5-2017

Analysis of ICT services by observing “fit for use” attributes

David Giraldo
Universidad de Los Andes - Colombia, dn.giraldo10@uniandes.edu.co

Andrea Herrera
Universidad de Los Andes - Colombia, anhesue@gmail.com

Mario Sánchez
Universidad de Los Andes - Colombia, mar-san1@uniandes.edu.co

Jorge Villalobos
Universidad de Los Andes - Colombia, jvillalo@uniandes.edu.co

ANALYSIS OF ICT SERVICES BY OBSERVING "FIT FOR USE" ATTRIBUTES

David Giraldo
Universidad de los Andes
dn.giraldo10@uniandes.edu.co

Andrea Herrera
Universidad de los Andes
a-herrer@uniandes.edu.co

Mario Sánchez
Universidad de los Andes
mar-san1@uniandes.edu.co

Jorge Villalobos
Universidad de los Andes
jvillalo@uniandes.edu.co

Abstract

As organisations depend more and more on ICT services to meet their missions, ICT disruptions constitute an important risk to their resilience. Therefore, a systematic approach to prevent, predict and manage ICT services disruptions along their life cycle is needed. Simulation and visualisation techniques have been suggested as a means to explore "what-if" scenarios that allow organisations to prepare for different outcomes and consequently help them to improve their resilience. The research discussed in this paper explores how visual analysis of simulated scenarios can be used as a decision support mechanism to evaluate ICT readiness for organisational resilience. In particular, it presents how this can be supported by our extension of xArchiMate, a tool for simulating and visualising enterprise architecture models. This approach is evaluated by conducting experiments using the tool, analysing the results, and discussing how other extensions can be made to model additional scenarios.

Keywords:

ICT services, visual analysis, organisational resilience, simulation

1. Introduction

Organisations play a key role in delivering essential services that our society relies on and therefore disruptions to their operation can have significant and widespread impacts. Furthermore, the ever-increasing number of organisational risks and their potential negative impacts have resulted in the need for organisations to become more proactive in the management of the unexpected (McManus, Seville, Vargo, & Brunsdon, 2008). The ability of an organisation to survive and thrive in a world of uncertainty is known as organisational resilience (OR) (British Standards Institute, 2014; Burnard & Bhamra, 2011). A widely accepted definition of OR is "the ability of an organization to anticipate, prepare for, respond and adapt to incremental change and sudden disruptions in order to survive and prosper" (British Standards Institute, 2014, p. 2). In other words, OR is an organisation's ability to achieve their mission consistently in both routine and non-routine conditions.

As organisations depend more and more on ICT services to meet their missions, ICT disruptions constitute an important risk to their resilience. This forces organisations to consider the resilience of their critical ICT services and develop their ICT readiness for OR (Herrera & Janczewski, 2016). The need for a systematic approach to prevent, predict and manage ICT services disruptions along their life cycle has been identified (British Standards Institute, 2011). This includes methods to analyse and make decisions in early stages of the life cycle so that, customers can be confident that their ICT services are being designed to effectively and consistently support their business requirements, as well as methods to support their maintenance minimising disruptions. Due to the high complexity and heterogeneity of ICT services architectures, this is not a trivial problem and both practitioners and academics have recognized that solutions to these challenges must have several dimensions (Caralli, Allen, Curtis, White, & Young, 2010; Tertilt & Krcmar, 2011).

From an ICT service management perspective, customers want to achieve business outcomes by using ICT services that are appropriate for their purpose. One way to define and assess the appropriateness of an ICT service is through what is known as a "warranty of service" which is a way "to communicate to customers in terms of commitments to the availability, capacity, continuity and security" (BMC Software Inc, 2016, p. 6) of the service. Specific requirements for each of these attributes are driven by business requirements, and a range of approaches dealing with the management of each of them separately are available in the literature (Bosse, Schulz, & Turowski, 2014; Dutta & Roy, 2003; Kumar & Bhat, 2009; Miller & Engemann, 2014). However, analysing them individually ignores the relationships between them. For instance, a service with very high response times due to not having enough capacity will be categorized by the customer as being unavailable. To add another example, when understanding capacity requirements it is necessary to forecast the impact of events such as a spike in the number of users in order to define the right strategy. In other words, there is a need to balance supply against demand while meeting the agreed targets. Therefore, this is an important part of designing and maintaining resilient ICT services. To proactively cope with these challenges, prescriptive and predictive analytics as well as simulation and visualisation techniques have been suggested as a means to explore "what-if" scenarios and allow organisations to prepare for different outcomes (IBM Corporation, 2015), whether they are considering short-term or long-term decisions in order to improve their resilience.

Therefore, we specify our overarching research question as follows: *RQ - How visual analysis of simulated scenarios can be used as a decision support mechanism for ICT readiness for OR.* In particular, the paper presents how this is supported by xArchiMate (Manzur, Ulloa, Sánchez, & Villalobos, 2015), a tool for simulation and visualisation of enterprise architecture models. This paper also reports the experiments we conducted using the tool; describes some extension we made in the simulation meta model in order to address dynamic and static requirements for each of the "fit for use" attributes and their dependencies; analyses the results, and discusses how other extensions can be made to model other scenarios. This paper is structured in five sections, including this introduction. Section 2 begins by presenting a brief overview of techniques for modelling and analysing ICT services attributes found in the literature. The third section is focused on describing the visual animation tool, the extensions made, and how the extended tool can be used as a mechanism to support decision makers in analysing scenarios and understanding how different ICT service designs might perform under different circumstances. Section 4 presents and discusses the findings from the series of experiments conducted. In Section 5 we draw conclusions, present contributions, and consider limitations and directions for future research.

2. Simulation and the ICT services "fit for use" attributes

When analysing the appropriateness of an ICT service through the "warranty of service" attributes – availability, capacity, continuity and security – we can identify three main dimensions: the domain dimension, temporal dimension and analysis technique dimension. The domain dimension refers to the service components that are been analysed (technological infrastructure, organisational resources, business processes and external events). The temporal dimension refers to when the analysis can be done either before or after the service has been implemented, that is, in the design stage or in the operations and transition stages. Finally, the analysis technique dimension refers to how ICT services attributes are studied and are commonly classified into qualitative, quantitative, and analytical and simulation approaches (Bosse et al., 2014; Miller & Engemann, 2014). Qualitative techniques provide estimations on the basis of expert opinions and are usually performed through questionnaires and interviews; results from this type of techniques are highly dependent on the expert performing the analysis and therefore difficult to compare (Bosse et al., 2014; Malek, Milic, & Milanovic, 2008; Miller & Engemann, 2014). Quantitative techniques are based on measurements. The key process in this category is to collect and store data from the real ICT service in order to monitor, assess and predict ICT services attributes. For instance, a robotic response time-monitoring agent monitors the performance and availability of a service by recording and playing back transactions in order to determine whether a transaction is performing as expected (Bosse et al., 2014). Whereas these techniques have proven themselves in several areas (Jiang & Hassan, 2015), the effects of changes in an ICT service cannot be analysed until new monitoring data is gathered and analysed, reducing the applicability of these techniques (Bosse et al., 2014; Malek et al., 2008). Analytical and simulation techniques are used when it is not possible to access the real system. Analytical methods describe the system through a set of equations and calculate quality attributes through numerical solutions, Simulation methods emulate the system behaviour through a model. These models facilitate the experimentation of different scenarios by observing possible results when the cost, time or risk of experimentation with the real system could be high or unfeasible (Orta, Ruiz, Hurtado, & Gawn, 2014).

As we are interested in techniques for modelling and analysing the “fit for use” attributes of complex ICT services and their dependencies at any stage of the ICT services life cycle, we conducted a literature review focusing on simulation techniques and searched four online databases: IEEE XPlore, Science Direct, ProQuest (ABI/INFORM), and Business Source Premier (BPS). This resulted in the identification of five main approaches that support informed decision-making processes in order to meet expected business outcomes and thus help to improve an organisation’s resilience. Next, we briefly present and compare these approaches.

Firstly, the Palladium Component Model (PCM) is a domain-specific modelling language that has been used to describe, model, analyse and simulate software architectures especially before their deployment (Becker, Koziolek, & Reussner, 2007; Bosse et al., 2014; Heinrich, Merkle, Henss, & Paech, 2015). Heinrich et al. (2015) extended the PCM model (PCM+) by including meta-models for business processes. This enabled an integrated simulation, including mutual impact between business processes and ICT services. The simulation is performed with the discrete event simulator included in the PCM framework. Bosse et al. (2014) define a simulation model based on Petri nets, which allows the estimation of the quality attributes of ICT services. Secondly, the Risk-Oriented Process Evaluation (ROPE) methodology consists of a meta-model that has three layers: the business process layer that models activities; the CARE (Condition, Action, Resources and Environment) layer that describes elements (actions, resources and environments), constraints and relationships between these elements or conditions; and the TIP (Threat Impact Process) layer that describes the external threads management (Tjoa, Jakoubi, & Quirchmayr, 2008). This methodology sets a starting point for modelling external threats for business process and ICT services.

The third approach is an artefacts-based simulation model that is intended to be used for maintaining and operating software architectures that implement ICT services in order to estimate performance parameters and ensure service level agreements (Hill, 2011). This model blends existing ICT services industry practices with tools to provide the needed artefacts to maintain and operate the service and uses them to design simulation elements within an infrastructure and process workflow layer. The fourth approach is an error-based simulation model that leverages information about existing service components and their interactions and provides concrete service behaviour in presence of a variety of errors (Wang, Sahai, & Pruyne, 2007). Such models are based on information models such as the Unified Modelling Language (UML) or the Common Information Model (CIM), which are enriched with an information model for internal errors of service components. They employ event-driven simulation to mimic the behaviour of an environment under errors. Finally, xArchiMate is a simulation meta-model for executing enterprise models (Manzur et al., 2015). It extends the ArchiMate® specification (The Open Group, 2016) – a visual modelling notation for enterprise architecture models – by collecting data about dynamic features that together with static information can be used to support decision-making processes. This simulation meta-model is a Java-based simulation engine that relies on model-driven engineering. Although xArchiMate is not as detailed as PCM+ regarding ICT services, it provides a simplified, easy-to-understand and easy-to-extend set of concepts that fully integrate business processes, applications and infrastructure.

At the beginning of this section we identified three main dimensions in order to analyse the appropriateness of an ICT service: the domain dimension, the temporal dimension and the analysis technique dimension. Given that we have chosen simulation as the analysis technique, any specific approach is applicable either before or after the ICT service implementation (temporal dimension), therefore, our comparison is mainly focused on the approaches' potential to model different domains, particularly business and infrastructure components as well as their potential to model the four listed attributes and their dependencies. Table 1 shows this comparison. PCM+ is a strong tool for analysing the software and hardware components of a service in detail, however the mutual impact does not allow modelling of external influences such as attacks, and thus it does not permit proper analysis of the security attribute. ROPE is a methodology which provides an interesting language for modelling external threats, however it does not describe enough the hardware and software components, which makes it difficult to analyse the service capacity attribute; ROPE also does not provide an implemented simulation model. The artefacts-based simulation model has the same weakness as PCM+ regarding the business environment and it does not model software and applications. The error-based model, meanwhile, lacks the potential to model and simulate the interaction between business process and environment. Finally, xArchiMate is able to model behavioural information in addition to the structural information of different domains, their relationships and their dependencies. Although it does not explicitly model any of the "warranty of service" attributes, it does provide a simplified and easy-to-extend set of concepts in order to mimic them. Based on these advantages, we chose xArchiMate as our simulation model. In the next section we explain the necessary extensions and how we have made them in order to fully address our requirements.

	Attribute				**Domain**			
	Capacity	Availability	Continuity	Security	Technology		Business	
Approach					Hw	Sw	Process	Environment
PCM+	√	√	√		√	√	√	
ROPE		√	√	√			√	√
Artefacts-based model	√	√	√		√		√	
Error-based models	√	√	√		√	√		
xArchiMate	√	√	√	√	√	√	√	√

Table 1: Comparison of approaches

3. Extending xArchiMate to visually analyse ICT services

As stated previously, understanding specific requirements for each of the "fit for use" attributes and their dependencies, as well as proactively defining their targets and measuring their potential levels, is an important part of designing and maintaining resilient ICT services. In order to do so, we propose to extend xArchiMate's meta model to explore "what-if" scenarios that allow organisations to prepare for different outcomes and consequently help them to improve their OR. In this section we describe xArchiMate's main features, followed by a brief description of the visualisation engine and the extensions made in order to support modelling and simulating the defined attributes.

ArchiMate® is an enterprise architecture modelling language that offers an integrated view of the enterprise – business and ICT – supporting the description, analysis and visualisation of different architecture domains and their underlying relations and dependencies (The Open Group, 2016). It has rapidly become the standard language for describing enterprise architecture models and now various tools support it. These include Archi® a free and open source ArchiMate® modelling tool. The core of ArchiMate® clusters enterprise elements into three layers: business (actors, roles, business services, processes, activities events, and business objects), application (services, functions, components and data objects) and infrastructure (services, functions, system software, devices, networks and artefacts); and into three categories: active elements (actors, roles, components, system software, devices, and networks), behaviour elements (processes, activities, services, functions) and passive elements (business object, data object, and artefact). Various methods to analyse ArchiMate® models have been proposed, however they are typically static, focusing mainly on structural features (Manzur et al., 2015). This makes it difficult to analyse dynamic attributes such as capacity and performance, or to understand how alternative designs would perform under different conditions.

To address this situation, Manzur et al. (2015) proposed xArchiMate, whose main goal is to create enriched ArchiMate® models with dynamic properties, allowing the execution / simulation of these models. In order to do so, xArchiMate focuses on the three key layers of ArchiMate® and adds a new dimension by classifying elements into definitions and instances. Definitions refer to elements or abstract concepts that can be included in the model, as they actually exist, while instances are elements that appear only at run time and are created during the simulation. An example of the former is a process definition, while an example of the latter is a specific process instance (or case). Passive and behavioural elements are instantiable but active elements are not instantiable. The three types of elements in xArchiMate – active, behaviour and passive – are implemented by open objects. Each open object depicts an element and consists of a Java entity that defines the attributes and behavioural actions of that element and a state machine that defines its possible states. An open object accesses other open objects by interacting with them through events (Manzur et al., 2015).

In order to visualise simulations run in xArchiMate, an animation-based engine was added on top of xArchiMate's simulation engine. It consists of two main components: a tracer that is responsible of storing the evolution of a simulation model into simulation traces during the simulation, and an Eclipse-based plugin that is integrated into Archi® to create dynamic ArchiMate® enterprise models. The visualisation engine (Figure 1) is responsible for visualising the simulation through customizable animated icons. These icons are associated to model elements and drawn over the components of an Archi® diagram.

The animated icons consist of shapes (polygons, text labels and images) and operations that modify their graphical attributes, and several operations are grouped into an animation. When the visualisation is running, the simulation's traces are read to determine which elements have changed, modifying the icons accordingly. Such an animated visualisation helps to develop a better, easier and clearer understanding about the structure and behaviour of a service, leading to important insights for solving problems and/or to making improvements (Gogi, Tako, & Robinson, 2016). In other words, visualisation is the way in which the simulation data is translated into human-understandable information and therefore it is expected to help in understanding the dynamic features of ICT services. This recognises that since a simulation model is a simplification of the real world, insights must be interpreted. Additional extensions were needed in the xArchiMate meta-model in order to fulfil our requirements. The two main additional extensions are explained below.

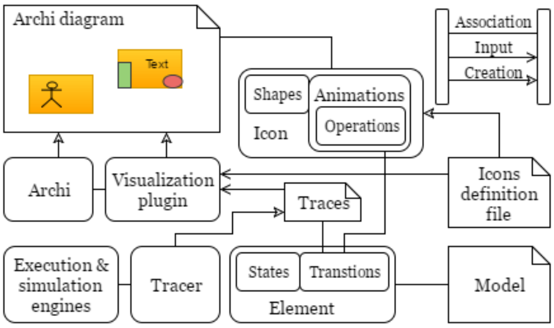


Fig 1: Structure of the visualisation engine using Archi®

Firstly, the open objects of the application function and system software types were modified in order to model request-processing more closely to reality. There are two cases: with the infrastructure layer and without the infrastructure layer. When the infrastructure layer is not modelled, request processing is performed by application functions; an example would be when an organisation is not interested in how required functions are implemented. When there is an infrastructure layer, request processing depends on nodes: a device provides resources (RAM, threads and disk) and its associated system software process uses resources. In the former case, the open object of an application function is modified to include two queues: one for concurrent processing requests and the other for pending requests. In the latter case, a RAM buffer is assigned to the system software in which the pending requests are held; the current request-processing depends on the device resources. Therefore, the processing capacity of application functions and nodes depends on the maximum number of concurrent processing requests and the capacity of the pending-requests queue; these parameters are specified within the experiment definition. In the case that the workload of the experiment makes the application function queues reach their maximum capacity, the component fails and goes into the unavailable state. We also included a third state (recovery), in which the pending requests must be processed before entering into the available state. The processing time of a request also depends on the quantity of current processing requests and pending requests; its degradation factor was also specified for the experiments.

The second extension is related to events and is useful for modelling the environment of an ICT service. The latest version of ArchiMate® (The Open Group, 2016) improves the original meta-model of business, application and infrastructure layers by including events that can trigger or be triggered by behavioural elements. Based on this idea, we created two new behavioural element types: event and event instance. These elements depict internal and external events of any nature. Internal events can be generated by active and behavioural elements as well as their instances, while external events depict external influences such as threats. Elements that react to an event can be active and definitions of behaviour elements (not instances). They experience a reaction that can be: instantiation, state change or down/up (the last one is used to depict situations in which the element fails and returns to available or recovery state after a given time). Finally, an event has associated with it a probability of success. For example, if an event depicts an attack to a server, the probability of success determines how likely it is that the server will crash; it is also possible to specify reactions when an event is unsuccessful.

4. Experiment results and analysis

In this section we describe the application of our tool for visual analysis of ICT services attributes through the running of controlled experiments. As the main goal is to explore how visual analysis of simulated scenarios can be used as an organizational resilience decision support mechanism, we design two experiments aiming to observe and visually analyse two scenarios. In the first experiment we focus on modelling and analysing the capacity, availability and continuity attributes of an ICT service as well as the dependencies among them, while in the second experiment we focus on analysing availability and continuity by adding an external event. The latter, as a starting point for analysing the security attribute, that can be easily modelled through events emulating several kinds of attacks and by running similar experiments. Next we introduce our case study "Alps Telecommunications" a simplified company from the telecommunication industry in order to facilitate the experiments. Then, we describe the experiments in order to illustrate the simulation and visualisation features of our extended version of xArchiMate, and finally, we discuss how we used them to simulate ICT services and visually analyse dynamic features of their "fit for use" attributes.

4.1 The corporate SMS delivery service

Alps Telecommunications provides a corporate SMS delivery service. For simplicity, we will only analyse the service for one fictitious destination, "Telco" (a mobile network that receives SMS). The service model is illustrated in Figure 2. The Telco SMS Delivery service is supported by an automatic activity that receives SMS delivery requests from channels of customers and uses vendors' channels to deliver these messages (a channel is an SMPP Short message peer-to-peer protocol Socket between the gateway of the company – the Traffic Management system software (TM SS) – and the partner's gateway). Each destination has one or more assigned vendors. A "heartbeat" is also included to monitor vendors: the Assign Telco Destination Vendor process and it is responsible of searching for new channels when one is unavailable.

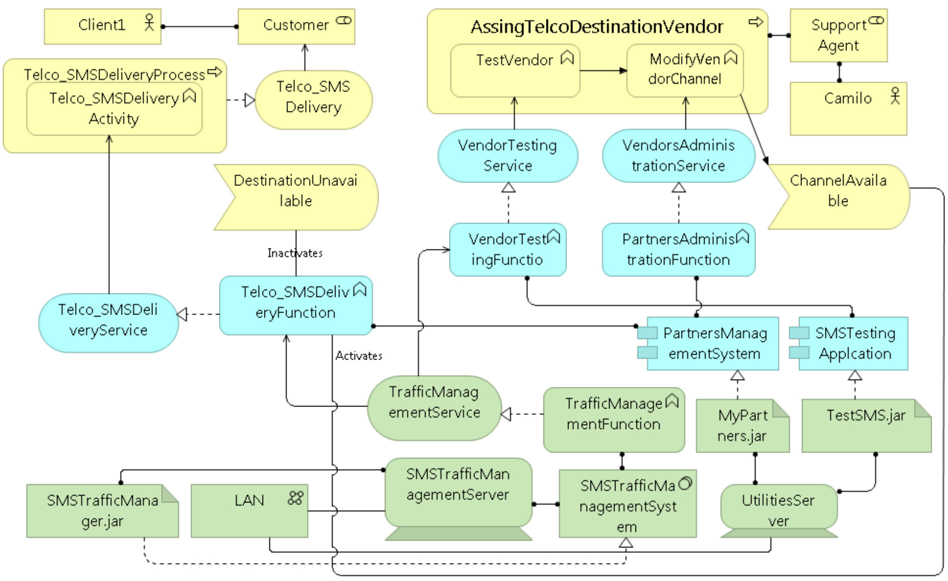


Fig 2: Telco SMS delivery service

Both the business service and the heartbeat are supported by a server (*Traffic Management* server) that stores the SMS TM SS, the core system responsible for managing the SMS traffic. The TM SS receives messages delivery requests from the *Telco SMS Delivery* function and from the *Vendor Testing* function. Messages are sent by the *Traffic Management* function that implements the *Traffic Management* infrastructure service. The *Vendor Testing* function is supported by the *SMS Testing Application* and the partners' administration function is implemented by the *Partners Management System*, an application that stores all partners' information and maps channels with destinations; it also implements the *Telco SMS Delivery* function. Both the *SMS Testing Application* and the *Partners Management System* are stored in the *Utilities* server and communicate with the TM SS through a *LAN*. In order to monitor the continuity of the service, the heartbeat or *Assign Telco Destination Vendor* process continually runs (experiment 1). In the simulation model, the destination unavailability is modelled by an external event, *Destination Unavailable,* which disables the *Telco SMS Delivery* function and triggers the *Assign Telco Destination Vendor* process (experiment 2). The *Channel Available* event is generated when an instance of the *Modify Vendor Channel* automatic activity is completed and when it is successful; the *Telco SMS Delivery* service is activated (if it is unavailable).

4.2 Visualisation of elements and experiment description

Table 2 shows the representations of active and behavioural elements used in the visualisation design. Device resources are depicted with three vertical bars at the left side: disk (red), memory (green) and threads (purple); next to the threads bar, there is also a label depicting the server requests counter; and at the right-hand side, a label denotes the server failures counter. The dynamic icon for a software consists of an aqua blue bar at the left side depicting the size of the pending-requests queue; next to this bar a label is located showing the number of current processing requests; at the right side, there is a label depicting the failures counter and an ellipse that depicts the system software state: available (green), unavailable (red) and retrieving (orange). A business service icon consists of an elliptical availability indicator located at the left-bottom corner; a label located on top of the icon shows the percentage of availability; and two labels at the right side show the last unavailability period. A process icon has two labels: the left-side label depicts the number of instances that have been invoked and the right one depicts the number of requests that have been rejected due to service unavailability. An icon for the application and infrastructure layers has an ellipse located at the right-bottom corner depicting their states (available/unavailable); the service includes one label depicting the number of instances that have been created. Instances are depicted by small circles that move along the relationships between the invoked element and the invoker element (Figure 3a). The *Traffic Management* service has a special icon designed to show the number of instances that have been invoked from both the *Telco SMS Delivery* function and *Vendor Testing* function (Figure 3a). An event icon consists of a label located at right-bottom corner depicting the total number of instances and two bars at the left side; the green bar depicts the proportion of successful events and the red bar depicts the proportion of unsuccessful events (Table 2). Event instances are depicted by circles that move between an event and the first triggered element; green circles depict successful instances while red circles depict unsuccessful ones (Figure 4).

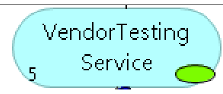
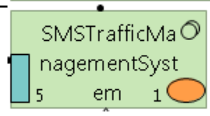
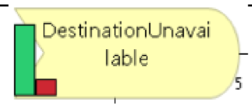
Active		Behavioural			
Device		Business service		Service and function	
Software		Process		Event	

Table 2: Visual representation

The first experiment consisted of processing 30 SMS delivery requests and 10 heartbeats within 5 seconds approximately (uniformly distributed). We assigned 10 threads of the *Traffic Management* server to process them; each request used 2 threads and the RAM and disk usage were considered insignificant. We wanted to analyse if 10 threads were sufficient to handle the workload and how the service would function in terms of its quality attributes. Out of the 30 requests, only 14 instances were completed because of a failure (Figure 3d, *"Telco SMS Delivery*" process); it is interesting to note the degradation in the processing time due to the limited resources and overload. In the sequence of images (Figure 3), we can observe the instances creation rate, the service availability, and the resources utilization; we can also see how loaded the system is (TM SS icon). At 10% of the total time (Figure 3a), 4 SMS delivery requests and 2 heartbeats had arrived; 5 of these were attended by the TM SS and 1 was buffered in the pending requests queue. At 25% (Figure 3b), the TM SS failed because it reached its maximum capacity and the service turned unavailable; at that time 3 requests had been completed, 5 were being processed and 5 were queued. At 45% (Figure 3c), the TM SS had recovered and had gone into the retrieving state in which it started processing all the pending requests (the 5 queued requests and the 5 that were being processed). At 65% (Figure 3d), the TM SS finished processing the pending requests and went into the available state again; the icon of the *Telco SMS Delivery* service shows that the service had an availability of 35.72% and that its last unavailability period was between 1568.17ms and 4390.15ms. We can also observe that 16 out of 26 service delivery requests and 5 out of 8 testing requests had been rejected. From the visualisation it is easy to identify that the assigned resources were not enough for this workload; the system availability was very poor; the processing time was high and too many requests were lost; and that the system was operating at its maximum capacity because the available threads were 0 all the time.

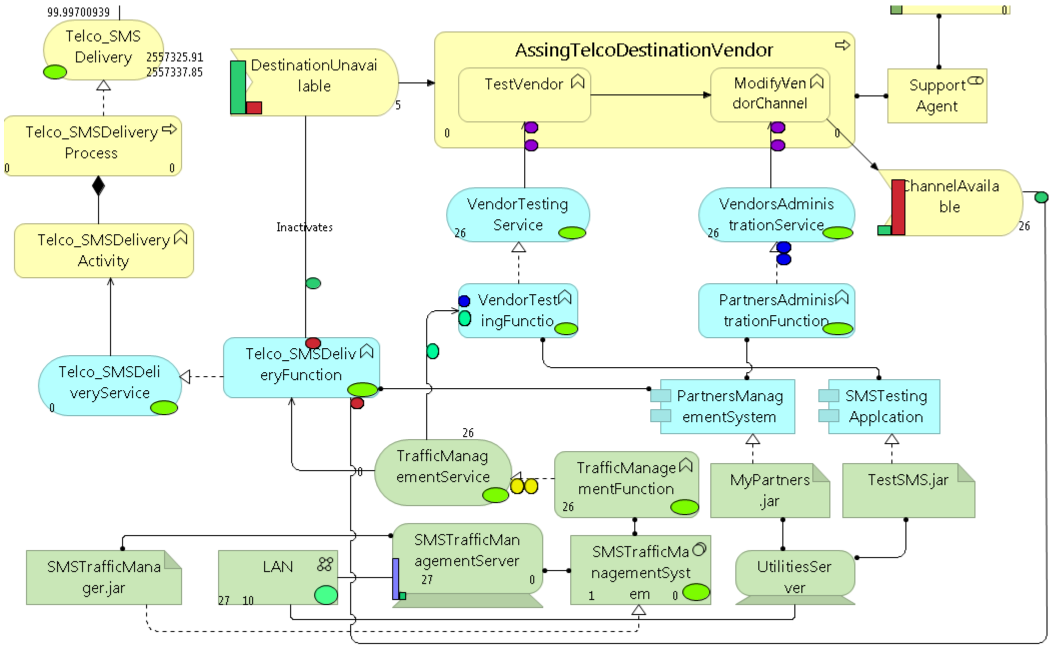
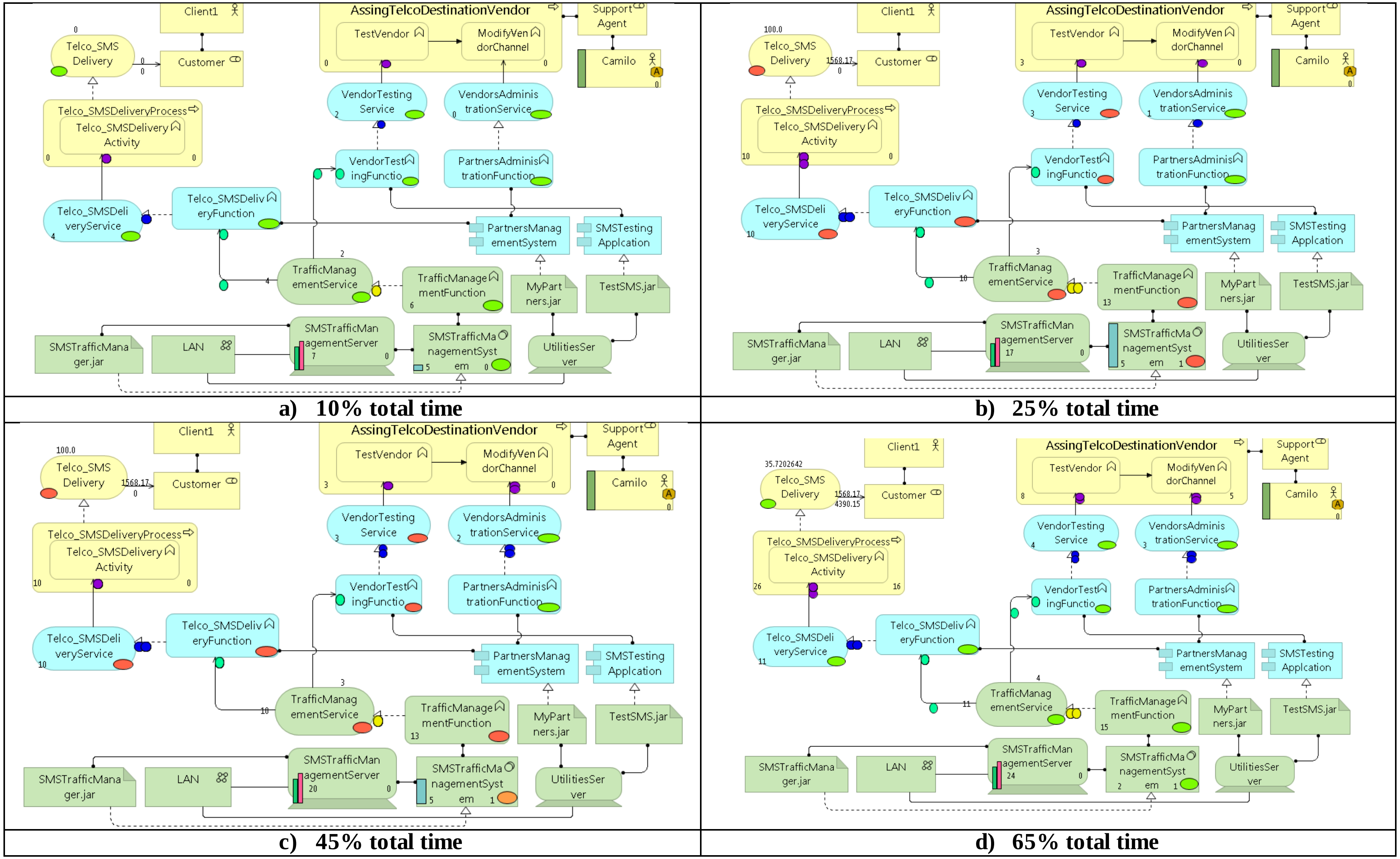


Fig 4: Experiment with events

In the second experiment we added an external event: the unavailability of the destination due to a lack of available vendors. This event affects the *Telco SMS Delivery* function by changing its state to unavailable and triggers the *Assign Telco Destination Vendor* process (if the event is successful). For this experiment the time interval was set to 10 minutes and we generated 6 *Destination Unavailable* event instances with a probability of success of 0.8 each. In order to handle the risk of *Destination Unavailable* events, the *Assign Telco Destination Vendor* process was instantiated, each time an instance of the *Modify Vendor Channel* activity finishes generates a *Channel Available* event instance indicating if the channel testing was successful, which means that a new available channel was found. Therefore, when a *Channel Available* event was successful, the Telco *SMS Delivery function* became available, and when unsuccessful, it triggered the *Assign Telco Destination Vendor* process again. The probability of success of each *Channel Available* event instance was 0.3. Figure 4 shows a screenshot of the visualisation of this experiment at the end of the animation.



a) 10% total time

b) 25% total time

c) 45% total time

d) 65% total time

Fig 3: Visualisation of the first experiment

We can observe the service availability variation because of the *Destination Unavailable* event; 4 instances were successful which means that the service failed 4 times and had an availability of 99.997%. We can also see the number of process instances that were created until one of them generated a successful *Channel Available* event.

These two experiments allowed us to observe, quantify and analyse the dynamic aspects of the simulated service over time. We were able to understand how, in the described scenarios, the service's attributes -capacity, availability and continuity– could be impact and based on these, we can make some conclusions: (1) the current configuration is not capable of handling the required workload and thus, it is not guaranteeing the service's continuity (experiment 1); and (2) the "heartbeat" monitoring vendors approach worked fine as a mechanism to maintain acceptable service's availability-levels. These initial results show an interesting potential of our extended version of xArchiMate as a decision support mechanism to evaluate ICT readiness for OR.

5. Conclusions

The described approach is useful for analysts and decision makers when there is a need for estimating the capacity, availability, continuity and security requirements, and their interdependencies, of an ICT service in order to strength OR capabilities. Specifically, simulation is a valuable alternative as it allows for the analysis and assessment of both initial ICT services architectures and their future states by (1) comparing alternatives, (2) identifying possible defects before going into production environments, and (3) providing information to answer what-if questions.

The first experiment we conducted allowed us to analyse capacity, availability and continuity attributes of an ICT service and to understand their behaviour and interdependencies. The second experiment analysed external threats, the preparations to handle them, and the strategies to improve the ICT service's availability and continuity. We are currently in the process of extending this work and experiment with more complex scenarios to measure other aspects of the "warranty of service" such as information integrity and/or confidentiality. We are also evaluating other possible xArchiMate extensions to manage artefacts, data objects, and business objects. Finally, it is important to emphasise that given our approach and the easy-to-extend set of concepts that xArchiMate provides, our extended version is fully capable of simulating and visualising a wide range of organisational scenarios as well as diverse ICT service sourcing models, for instance, ICT services in a "pay as you go" model. Thus, our extended version of xArchiMate represents a comprehensive experimental platform that supports decision-making processes related to defining, negotiating and agreeing on ICT service "fit for use" attributes.

References

Becker, S., Koziolek, H., & Reussner, R. (2007). *Model-based performance prediction with the palladio component model*. Paper presented at the International workshop on Software and performance.

BMC Software Inc. (2016). *Focus On: ITIL® Service Design For ITIL 2011*. Retrieved from http://www.bmc.com/forms/ESM_TL_ITIL_ReferenceBooks_BMCcom_EN_Apr2012_V2.html?PURL=283315281

Bosse, S., Schulz, C., & Turowski, K. (2014). *Predicting availability and response times of IT services*. Paper presented at the European Conference on Information Systems (ECIS).

British Standards Institute. (2011). BS ISO/IEC 27031:2011 Information technology. Security techniques. Guidelines for information and communication technology readiness for business continuity. Geneva, Switzerland.

British Standards Institute. (2014). BS 65000:2014 Guidance on organizational resilience.

Burnard, K., & Bhamra, R. (2011). Organisational resilience: development of a conceptual framework for organisational responses. *International Journal of Production Research, 49*(18), 5581-5599.
Caralli, R., Allen, J., Curtis, P., White, D., & Young, L. (2010). *CERT® Resilience Management Model v1.0: Improving Operational Resilience Processes* (CMU/SEI-2010-TR-012 / ESC-TR-2010-012). Retrieved from http://www.cert.org/resilience/products-services/cert-rmm/
Dutta, A., & Roy, R. (2003). *The dynamics of organizational information security*. Paper presented at the International Conference on Information Systems (ICIS).
Gogi, A., Tako, A., & Robinson, S. (2016). An experimental investigation into the role of simulation models in generating insights. *European Journal of Operational Research, 249*(3), 931-944.
Heinrich, R., Merkle, P., Henss, J., & Paech, B. (2015). Integrating business process simulation and information system simulation for performance prediction. *Software & Systems Modeling*, 1-21.
Herrera, A., & Janczewski, L. (2016). *Cloud Supply Chain Resilience Model: Development and Validation*. Paper presented at the Hawaii International Conference on System Sciences (HICSS).
Hill, T. (2011). *Software maintenance and operations hybrid model: An IT services industry architecture simulation model approach*. Paper presented at the International Conference on Research Challenges in Information Science (RCIS).
IBM Corporation. (2015). *Redefining Boundaries: Insights from the Global C-suite Study*. Retrieved from http://www-935.ibm.com/services/c-suite/study/
Jiang, Z. M., & Hassan, A. E. (2015). A Survey on Load Testing of Large-Scale Software Systems. *IEEE Transactions on Software Engineering, 41*(11), 1091-1118.
Kumar, M., & Bhat, J. (2009). *Discrete Event Monte-Carlo Simulation of Business Process for Capacity Planning: A Case Study*. Paper presented at the Pacific Asia Conference on Information Systems (PACIS).
Malek, M., Milic, B., & Milanovic, N. (2008). *Analytical Availability Assessment of IT Services*. Paper presented at the International Service Availability Symposium (ISAS).
Manzur, L., Ulloa, J. M., Sánchez, M., & Villalobos, J. (2015). xArchiMate: Enterprise Architecture simulation, experimentation and analysis. *Simulation, 91*(3), 276-301.
McManus, S., Seville, E., Vargo, J., & Brunsdon, D. (2008). Facilitated process for improving organizational resilience. *Natural Hazards Review*, *9*(2), 81-90.
Miller, H., & Engemann, K. (2014). Using reliability and simulation models in business continuity planning. *International Journal of Business Continuity and Risk Management, 5*(1), 43-56.
Orta, E., Ruiz, M., Hurtado, N., & Gawn, D. (2014). Decision-making in IT service management: a simulation based approach. *Decision Support Systems, 66*, 36-51.
Tertilt, D., & Krcmar, H. (2011). *Generic performance prediction for ERP and SOA applications*. Paper presented at the European Conference on Information Systems (ECIS).
The Open Group. (2016). ArchiMate® 3.0 Specification. Berkshire, United Kingdom.
Tjoa, S., Jakoubi, S., & Quirchmayr, G. (2008). *Enhancing business impact analysis and risk assessment applying a risk-aware business process modeling and simulation methodology*. Paper presented at the International Conference on Availability, Reliability and Security (ARES)
Wang, L., Sahai, A., & Pruyne, J. (2007). *A model-based simulation approach to error analysis of it services*. Paper presented at the IFIP/IEEE International Symposium on Integrated Network Management.