2017 IEEE 31st International Conference on Advanced Information Networking and Applications

A Reuse-based Approach to Promote the Adoption of Visualizations for Network Management Tasks

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*
*Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre/RS, Brazil
{vtguimaraes,glsantos,gcrodrigues,carla,tarouco,granville}@inf.ufrgs.br
†Universidad del Cauca, Popayán, Cauca, Colombia
omcaicedo@unicauca.edu.co

***Abstract*— Information Visualization (InfoVis) is a powerful tool to assist network administrators in daily tasks, and several authors report investigations about visualization techniques for network management. However, we have observed a fundamental issue that has not been addressed yet: how to promote the adoption of visualizations by network administrators, focusing on improving productivity and reducing costs? We claim that, in general, administrators do not have expertise in the InfoVis domain. Thus, the adoption of visualizations tends to be expensive and decrease the productivity of administrators, in special, because they lose focus on their core tasks. To overcome the raised issue, in this paper, we introduce a reuse-based approach (named as VisNSM) that aims to promote and encourage the adoption of visualizations for network management. To analyze and evaluate the feasibility of VisNSM, we have conducted a case study, and the obtained results show that our approach can significantly reduce costs and improve productivity.**

I. Introduction

Network management is accomplished through the use of multiple tools that help network administrators to perform the management process. Among such tools is visualization, which means the use of Information Visualization (InfoVis) [1] techniques to amplify human cognition in order to support users tasks more efficiently. Actually, although network analysis can be almost fully automated, that is only possible after having the main network scenarios properly identified by human reasoning. In this sense, InfoVis becomes essential to assist network administrators, for example, in the identification of usual/unusual patterns and the analysis of performance measurements.

InfoVis for network management is an old research topic [2], and several investigations were developed throughout the years [3]. In short, investigations have explored visualizations to help in everyday tasks of different management topics like Internet Service Provider (ISP) [4], Wireless Sensor Networks (WSN) [5], Mobile Ad-hoc Networks (MANET) [6], the Simple Network Management Protocol (SNMP) [7], Internet routing [8], and security management [9]. However, although these research have achieved relevant results and consolidated an important background in the field, we have observed a fundamental issue that has not been addressed yet: *how to promote the adoption of visualizations by network administrators, focusing on improving productivity and reducing costs?*

Essentially, we have realized that visualizations has not been widely adopted by network administrators mainly because: (*i*) they are often overwhelmed with their daily tasks [10] without enough time for non-primary assignments like building up visualizations, or consulting InfoVis designers and developers; and (*ii*) developing visualizations from scratch is usually a very time-consuming and expensive task, in special, for network administrators that, in general, are not skilled in programming [11]. Therefore, in most cases, administrators have to adapt to the visualizations available in traditional tools (*e.g.*, MRTG, CACTI, and Nagios) that are "as is". Based on that, as far as we are aware, network administrators are not able to enjoy the high-potential of visualizations because there is no solution that promotes the adoption of visualizations and takes into account productivity and cost.

In this paper, we introduce an approach (named as VisNSM) that aims at promoting the adoption of visualizations by encouraging the reuse. We advocate that a reuse-based approach is promising in four-fold: (*i*) it takes advantage of the currently available visualizations, (*ii*) it provides an easy path to share as well as 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 a nutshell, in VisNSM, we propose the use of concepts from the software reuse discipline and Semantic Web. Regarding the software reuse, we have explored the Reuse Asset Management Process of the IEEE Std 12207-2008 [12], which offers the guidelines for asset reuse. The Semantic Web [13] provides capabilities to share knowledge and consensus building to support a systematic reuse. To the best of our knowledge, there are no proposals to promoting the adoption of visualizations by network administrators, focusing on improving productivity and cost reduction. Therefore, the contributions of this paper are: (*i*) a reuse-based approach that aims at diminishing cost and improving the productivity of network administrators when they adopt visualizations; and (*ii*) a case study that demonstrates how VisNSM can significantly reduce costs and improve productivity.

The remainder of this paper is organized as follows. In Section II, we provide a brief description of the main background

1550-445X/17 $31.00 © 2017 IEEE
DOI 10.1109/AINA.2017.24

concepts associated with our approach and related work. In Section III, we introduce our proposal. In Section IV, we present the case study. Finally, in Section V, we conclude the paper with final remarks and perspective for future work.

II. Background and Related Work

This section introduces InfoVis for network management, and the main concepts of software reuse and Semantic Web.

A. InfoVis & Network and Service Management

Essentially, InfoVis comprises visualization techniques for “abstract” data, *i.e.*, data in a non-spatial domain and structures like multi-dimensional databases, text, graphs, and trees [14]. A widely known InfoVis reference model [1] describes the flow of information in InfoVis applications. Initially, raw data (*e.g.*, network management data) are gathered in data tables together with other derived information. Next, data tables are processed and transformed into one or more visual representations. Finally, the end-user (*e.g.*, the network administrator) manipulates and interacts with the visual representation in one or more views.

The research line on InfoVis for network management is not recent [2]. Throughout the years, several works have proposed InfoVis techniques for network management [3]. In the 90s, for example, visualizations were used to assist in the management of the AT&T’s network [4], and to display the Internet graph using routing and reachability data [15]. More recently, visualizations have been investigated for network traffic analysis [16], to assist in understanding the use of Simple Network Management Protocol (SNMP) [7], to help in the management of Wireless Sensor Networks (WSN) [5] and Mobile Ad-hoc Networks (MANET) [6], and to aid in comprehending the Internet routing data based on Border Gateway Protocol (BGP) information [8]. Moreover, many InfoVis investigations focused on security management [9], for instance, to help in security situational awareness [17], to detect port-based security events [18], and to assist in intrusion detection [19].

Despite the vast amount of investigations in the literature [3], we have realized that the fundamental issue concerning the adoption of visualizations by network administrators have not been observed yet. Thus, in this paper, we take a step further by investigating how to employ the principles of software reuse and Semantic Web to promoting the adoption of visualizations by network administrators with focus on improve productivity and cost reduction.

B. Software Reuse & Semantic Web

Software reuse is a traditional research topic in the system and software engineering field [20] and consists in the use of existing software or software knowledge to construct new software that includes, for example, libraries, domain engineering methods and tools, background knowledge, design, and design patterns [21]. In this paper, we have used some principles of software reuse to design and build VisNSM. Specifically, we have investigated the Software Reuse Process that is part of the Software Specific Processes defined in the international standard IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes [12]. Our basic insight is that, as in the system and software engineering domain, a reuse-based approach is effective to promote the adoption of visualization by networks administrators because they can take advantages of reusable assets (*e.g.*, visualizations and background knowledge) without decreasing their productivity and increase cost.

Regarding the Semantic Web [13], we argue that it provides helpful capabilities (like creating data stores, building vocabularies, and write rules for handling data) for supporting and improving the VisNSM approach. In short, the reuse process relies on a set of relevant information that documents reusable assets (*e.g.*, input and output parameters). Such information is essential to assist stakeholders for reusing assets in an easy way. Thus, we understand that such relevant information must be organized, stored, and shared to build a base of reusable knowledge and to avoid, among other things, ambiguities caused by heterogeneous descriptions.

III. VisNSM

This section describes the VisNSM proposal from the context and motivation to the software architecture.

A. Context & Motivation

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). We understand that the ultimate goal of such tools is to provide means to decrease the complexity and, consequently, to optimize the everyday tasks of administrators. Thus, network administrators can increase their productivity, and enhance the company’s profits through the reduction of costs.

Regarding visualizations, we argue that it is a powerful supporting tool for network administrators but its wide adoption is not a reality yet, mainly because it can be very laborious and expensive. First of all, the expertise in InfoVis domain is not a usual skill for network administrators. So, we have observed that the choice to adopt visualizations by administrators requires, in most cases, a diving into the unknown that can be too risky in terms of increasing costs and decreasing of productivity. In this sense, to better explain our arguments and motivation, we check three alternatives concerning the adoption of visualizations as follows:

(a) The network administrator adopts visualizations provided by traditional management tools. In general, such tools are proprietary or open-source. For the proprietary ones (*e.g.*, Solar-Winds and Hyperglance), we realize two major drawbacks: (*i*) it increases the Capital Expenditure (CapEx); and (*ii*) it affects the Total Cost of Ownership (TCO). The open-source tools (*e.g.*, MRTG and Elasticsearch-Logstash-Kibana - a.k.a. ELK) do not generate direct financial expenses but they bring hidden costs that increase the Operational Expenditures

(OpEx) (*e.g.*, setup, deployment, training, and maintenance costs). 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 to explore groundbreaking visualizations that can better fit with their management needs.

(b) Here, we envisage two possibilities. In the first one, the network administrator seeks for generic end-user tools that generate visualizations and do not require any coding (*e.g.* Tableau Public, Datawrapper, and Excel). The key pitfall of this possibility is the need of administrators to work with tools and technologies that are outside of their skills, which can be boring and time-consuming. Moreover, generic end-user tools address wide-range topics like global health, policy, and sports that are far from network management issues. The second possibility is the use of developer tools (*e.g.*, D3js and Google Charts) where the administrator builds up the visualization from scratch. Thus, he/she is in charge of select an adequate visualization (including interaction features, color effects, and so on), select a development framework, seek for supporting material for programming, deploying and maintaining. In short, this possibility is in the opposite way of facilitating and promoting the adoption of visualizations.

(c) The VisNSM reuse-based approach. In VisNSM, visualizations are reusable assets stored in a shared repository jointly with background knowledge. Thus, network administrators can seek for visualizations in the repository and use the background knowledge to support the reuse. Fundamentally, we advocate that the VisNSM is an effective approach to promote the adoption of visualizations by network administrators mainly because it is focused on visualizations for the network management domain, and takes advantage of reuse. We believe that the focus on network management of VisNSM is essential to engage network administrators since they are able to explore visualizations into their area of expertise. In addition, the reuse addresses the two key aspects previously introduced: productivity and cost.

B. Key Concepts

The central concepts that basis the VisNSM proposal are software reuse and Semantic Web. As the first step, we present the concept of the term **asset** and introduce the reusable assets of VisNSM. 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*" [22]. In VisNSM, there are two types of assets: (*i*) software elements, and (*ii*) background knowledge. The first one refers to a set of reusable software elements that are used, for example, to retrieve data from a given management dataset and to display such data through a visualization. The second one is the background knowledge built up by stakeholders through the systematic reuse of the software elements assets. A detailed explanation of reusable assets is presented in Section III-D.

Regarding the adaptation of software reuse concepts in VisNSM, we have explored the Software Reuse Process standardized in the international standard IEEE Std 12207-2008 - Systems and software engineering - Software life cycle processes [12]. The Software Reuse Process is divided into three main axes: (*i*) Domain Engineering Process, (*ii*) Reuse Asset Management Process; and (*iii*) Reuse Program Management Process. We focused on the Reuse Asset Management Process that aims to manage the life of reusable assets from conception to retirement. In particular, we are interested in the adaptation of three expected outcomes from a successful implementation of the Reuse Asset Management Process as follows: (*i*) the asset classification scheme, (*ii*) the asset storage and retrieval mechanism, and (*iii*) the asset usage recording. The IEEE Std 12207-2008 describes the expected outcomes in a general manner, *i.e.*, there are no low-level specifications concerning mechanisms, techniques, or technologies to achieve such outcomes. Based on this flexibility, in VisNSM, we have adapted concepts of the Semantic Web to materialize the three expected outcomes as follows.

Asset classification scheme - It relies on the Simple Knowledge Organization System (SKOS) [23] both to classify the software elements assets as for supporting the organization of the background knowledge. SKOS allows, among other things, porting existing knowledge organization systems (like taxonomies and classification schemes) to the Semantic Web. In this sense, VisNSM takes advantage of, for example, well-known taxonomies and classifications from the network management domain [24] and the InfoVis domain [25] for supporting the asset classification scheme.

Asset storage and retrieval mechanism - It is applied to the reusable software elements. In essence, each software element is composed of a Web Service and its metadata that are uniquely identified by a Uniform Resource Identifier (URI). Thus, the reusable software elements are stored in a traditional Web server and the retrieval of them is performed using the respective URI. This approach is effective in twofold: (*i*) it enables platform-independent access for humans through a Web browser; and (*ii*) it is machine-readable, allowing interaction with other systems (*e.g.*, mashups) in a simple way.

Asset usage recording - It records the usage of reusable software elements in four ways: (*i*) recording a log of assets usage that provides quantitative metrics such as the most used ones, (*ii*) enabling stakeholders to provide annotation related to the usage of reusable software elements by means of keywords (tagging), (*iii*) allowing stakeholders to write free-text annotations by describing opinion/insight/advice about the reused assets; and (*iv*) through a ranking system where stakeholders can give a grade after using a reusable asset.

C. Stakeholders

We are proposing the VisNSM architecture based on three stakeholders as follows (see also Figure 1).

Network administrator - he/she is the primary stakeholder of VisNSM and can be divided into two groups: (*i*) seniors that have significant knowledge and are experienced in network and service management (*e.g.*, a team leader); and (*ii*) mid-level/beginners that do not have advanced skills in network and service management (*e.g.*, operation staff). Both groups take advantage from VisNSM for employing visualizations in daily tasks through reuse. The first group is also encouraged to improve the knowledge database. For example, a senior administrator provides knowledge that helps to extend and improve the VisNSM database, or he/she describes management tasks in which reusable assets are helpful. The second group is encouraged to rate and provide feedbacks about reusable assets that they have used, and to learn from the knowledge supplied by seniors.

InfoVis expert - he/she has excellent skills in InfoVis, especially to build up visualizations for Web-based interfaces. In the context of VisNSM, the InfoVis expert could explore the network and service management domain to design and develop ground-breaking reusable visualizations, which is a promising way to enhance the *Visualization Repository* with novel visualizations. In addition, is expected that the InfoVis expert shares his/her know-how to improve the *Background Knowledge* database by providing, for example, how to better explore interactive features of a given visualization.

Wrapper developer - he/she has advanced skills in programming, in special, on Application Programming Interfaces (API) (*e.g.*, to pull data from management datasets). Wrapper developers could contribute to increasing the *Wrapper Repository* by developing novel *Data Pull* and *Data Push* assets as well as by reusing existing ones for new wrappers. They also take advantage of the background knowledge to understand the problem domain (*e.g.*, he/she must be aware of the semantics of management data to pull and parse information properly).

D. Conceptual Architecture, Elements & Interactions

VisNSM relies on concepts of the Service-Oriented Architecture (SOA) and the three-tier Web application architecture as described below (see Figure 1).

Presentation Tier (PT) - This tier has three primary functions: (*i*) admitting/denying requests performed by users (*e.g.*, network administrators) via Web browser (*e.g.*, to search for reusable assets), (*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 (LT) - 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*.

Data Tier (DT) - In essence, this tier handles the storage and management of persistent data (*e.g.*, the *Visualization Repository*). It is noteworthy that the *Data Tier* is not composed only of database systems because, as presented in Subsection III-B, reusable software elements (*e.g.*, visualizations) are Web services. Thus, the *Visualization Repository* is divided into a relational database that stores the metadata templates and the Web server that stores the Web services.

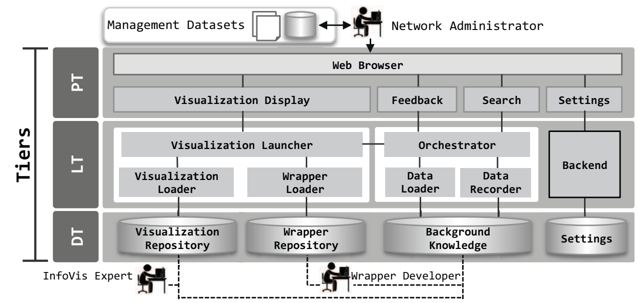

Figure 1. VisNSM architecture and stakeholders

We describe the architecture elements and its interactions by following the steps of a network administrator interested in using VisNSM. First, the network administrator accesses the search component in the PT to retrieve the desired visualization by typing keywords of interest. After the network administrator submits the search query, the *Search* component calls the *Orchestrator* component that performs the search by querying the *Background Knowledge* database. Let's suppose that the network administrator seeks for visualizations to display Internet performance measurements. The *Search* component will display a list of the available visualizations that meet this query. Then, the network administrator selects the one that fits his/her requirements. Such list shows a set of additional information (*e.g.*, ranking, annotations, and tags) that aims at supporting the network administrator's choice.

After the visualization is selected, the *Orchestrator* invokes the *Visualization Launcher* component that is in charge of running the *Visualization Loader* and *Wrapper Loader*. The *Visualization Loader* retrieves the selected visualization from the *Visualization Repository*, whereas the *Wrapper Loader* retrieves the corresponding wrapper that supports both the selected visualization and the management dataset provided by the network administrator. The *Visualization Repository* stores one of the reusable software elements introduced in Section III-B, named as reusable visualization. Thus, this repository stores the URI and the metadata template of each reusable visualization. The metadata template defines the structure of the visualization parameters (*e.g.*, settings, and data series format) to load and display the visualization.

The *Wrapper Repository* records the wrapper elements that are responsible for handling management datasets and prepares it according to the visualization metadata template. Wrappers are divided into three components (see Figure 2): the *Data Pull*, *Kernel*, and *Data Push*. The other reusable software elements introduced in Section III-B are the *Data Pull* and the *Data Push*. In essence, for each type of dataset there is one *Data Pull* component, and for each reusable visualization, there is one *Data Push* component. For example, in Figure 2 the W_1 and W_2 use the same *Data Pull* component, whereas the W_2 and W_4 use the same *Data Push* component. We argue that the design of modular wrappers is a crucial feature because it allows the extensibility of the architecture, makes reusable visualizations agnostic of the management dataset, speeds up the development of new wrappers and, consequently, the usage of reusable visualizations.

Resuming to the workflow of VisNSM, the *Visualization launcher* works as follows (see Figure 2). The network administrator provides a management dataset (*e.g.*, through access to a remote database or a dump file) that is loaded by the wrapper *Data Pull* component. The *Kernel* component processes the loaded data and delivers it to the *Data Push* component that formats the data according to the reusable visualization metadata template. Then, the *Visualization Launcher* releases the visualization for the *Visualization Display* in the PT.

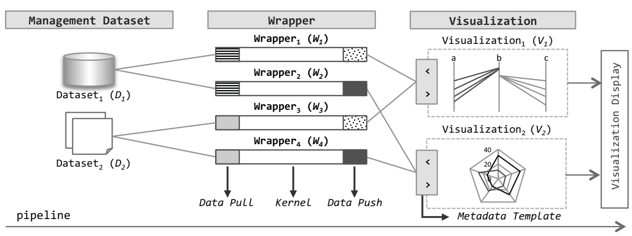

Figure 2. "*Visualization Launcher*" pipeline

As mentioned, the *Visualization Display* component shows the visualizations, and it is integrated with the *Feedback* component. Through the *Feedback* component, the network administrator can tag, rank, and provide free-text annotation concerning visualizations. Information gathered by the *Feedback* component is sent to the *Orchestrator* to update the *Background Knowledge* database. 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 DT.

IV. Case Study

This section presents a case study to demonstrate the feasibility of VisNSM concerning improve productivity and reduce cost.

A. Methodology

We defined four assessment scenarios, each one represented as follows: $VIS_n[D_n, W_n, V_n]$. VIS_n is a visualization that is composed of a triple where D_n represents the dataset, W_n the wrapper, and V_n the visualization itself. For each scenario, we measured the effort in terms of consuming of time (in work hours) to build up VIS_n. Such effort is calculated as $Effort(VIS_n) = Effort(W_n) + Effort(V_n)$, *i.e.*, by summing up the effort to develop a wrapper W_n, and the effort to develop V_n. Subsequently, we describe the datasets, wrappers, and visualizations. Afterward, we introduce each assessment scenario. Finally, we explain the method employed to measure effort, productivity, and cost. Table I summarizes the employed scenarios.

1) Datasets: We have used two open-access datasets of Internet performance measurements provided by (*i*) the Federal Communications Commission (FCC)[1] - the national regulator in the United States - in a campaign named as measuring broadband America; and (*ii*) the Office of Communications (Ofcom)[2], the national regulator in the United Kingdom. Both regulators have used the SamKnows platform to measure multiple broadband providers (*i.e.*, fixed-line access). We understand that such data sets fit with our case study because they are plenty of management information (*e.g.*, download and upload speed, RTT, latency, and packet loss) that can be better analyzed through visualizations. Hereinafter, we call the FFC's datasets as D_1 and the Ofcom's as D_2.

2) Visualizations: We employed two visualizations in this case study, named as V_1 and V_2. V_1 displays an interactive parallel coordinates view (at the top) integrated with a grid panel that details the information presented in the view (at the bottom). The interactive features of V_1 are linking and brushing. V_2 displays a radar chart (also known as spider or star plot), and its interactive features are (*i*) highlighting a data series when the mouse is over a particular point of it; and (*ii*) showing detailed information about a data point when the mouse is over it. Both techniques are helpful to display and analyze multivariate data in the form of a two-dimensional chart.

3) Wrappers: There are four wrappers W_1, W_2, W_3, and W_4. Each wrapper has a specific "*Kernel*" that parses the expected information for each scenario. On the other hand, the wrappers uses the "*Data Pull*" (*i.e.*, $DPull(D_1)$ and $DPull(D_2)$) and "*Data Push*" (*i.e.*, $DPush(V_1)$ and $DPush(V_2)$) software elements according to the scenarios defined for this case study (see Table I).

4) Scenarios: In the **Scenario 1** (S_1), we depict the worst case where the administrator does not take advantage of the $reuse$. In S_1, V_1 displays the average RTT, min RTT, max RTT, successes, and failures of the location with the highest average RTT. The **Scenario 2** (S_2) reuses the "*Data Pull*" component ($DPull(D_1)$), and V_2 displays a mean of the average RTT, min RTT, and max RT for the top ten measurement points (*i.e.*, the ten locations with the highest mean of average RTT). The **Scenario 3** (S_3) reuses the "*Data Push*" component ($DPush(V_1)$) and V_1. In S_3, V_1 displays information from different measurement points like the Internet Service Provider (ISP), headline speed, up/download speed for 24h, max up/download speed, up/download speed between 8-10pm in weekdays, and technology. In the **Scenario 4** (S_4), we depict the best case where the administrator takes advantage of the $reuse$ of "*Data Pull*" ($DPull(D_2)$), "*Data Push*" ($DPush(V_2)$), and V_2. In S_4, V_2 displays 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.

[1] https://www.fcc.gov/general/measuring-broadband-america

[2] http://stakeholders.ofcom.org.uk/market-data-research/other/telecoms-research/broadband-speeds/

Table I
SCENARIOS

Scenario	Dataset	Software Elements					Which is reusable?		
		Wrapper				Visualization	Wrapper Data Pull	Wrapper Data Push	Visualization
			Data Pull	Kernel	Data Push				
S_1	D_1	W_1	$W_1(DPull(D_1))$	$W_1Kernel$	$W_1(DPush(V_1))$	V_1			
S_2	D_1	W_2	$W_2(DPull(D_1))$	$W_2Kernel$	$W_2(DPush(V_2))$	V_2	✓		
S_3	D_2	W_3	$W_3(DPull(D_2))$	$W_3Kernel$	$W_3(DPush(V_1))$	V_1		✓	✓
S_4	D_2	W_4	$W_4(DPull(D_2))$	$W_4Kernel$	$W_4(DPush(V_2))$	V_2	✓	✓	✓

5) Effort estimation: Essentially, we advocate that building up the wrappers and visualizations for the presented scenarios is a software development activity. Thus, we have adopted the Common Software Measurement International Consortium (COSMIC) Functional Size Measurement (FSM) method that measures software sizing through Function Points (FP). We relied on the COSMIC measurement manual version 4.1.0 [26], which is the COSMIC implementation guide for the standards defined in ISO/IEC 19761:2011. We have adopted the COSMIC method for three main reasons: (*i*) it has generic applicability that comprehends software from different domains, (*ii*) it is used worldwide in various industries [27]; and (*iii*) it is entirely open, *i.e.*, all method documentation is available in the public domain.

The COSMIC unit of measurement is 1 CFP (**C**osmic **F**unction **P**oint), which is the size of one *data movement*. A *data movement* moves a single *data group* within elementary software components called "*functional processes*". There are four types of *data movement* that can be briefly described as follows: (*i*) an "*Entry*" moves data from a functional user into a *functional process*, (*ii*) an "*Exit*" moves data from a *functional process* to a functional user, (*iii*) a "*Read*" moves data from a persistent storage to a *functional processes*; and (*iv*) a "*Write*" moves data lying inside a *functional process* to persistent storage.

Based on the COSMIC method, the estimation of the total size of a visualization in CFPs is $Size(VIS_n) = Size(V_n) + Size(W_n)$, where $Size(W_n) = Size(W_n(DPull(D_n))) + Size(W_n(Kernel)) + Size(W_n(DPush(D_n)))$. To estimate the total effort in work hours, we use the following equation $Effort(VIS_n) = Size(VIS_n) * DPR$, where DPR means the Productivity Delivery Rate. DPR is defined in terms of hours per function point, *i.e.*, the average number of hours spent to produce a function point. In this case study, we used $PDR = 11$ hours per function point that is based on the average PDR for development team effort [28]. To estimate costs ($Cost$) we adopted the mid-range wage (in U$) of network administrators provided by the 2015 career reviews[3]. The 50th percentile within the salary range of network administrators is U$ 72,560 annually. Thus, we used the earning per hour (in U$) as follows $E_h = 37.79$, *i.e.*, 160 hours a month, twelve months per year.

[3]http://computer-careers-review.toptenreviews.com/

B. Prototype Implementation

We developed a VisNSM prototype to support this case study, using Web technologies and following the Model-View-Controller (MVC) design pattern in a client-server model. The *Model* layer is responsible for handling data repositories (*i.e.*, in DT) whereas the *Controller* layer (*i.e.*, in LT) processes data gathered from the *Model* layer, implementing the business logic, and forwarding the appropriate information to the *View* layer (*i.e.*, in PT). The *View* layer uses JavaScript, HTML and CSS languages. Reusable visualizations are implemented through the D3.js[4] library. Wrappers are developed in PHP and Shell Script languages. The server-side uses an Apache Web server running PHP on a Linux-based server. The *Visualization Repository* and *Wrapper Repository* are accessed through RESTful Web services. We used a relational database to store metadata and settings, and the *Background Knowledge* database is codified through JSKOS[5].

C. Results & Discussion

Based on the effort estimation methodology presented in Section IV-A5, we estimate separately the $Effort$ and $Cost$ (see Table II) for building up each software element defined for this case study (see Table I).

Table II
ESTIMATED VALUES

Software elements	CFP	$Effort$	$Cost = Effort * E_h$ (U$)
$DPull(D_1)$	6	66	2,594.14
$DPull(D_2)$	3	33	1,247.07
$DPush(V_1)$	3	33	1,247.07
$DPush(V_2)$	3	33	1,247.07
$W_1Kernel$	2	22	831.38
$W_2Kernel$	2	22	831.38
$W_3Kernel$	2	22	831.38
$W_4Kernel$	2	22	831.38
V_1	10	110	4,156.90
V_2	5	55	2,078.45

The score in CFPs for $W_1Kernel$, $W_2Kernel$, $W_3Kernel$, $W_4Kernel$ has produced the same result because these elements have two *functional processes* (*Retrieve data from* $DPull(D_n)$ and *Deliver data to* $DPush(V_n)$) and one *data group* (*Measurement data*) resulting in two *data movements*.

[4]https://d3js.org/
[5]https://gbv.github.io/jskos/jskos.html

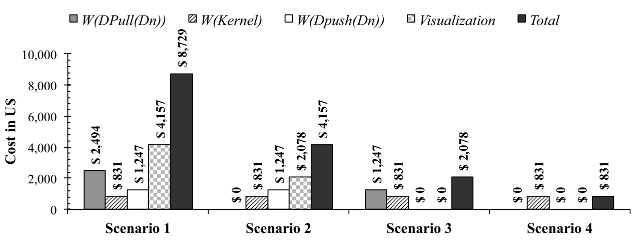

Figure 3. Estimated $Cost$ for each scenario

The same occurs for $DPush(V_1)$ and $DPush(V_2)$ that have three *functional processes* and one *data group* resulting in three *data movements*. It is important to highlight that the COSMIC Generic Software Model defines that *data manipulation* is not separately measured, *i.e.*, any *data manipulation* is assumed to be accounted for by the *data movement* with which it is associated. This definition helps us to clarify that, although these software elements have the same *functional processes* and *data group*, it does not mean that they have the same logical processing.

The estimation of $DPull(D_1)$ and $DPull(D_2)$ achieved 6 and 3 CFPs, respectively. Although the data provided by D_1 and D_2 are similar, we defined that D_1 is stored in a remote database and D_2 in a dump file. This difference in the storage type has added one *functional process* (*i.e.*, database access) and one *data group* (*i.e.*, database settings) resulting in additional 3 CFPs in the score of $DPull(D_1)$ compared to $DPull(D_2)$. Regarding V_1 and V_2 the higher score in CFPs for V_1 is mainly due to the features interactive linking and brushing and the integration with the data grid panel. Such features added 4 *data movements* in the V_1 score.

With the estimation in hands, we start from the effort spent for each scenario. In S_1 (without reuse) the effort corresponds to 231 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.*, $\simeq$ one month and a half). In S_2, the effort decreases to 110 hours with the reuse of the $W_2(DPull(D_1))$ component. In S_3, the effort corresponds to 55 hours (more than a week of work) by reusing $W_3(DPush(V_1))$ and V_1. Finally, in S_4 (by reusing $W_4(DPull(D_2))$, $W_4(DPush(V_2))$, and V_1), the effort reaches 22 hours (less than three days of work). Figure 3 details the costs accounting for each scenario. From the costs accounting one can infer that, in S_2, the reuse of $W_2(DPull(D_1))$ saves 66 hours (*i.e.*, U\$ 2,494), whereas in S_3 the reuse of $W_3(DPush(V_1))$ and V_1 sums 143 hours (*i.e.*, $\simeq$ U\$ 5,403) of saving. Thus, we can state that the reuse proposed by ViNSM approach can significantly reduce costs and improve productivity. For example, the effort spent in S_4 is $\simeq$ 95% lower than the effort in S_1, and the cost reduces in 50% when comparing S_2 and S_3.

We understand, however, that VisNSM provides capabilities that go beyond the showed numbers. In a nutshell, we advocate that the focus on network and service management associated with knowledge sharing and consensus building can decrease intangible costs (*e.g.* a misinterpretation of data because the network administrator is unfamiliar with the visualization technique). We take as an example the visualizations of S_2 (see Figure 4) and S_3 (see Figure 5). In the radar chart (S_2), the green and orange areas (Figure 4) 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 (S_3), the dark line in the view highlights an unexpected behaviour when the coordinate "Technology" is selected on ADSL1 and ADSL2+ (Figure 5). The measurements of download and upload speed exceed 65Mbps and 18Mbps, respectively, whereas the headline speed is 8Mbps. In this view, the unexpected behaviour can be promptly identified by analyzing the pattern of the other lines that are concentrated in the same range. In essence, we understand that the background knowledge provided by VisNSM bridges an important gap to promote the wide adoption of visualizations in the network and service management domain, in special when doing a comparison with the alternatives presented in Section III-A.

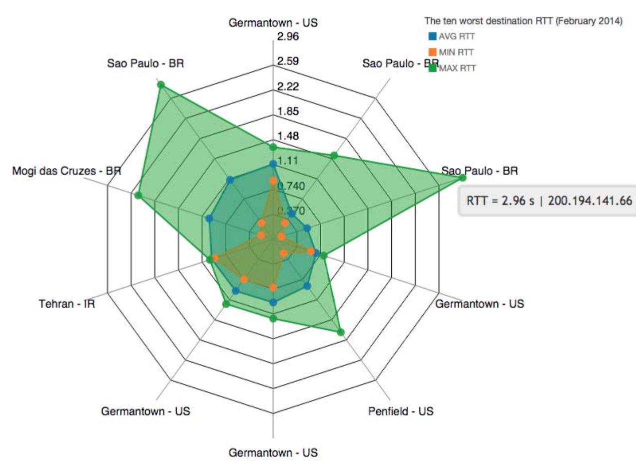

Figure 4. Radar chart in S_2

V. Conclusions and Future Work

In this paper, we introduced a reuse-based approach for promoting the adoption of visualizations for network management, named as VisNSM. We described VisNSM outlining its motivation, overview, conceptual architecture, and stakeholders. To evaluate the feasibility of our proposal we conducted a case study, where we have corroborated that VisNSM can significantly reduce costs and improve productivity (in terms of consuming of time) when network administrators adopt visualizations. To achieve that, we propose four assessment scenarios where we estimated the effort and costs to build up visualizations with and without reuse. Our results demonstrate that the effort spent in a scenario with reuse can be $\simeq$ 95% lower than the effort in a scenario without reuse. Moreover, we

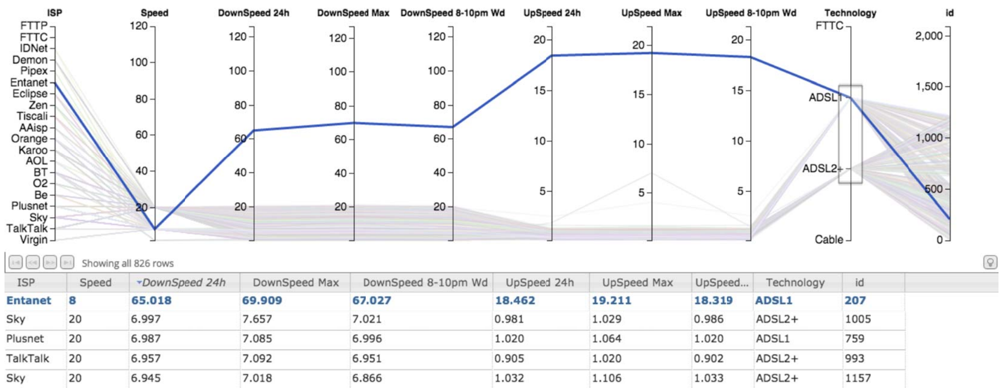

Figure 5. Parallel coordinates view in S_3

analyzed how the knowledge sharing and consensus building allowed by VisNSM can assist the employment of visualizations by network administrators. In short, we conclude that VisNSM is a promising approach to reduce costs and improve productivity.

As future work, we plan to develop more case studies exploring VisNSM for other network management scenarios (*e.g.*, Software-Defined Networks). Furthermore, we intend to investigate recommender mechanisms in order to use the information stored in the background knowledge database. We believe that a recommender system can make the reuse of visualizations easier and faster.

Acknowledgment

This work has been supported by PROPESP/IFSul, INF/UFRGS, CNPq, and CAPES-SETEC.

References

[1] S. K. Card, J. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*, 1st ed. San Diego, CA, USA: Academic Press, 1999.

[2] A. Pras, J. Schoenwaelder, M. Burgess, O. Festor, G. Martinez Perez, R. Stadler, and B. Stiller, "Key Research Challenges in Network Management," *IEEE Communications Magazine*, vol. 45, no. 10, pp. 104–110, October 2007.

[3] V. T. Guimarães, C. M. D. S. Freitas, R. Sadre, L. M. R. Tarouco, and L. Z. Granville, "A survey on information visualization for network and service management," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 285–323, Firstquarter 2016.

[4] R. A. Becker, S. G. Eick, and A. R. Wilks, "Visualizing network data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 1, pp. 16–28, Mar. 1995.

[5] S. N. Sato, K. and H. Shimada, "Visualization and management platform with augmented reality for wireless sensor networks," *Wireless Sensor Network*, vol. 7, no. 1, pp. 1–11, January 2015.

[6] Y. Tsutsui, Y. Kitaura, K. Kikuchi, T. Ohta, and Y. Kakuda, "Development of a Real-Time System for Visualizing Network Topology and Packet Flow in MANET Field Experiments," in *Mobile Ad-hoc and Sensor Networks*, Dec 2014, pp. 151–157.

[7] V. T. Guimarães, G. L. dos Santos, G. da C. Rodrigues, L. Z. Granville, and L. R. Tarouco, "A collaborative solution for SNMP traces visualization," in *ICOIN 2014*, Feb 2014, pp. 458–463.

[8] S. Papadopoulos, G. Theodoridis, and D. Tzovaras, "BGPfuse: Using Visual Feature Fusion for the Detection and Attribution of BGP Anomalies," in *VizSec'13*, New York, NY, USA, 2013, pp. 57–64.

[9] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 8, pp. 1313–1329, Aug. 2012.

[10] J. Hernantes, G. Gallardo, and N. Serrano, "It infrastructure-monitoring tools," *Software, IEEE*, vol. 32, no. 4, pp. 88–93, July 2015.

[11] O. Caicedo Rendon, F. Estrada-Solano, and L. Zambenedetti Granville, "A mashup ecosystem for network management situations," in *GLOBECOM 2013*, Dec 2013, pp. 2249–2255.

[12] "ISO/IEC/IEEE standard for systems and software engineering–software life cycle processes," *IEEE STD 12207-2008*, pp. c1–138, Jan 2008.

[13] H. Chen, Z. Wu, and P. Cudre-Mauroux, "Semantic web meets computational intelligence: State of the art and perspectives," *Computational Intelligence Magazine, IEEE*, vol. 7, no. 2, pp. 67–74, May 2012.

[14] M. Tory and T. Möller, "Rethinking visualization: A high-level taxonomy," in *INFOVIS '04*, Washington, DC, USA, 2004, pp. 151–158.

[15] B. Cheswick, H. Burch, and S. Branigan, "Mapping and visualizing the internet," in *USENIX 2000*, 2000, pp. 1–12.

[16] R. Hofstede and T. Fioreze, "SURFmap: A network monitoring tool based on the Google Maps API," in *IFIP/IEEE IM '09*, June 2009, pp. 676–690.

[17] K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: Netflow visualizations of system state for security situational awareness," in *VizSEC/DMSEC '04*, New York, NY, USA, 2004, pp. 65–72.

[18] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, "PortVis: A tool for port-based detection of security events," in *VizSEC/DMSEC '04*, New York, NY, USA, 2004, pp. 73–81.

[19] E. Corchado and A. Herrero, "Neural visualization of network traffic data for intrusion detection," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2042–2056, Mar. 2011.

[20] M. A. Rothenberger, K. J. Dooley, U. R. Kulkarni, and N. Nada, "Strategies for software reuse: A principal component analysis of reuse practices," *IEEE Trans. Softw. Eng.*, vol. 29, no. 9, pp. 825–837, Sep. 2003.

[21] W. Frakes and K. Kang, "Software reuse research: status and future," *Software Engineering, IEEE Transactions on*, vol. 31, no. 7, pp. 529–536, July 2005.

[22] "IEEE standard for information technology–system and software life cycle processes–reuse processes," *IEEE Std 1517-2010 (Revision of IEEE Std 1517-1999)*, pp. 1–51, Aug 2010.

[23] W3C, "SKOS Simple Knowledge Organization System Reference," 2009, available at: http://www.w3.org/TR/skos-reference/. Acessed: March 2016.

[24] CNOM, "Network and Services Management Taxonomy," 2013, available at: http://sites.ieee.org/tc-cnom/resources/ . Acessed: March 2016.

[25] T. Munzner, *Visualization Analysis and Design*, ser. AK Peters Visualization Series. CRC Press, 2014.

[26] COSMIC, "The COSMIC functional size measurement method - version 4.0.1," 2015.

[27] C. Ebert and H. Soubra, "Functional size estimation technologies for software maintenance," *Software, IEEE*, vol. 31, no. 6, pp. 24–29, Nov 2014.

[28] P. Morris, "The cost of speed," Total Metrics, Available at: http://www.totalmetrics.com/ Acessed: March 2016.