap . Acessed: August 2016.

WIKIPEDIA. **Radar Chart**. 2016. Available at: https://en.wikipedia.org/wiki/Radar_chart . Acessed: August 2016.

WIKIPEDIA. **Web Developer**. 2016. Available at: https://en.wikipedia.org/wiki/Web_developer. Acessed: August 2016.

YANG, J.; EDWARDS, W. K.; HASLEM, D. Eden: Supporting home network management through interactive visual tools. In: ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY (UIST), 23, 2010. **Proceedings...** New York: ACM, 2010. p. 109–118. Available from Internet: http://doi.acm.org/10.1145/1866029.1866049.

YIN, X. et al. VisFlowConnect: Netflow visualizations of link relationships for security situational awareness. In: ACM WORKSHOP ON VISUALIZATION AND DATA MINING FOR COMPUTER SECURITY, 2004. **Proceedings...** New York: ACM, 2004. p. 26–34. Available from Internet: http://doi.acm.org/10.1145/1029208.1029214.

APPENDIX A - TAXONOMIES

The taxonomies used in this thesis are presented below (Table A.2 and A.1).

Table A.1: Information Visualization Taxonomy

Topic	Subtopic
Dataset types	- Tables (MUNZNER, 2014) - Multidimensional tables (MUNZNER, 2014) - Link/Node (MUNZNER, 2014) - Trees (MUNZNER, 2014) - Fields (MUNZNER, 2014) - Geometry (MUNZNER, 2014) - Static file (MUNZNER, 2014) - Dynamic stream (MUNZNER, 2014)
InfoVis Techniques	- Standard 2D/3D displays (KEIM, 2002) - Geometrically transformed displays (KEIM, 2002) - Icon-based displays (KEIM, 2002) - Dense pixel displays (KEIM, 2002) - Stacked displays (KEIM, 2002)
Tasks Interactions	- Overview (SHNEIDERMAN, 1996) - Zoom/Interactive Zooming (SHNEIDERMAN, 1996),(KEIM, 2002) - Filter/Interactive Filtering (SHNEIDERMAN, 1996),(KEIM, 2002) - Details-on-demand (SHNEIDERMAN, 1996) - Relate (SHNEIDERMAN, 1996) - History (SHNEIDERMAN, 1996) - Extract (SHNEIDERMAN, 1996) - Linking and Brushing (KEIM, 2002) - Moving/Rotate

Table A.2: Network and Service Management Taxonomy

Topic	Subtopic
Network Management	- Ad-hoc networks - Wireless & mobile networks - IP networks - LANs - Optical Networks - Sensor Networks - Overlay Networks - Virtual Networks - Software Defined and Programmable Networks - Data Center Networks - Smart Grids
Service Management	- Multimedia services (e.g., voice, video) - Data services (e.g., email, web) - Hosting (virtual machines) - Grids - Cloud services - Resource provisioning and management - QoE-centric management - Service discovery, migration and orchestration
Business Management	- Legal & ethical issues - Process management
Functional Areas	- Fault management - Configuration management - Accounting management - Performance management - Security management - SLA management - Event management
Management Approaches	- Centralized management - Distributed management - Autonomic and self management - Policy-based management - Federated network management - Pro-active management - Energy-aware network management
Technologies	- Protocols - Middleware - Mobile agents - P2P - Grid - Data, information, and semantic modeling - Cloud computing - Internet of Things - Human Machine interaction - Operations and Business Support Systems (OSS/BSS)
Methods	- Control theories - Optimization theories - Economic theories - Machine learning and genetic algorithms - Logics - Probabilistic, stochastic processes, queuing theory - Simulation - Experimental approach - Design - Monitoring & Measurements - Data mining and (big) data analytics

APPENDIX B - FCC AND OFCOM DATASETS

This appendix presents the metrics provided by the FCC (see Table B.1) and Ofcom (see Table B.2) datasets (in the evaluation referred to as D_1 and D_2, respectively).

Table B.1: Metrics provided by the MBA program of FCC

Metric	**Description**
Download speed	Throughput in Megabits per second (Mbps) utilizing three concurrent TCP connections
Upload speed	Throughput in Mbps utilizing three concurrent TCP connections
Web browsing	Total time to fetch a page and all of its resources from a popular website
UDP latency	Average round trip time of a series of randomly transmitted UDP packets distributed over a long timeframe
UDP packet loss	Fraction of UDP packets lost from UDP latency test
Video streaming	Initial time to buffer, number of buffer under-runs and total time for buffer delays
Voice over IP	Upstream packet loss, downstream packet loss, upstream jitter, downstream jitter, round trip latency
DNS resolution	Time taken for the ISP recursive DNS resolver to return an A record25 for a popular website domain name
DNS failures	Percentage of DNS requests performed in the DNS resolution test that failed
ICMP latency	Round trip time of five regularly spaced ICMP packet
ICMP packet loss	Percentage of packets lost in the ICMP latency test
Latency under load	Average round trip time for a series of regularly spaced UDP packets sent during downstream/upstream sustained tests
Availability	Total time the connection was deemed unavailable for any purpose, which could include a network fault or unavailability of a measurement point
Consumption	A simple record of the total bytes downloaded and uploaded by the router

Table B.2: Metrics provided by Ofcom.

Metric	Description
Download speed-24h	The average download speed (in Mbps) during 24 hours of measurement
Download speed-Max	The average of the Max download speed (in Mbps)
Download speed-8-10pm	The average download speed (in Mbps) between 8pm and 10pm in week days
Upload speed-24h	The average upload speed (in Mbps) during 24 hours of measurement
Upload speed-Max	The average of the Max upload speed (in Mbps)
Upload speed-8-10pm	The average upload speed (in Mbps) between 8pm and 10pm in week days
DNS resolution-24h	The average time taken (in ms) to resolve a domain name during 24 hours of measurement
DNS resolution-8-10pm	The average time taken (in ms) to resolve a domain name between 8pm and 10pm in week days
DNS failure-24h	The percentage of DNS requests performed in the DNS resolution test that failed during 24 hours of measurement
DNS failure-8-10pm	The percentage of DNS requests performed in the DNS resolution test that failed between 8pm and 10pm in week days
Jitter up-24h	The average upstream jitter (in ms) during 24 hours of measurement
Jitter up-8-10pm	The average upstream jitter (in ms) between 8pm and 10pm in week days
Jitter down-24h	The average downstream jitter (in ms) during 24 hours of measurement
Jitter down-8-10pm	The average downstream jitter (in ms) between 8pm and 10pm in week days
Latency-24h	The average latency (in ms) during 24 hours of measurement
Latency-8-10pm	The average latency (in ms) between 8pm and 10pm in week days
Packet Loss-24h	The percentage of packet loss during 24 hours of measurement
Packet Loss-8-10pm	The percentage of packet loss between 8pm and 10pm in week days
Web page-24h	The average time taken to loading a Web page during 24 hours of measurement
Web page-8-10pm	The average time taken to loading a Web page between 8pm and 10pm in week days

APPENDIX C - DOCUMENTATION OF COSMIC SIZE MEASUREMENT

This appendix presents the results of COSMIC size measurement for reusable software components V_1, V_2, V_3, and V_4 (see Figures C.1, C.2, C.3, and C.4, respectively), and W_1, W_2, W_3, W_4, and W_5 (see Figures C.5, C.6, C.7, C.8, and C.9, respectively). The employed structure is a recommendation of the COSMIC measurement manual version 4.1.0 (COSMIC, 2015) to documenting the results of a measurement. This structure is also referenced as the Generic Software Model matrix.

Figure C.1: COSMIC size measurement matrix of V_1

		Data group names						
	Parallel coordinates	Settings	Data Series	Entries	Exits	Reads	Writes	Total
Functional Process	Parallel coordinates rendering	X	X		X	X		3
	Grid Rendering	X	X		X	X		3
	Linking and Brushing		X	X	X	X		3
	Reordering axes		X	X	X	X		3
	Views' integration		X	X	X	X		3
							Total	**15**

Source: by author (2016).

Figure C.2: COSMIC size measurement matrix of V_2

		Data group names						
	Treemap	Settings	Data Series	Entries	Exits	Reads	Writes	Total
Functional Proces	Rendering	X	X		X	X		3
	Node details on mouseover		X	X	X	X		3
							Total	**6**

Source: by author (2016).

Figure C.3: COSMIC size measurement matrix of V_3

		Data group names						
	Star Plot	Settings	Data Series	Entries	Exits	Reads	Writes	Total
Functional Process	Rendering	X	X		X	X		3
	Hint on data series		X	X	X	x		3
							Total	**6**

Source: by author (2016).

Figure C.4: COSMIC size measurement matrix of V_4

		Data group names						
	Network Topology View	Settings	Topology data	Entries	Exits	Reads	Writes	Total
Functional Process	Topology rendering	X	X		X	X		3
	Floating Menu	X	X	X	X	X	X	6
	Moving Elements		X	X	X	X	X	4
	Zooming		X	X	X			2
	Panning		X	X	X			2
	Changing elements	X	X	X	X	X	X	6
	Simple dashboard integration	X	X	X	X	X	X	6
							Total	**29**

Source: by author (2016).

Figure C.5: COSMIC size measurement matrix of W_1

		Data group names							
	W1	Settings	Retrieved Data	Output Data	Entries	Exits	Reads	Writes	Total
Functional Process	Database handler	X			X	X			2
	Query handler	X			X	X			2
	Data gathering		X			X	X		2
	Data processing		X				X	X	2
	Release data to visualization		X	X		X	X	X	3
								Total	**11**

Source: by author (2016).

Figure C.6: COSMIC size measurement matrix of W_2

		Data group names							
	W2	Settings	Retrieved Data	Output Data	Entries	Exits	Reads	Writes	Total
Functional Process	File handler	X			X	X			2
	Data processing		X				X	X	2
	Release data to visualization		X	X		X	X	X	3
								Total	**7**

Source: by author (2016).

Figure C.7: COSMIC size measurement matrix of W_3

		Data group names							
	W3	Settings	Retrieved Data	Output Data	Entries	Exits	Reads	Writes	Total
Functional Process	Controller handler	X			X	X			2
	Topology data gathering		X			X	X		2
	Switch stats gathering		X			X	X		2
	Data processing		X	X			X	X	2
	Release data to visualization		X	X		X	X		2
								Total	**10**

Source: by author (2016).

Figure C.8: COSMIC size measurement matrix of W_4

	W4	Settings	Retrieved Data	Output Data	Entries	Exits	Reads	Writes	Total
		Data group names							
Functional Process	Controller handler	X			X	X			2
	Data gathering		X			X	X		2
	Data processing		X	X			X	X	2
	Release data to visualization		X	X		X	X		2
								Total	**8**

Source: by author (2016).

Figure C.9: COSMIC size measurement matrix of W_5

	W5	Settings	Retrieved Data	Output Data	Entries	Exits	Reads	Writes	Total
		Data group names							
Functional Process	Controller handler	X			X	X			2
	Data gathering		X			X	X		2
	Data processing		X	X			X	X	2
	Release data to visualization		X	X		X	X		2
								Total	**8**

Source: by author (2016).

APPENDIX D - PUBLISHED PAPERS AND COLLABORATIONS

Published Papers

1. **[Qualis A2]** - Vinícius Tavares Guimarães, Oscar Mauricio Caicedo Rendon, Gléderson Lessa dos Santos, Guilherme da Cunha Rodrigues, Carla Maria Dal Sasso Freitas, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville. **Improving Productivity and Reducing Cost Through the Use of Visualizations for SDN Management**. 21st IEEE Symposium on Computers and Communications (ISCC 2016), 27-30 June 2016, Messina, Italy, p. 556-563.

2. **[Qualis A1]** - Vinícius Tavares Guimarães, Carla Maria Dal Sasso Freitas, Ramin Sadre, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville. **A Survey on Information Visualization for Network and Service Management**. IEEE Communications Surveys and Tutorials, v. 18, p. 285-323, 2016. ISSN: 1553-877X.

3. **[Qualis B1]** - Vinícius Tavares Guimarães, Gléderson Lessa dos Santos, Guilherme da Cunha Rodrigues, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville. **A collaborative solution for SNMP traces visualization**. In: 2014 International Conference on Information Networking (ICOIN), 2014, Phuket, p. 458-463.

Other collaborations

1. **[Qualis A1]** - Guilherme da Cunha Rodrigues, Rodrigo N. Calheiros, Vinícius Tavares Guimarães, Gléderson Lessa dos Santos, Márcio Barbosa Carvalho, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville and Rajkumar Buyya. **Monitoring of Cloud Computing Environments: Concepts, Solutions, Trends, and Future Directions**. In: 31st Annual ACM Symposium on Applied Computing, 2016, Pisa. Proceedings of the 31st Annual ACM Symposium on Applied Computing, New York, USA, 2016, v. 1, p. 378-383.

2. **[Qualis A1]** - Oscar Mauricio Caicedo Rendon, Felipe Estrada-Solano, Vinícius Tavares Guimarães, Liane Margarida Rochenback Tarouco, Lisandro Zambenedetti Granville. **Rich dynamic mashments: An approach for network management based on mashups and situation management**. Computer Networks (1999), v. 94, p. 285-306, 2016. ISSN: 1389-1286.

3. **[Qualis A2]** - Gléderson Lessa dos Santos, Vinícius Tavares Guimarães, Guilherme da Cunha Rodrigues, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco. **A DTLS-based security architecture for the Internet of Things**. In: 2015 IEEE Symposium on Computers and Communication (ISCC), 2015, Larnaca, p. 809-815.

4. **[Qualis B1]** Guilherme da Cunha Rodrigues, Gléderson Lessa Dos Santos, Vinícius Tavares Guimarães, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco. **An architecture to evaluate Scalability, Adaptability and Accuracy in cloud monitoring systems**. In: 2014 International Conference on Information Networking (ICOIN), 2014, Phuket, p. 46-51.

5. **[Qualis B1]** Guilherme da Cunha Rodrigues, Vinícius Tavares Guimarães, Gléderson Lessa Dos Santos, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco. **Network and Services Monitoring: A Survey in Cloud Computing Environments**. In: The Eleventh International Conference on Networks, 2012, Saint Gilles, p. 7-13.