Summer 10-6-2011

REDESIGN AND PERFORMANCE OF SERVICE NETWORKS: A SYSTEMS DYNAMICS APPROACH

Martin Smits

Willem-Jan van den Heuvel

Christos Nikolaou

REDESIGN AND PERFORMANCE OF SERVICE NETWORKS: A SYSTEMS DYNAMICS APPROACH[1]

Martin Smits, European Research Institute in Service Science & Department of Information Systems, Tilburg University, The Netherlands (m.t.smits@uvt.nl)

Willem-Jan van den Heuvel, European Research Institute in Service Science & Department of Information Systems, Tilburg University, The Netherlands (wjheuvel@uvt.nl)

Christos Nikolaou, Transformation Services Lab, University of Crete, PO Box 71003, Heraklion, Greece (nikolau@tsl.gr)

Abstract

Continuous innovation of service networks is important for the success of modern service based economies. Over the last years, several analysis and design methods have emerged for engineering SOA applications. However, none of these methods can effectively cope with the increased level of complexity and dynamics that come with service network innovation. In addition, and even more problematic, existing methods fall largely short in assisting application designers and business managers in evaluating and predicting the impact of design decisions on the performance of the service-enabled applications and the end to end business processes at the network level. In this paper we use system dynamics simulation to predict the impact of design decisions for software services and human-operated services on service network performance in the automobile industry. We show that a single design decision at the software level influences 10 out of 12 performance indicators at the firm and the inter-organizational business network level.

Keywords: Service design, Business Network Performance, System Dynamics, Simulation

[1] The research leading to these results has received funding from the European Community's Seventh Framework Program under the Network of Excellence S-Cube grant agreement 215483.

1 Introduction

The growth of services economies and the evolution of powerful digital communication networks help transform service companies from local businesses to globally integrated service networks, also referred to as service networks (Caswell et al, 2008). Service networks are open, complex and fluid, socio-economic systems of organizations and processes that break away from classical hierarchies of knowledge and power, to accommodate the co-production of new knowledge and services through organic peer-to-peer interactions (Basole and Rouse, 2008). For this purpose, service networks embody end-to-end processes that are layered on services that providers provide and clients consume, and that may be connected at a global scale (Maglio et al, 2007; Maglio and Spohrer, 2008).

Service oriented computing (SOC) has been heralded as a natural technology candidate to develop and manage service networks as choreographed, event-driven software and human-operated services that collectively realize end-to-end business processes (Taylor et al, 2009). Over the last years, several SOC methods have emerged for engineering service-enabled applications. However, none of these methods were designed with complex inter-organizational service networks in mind, and are based on closed-world assumptions that assume that service applications have clear boundaries and will be executed in fully controlled, homogeneous, predictable and stable execution environments (Van den Heuvel, 2009). These SOC methods thus cannot be expected to effectively cope with the increased levels of complexity and dynamics of service networks that typically exhibit non-linear, non-deterministic and unpredictable behavior. In addition, and even more problematic, existing SOC methods largely fall short in assisting application designers and business managers in evaluating the impact of design decisions on the performance at the level of service-enabled applications and the business processes they support at the business network level.

Innovation of service networks is poorly understood (Gadrey and Gallouj, 2002). Maglio and Spohrer (2008) argue that business, social, and technological knowledge must be combined to explain the growth and development of service systems and to solve fundamental problems such as how to invest optimally to improve service productivity and quality. The aim of this paper is to add to current analysis and design methods for service networks in such way that the impact of service design decisions on service network performance can be predicted. We focus on the interactions between service designers and business managers during the service design process. Our research focuses on the innovation of service systems, in particular on how to predict effects of service design choices on service network performance at different levels of aggregation.

The paper is structured as follows. We first summarize background theory on service network innovation and performance analysis in section two. We then introduce system the dynamics simulation method that we apply to a case that we take from the automotive industry. We simulate the case to analyze the impact of service design options in the car repair process, which is an end-to-end business network level process in the automotive industry. Section 4 introduces the simulation model and the service performance analysis. Finally, section 5 summarizes the key findings and plots the path for future research.

2 Service Network Innovation and Performance Analysis

Service systems, also known as service networks, are value creating networks composed of people, technology, and organizations (Maglio et al, 2006; Caswell et al, 2008). Innovation of these networks occurs through an innovation process that consists of adapting technologies to support business behaviour as well as adapting business behaviour to be able to use the technologies. These technology – social adaptation processes are known as social shaping of technology (SST) (Williams and Edge, 1996). Several SST theories have emerged in the 1980s such as the Social Construction of Technology framework (SCTF), Actor-Network Theory (ANT), and the Socio-Technical Theory (STT) (Mumford, 2000).

Socio-Technical Theory distinguishes between three subsystems (of a service system): (i) the technical subsystem of technologies and tasks, (ii) the social subsystem of structure and people (employees, knowledge, skills, attitudes, values, needs, reward systems and authority structures), and (iii) the environmental subsystem, including the wider range of the organization including customers, suppliers, rules and regulations which govern the organization at large. SST assumes that organizational systems will maximise performance only if the technical, the social, and the environmental subsystems are designed to work in harmony (Bostrom and Heinen, 1977).

SST theories have a common feature of criticism of the linear model of innovation and technological determinism. SST differs from these notably in the attention it pays to the influence of the social and technological context of development which shapes innovation choices. SST is concerned to explore material consequences of different technical choices, but criticises technological determinism, which argues that technology follows its own developmental path, outside of human influences, and in turn, influences society. In this way, social shaping theorists conceive the relationship between technology and society as one of 'mutual shaping' (Williams and Edge, 1996).

According to Williams and Edge (1996), central to Social Shaping of Technology (SST) is the concept that there are choices inherent in both the design of individual artifacts and systems, and in the direction or trajectory of innovation programs. Different routes are available, potentially leading to different technological outcomes. Significantly, these choices could have differing implications for society and for particular social groups.

We use the SST perspective to study design choices of managers (in the business or social subsystem) and service engineers (in the technical subsystem) in service network innovation. Such design choices by different actors may dynamically interact and may lead to unexpected outcomes. Design choices are taken in particular phases of the service design process, as explained in the next section.

2.1 Service Network Design

Many SOA lifecycle methods have been introduced over the past decade, embodying the well-known phases of the software engineering lifecycle. Prominent examples of these methods include the Service Lifecycle Process (SOA guide 2006), the Service-Oriented Modelling Framework, the Mainstream SOA Methodology (Bell, 2008), and the Service-Oriented Modelling and Architecture (Arsanjani, 2004). Service-oriented design and development methods for service networks are typically based on an iterative and incremental process. Figure 1 shows a generic Service Network Engineering Lifecycle model (Zur Muehlen and Shapiro, 2009) comprising five main phases that may be traversed iteratively catering for service network design centered on performance analytics. The phases are: modeling (network analysis and design), implementation and testing, deployment and execution, analyzing and monitoring, and measuring and optimizing.

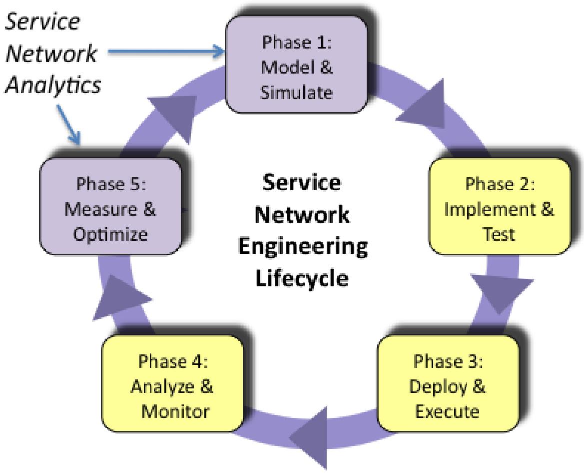


Figure 1 Adapted Service Network Engineering Lifecycle model (Zur Muehlen and Shapiro, 2009)

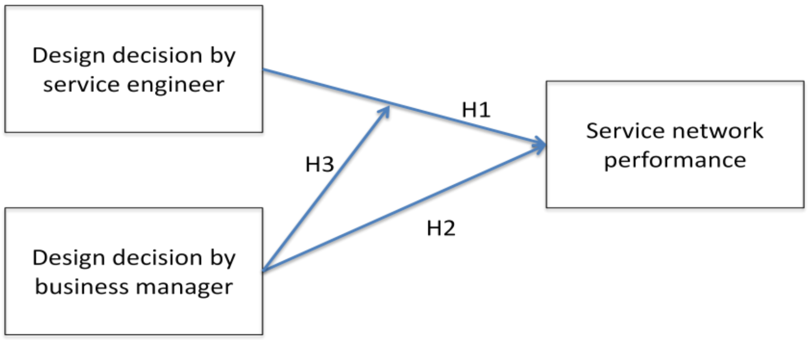
The first phase produces a logical and physical design of the service network. First, the service designer starts with conceptualizing the network in terms of the network partners (with different, sometimes contrasting objectives), the end-to-end processes that live within the network, and choreographed software/human services that implement them. The logical design typically entails abstract models of the process and service choreography rendered in conceptual notations such as the BPMN-2.0 Business Process Diagram, Collaboration Diagram and Choreography Diagram. Ideally, these models are calibrated to meet performance requirements for the end-to-end process in terms of Key Performance Indicators (KPIs), and, supporting service resources in terms of Quality-of-Service (QoS). In phase two, the logical design of the service network is refined into a physical design, typically relying on well-know standards from the WS-stack, such as WS-Policy, WSDL and BPEL. Service network implementation and testing involves coding or identifying reusable service resources and choreographing them into end-to-end processes using the physical specifications. It also involves testing coded services and processes for functional correctness and completeness as well as for interoperability. The service network deployment and execution phase continues enforcing the business model for service provisioning; addressing issues service metering, service rating and service billing. Once the provisioning model has been established, the service network may be deployed recursively, involving deployment of human-operated and software (Web) services by all partners in the service network. Execution includes the actual binding and run-time invocation of the deployed choreographed services. The next phase involves monitoring and analyzing the execution of the service network, resolving potential process and service anomalies including unforeseen interoperability conflicts. Lastly, progress of executing end-to-end processes in the service network are measured against performance metrics, such as KPIs, and optimized on an as-needed basis.

We adapt the Service Network Engineering Lifecycle in two ways to take into account both technical and business performance concerns in service networks. Basically, we leverage the conventional approaches used in phase 1 (design time) and 5 (run time) with two types of performance analytics of service networks: design-time and run-time. Design-time service performance analytics utilizes conceptualizations, the logical models, of service networks to verify their performance against agreed-up service levels of partner-level and network-level processes. Run-time service performance analytics study event logs that are provided by service monitoring tools, and measure progress of end-to-end processes against performance metric, and proactively pinpoint areas for process improvement and troubleshoot the root-cause of bottlenecks. In the remainder of this article only design-time (phase 1) performance analytics will be considered including interactions between design decision makers in a service network. In a later stage of research we may also include run-time performance analytics and (automated) redesign and adaptation of (web) services in order to realize self-adaptive systems.

2.2 Service Network Performance

Performance of a service network refers to the ability to accomplish service level objectives at the level of service resources, including human-operated and software services, and strategic business objectives at the level of the end-to-end processes that live within them. Business performance is often assessed using a balanced scorecard in which traditional financial performance measures are supplemented with criteria that measure performance from three additional perspectives: those of customers, internal business processes, and learning, growth and innovation (Kaplan and Norton, 1992). The use of a balanced scorecard including multiple performance indicators supports alignment between business and technology subsystems, organizational learning, and improved business outcomes (Chenhall, 2005).

In the remainder of this article we determine the set of performance indicators used in a service network at different management levels and determine the impact of design-time decisions to (re-) design services in a service network. The conceptual model that we test is given in figure 2 Independent variables are design decisions by service engineers and business managers may impact service network performance (H1 and H2). Also, design decisions by service engineers and business managers may interact and influence service network performance (H3).



Figuur 2 Conceptual model of influence of design decisions on service network performance.

3 Research Method

We use simulation to predict the impact of combinations of design decisions on service network performance. Computer simulation is a research method that allows researchers to answer "what if" questions. Simulation enables studies of complex systems by creating predictions, whereas other research methods in principal look backwards to determine sequential relations (as in process research) or relations between factors (cross-sectional research) (Langley, 1999). System dynamics modelling has been used frequently to solve management problems. Notably, over 1500 publications have been identified to solve management problems with system dynamics in health care (Brailsford, 2008) and even more for policy modelling in supply chain management (Sterman, 2000). An and Jeng (2005) applied system dynamics simulation successfully for the purpose of Web services management. In particular, they introduced an automated Web service management system that employs feedback loops and service adaptation to ensure SLA requirements are continuously met. Our approach uses a similar tactic, alleviating the problem of resource (re-) allocation for singular Web service implementations to the level of service networks.

Simulation is a powerful, rigorous yet practical suite of methods and tools that help to analyze and predict qualitative and quantitative effects on service systems. Simulation not only helps to better understand and manage service systems at large, but also the processes that embody them as well as their supporting information systems. Simulation plays a critical role in the analysis, design and management of service networks (Forrester 1961, 1994; Sterman, 2000). In particular, simulation helps to identify performance leakages, better understand and explore the impact of change scenarios, to test and verify compliance towards resource constraints and business rules and goals, and to assess risk by examining operational impact, i.e., timeliness and quality, on the network.

Two mainstream simulation models have emerged that combine several of the above basic simulation models, viz. discrete-event dynamic systems (DEDS) simulation models and system dynamic (SD) simulation models. DEDS deals with individual events, such as a customer request or the shipment of a product, and can deal with uncertainties. A third simulation technique is agent based modelling (anylogic, see www.xjtek.com). SD was developed in the 1950s (Kleijnen and Smits, 2003) and promotes a dynamic, continuous deterministic simulation approach from an aggregated, non-discrete perspective. In its basic form, system dynamics analyzes positive and negative feedback loops and emerging behavioural effects –such as exponential growth or decline– that result from them. Typically, dynamic behaviour in service networks manifests itself as oscillating behaviour where corrective actions force the network to a steady state where performance is tuned between end-to-end processes and supportive service resource.

In this research, we adopt system dynamics simulation (in Vensim software, www.vensim.com) for three reasons. First, combining design decisions by multiple actors is "dynamically complex", meaning that service design decisions in (end-to-end) processes may provoke planned and unplanned consequences, which cannot be easily predicted without the help of a computer simulation model.

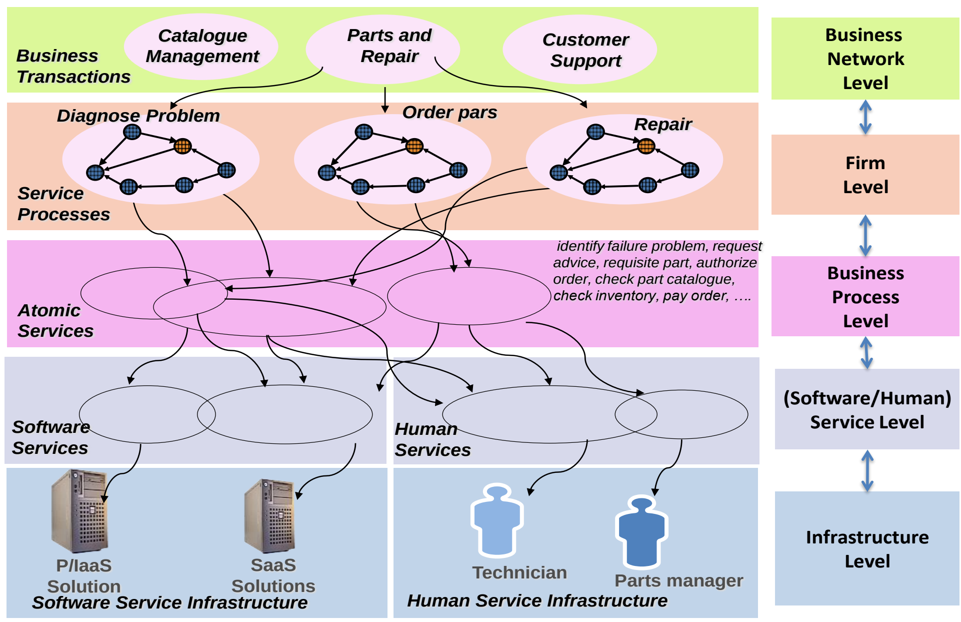
Second, the multi-actor design problem is long term, meaning that effects of service design decisions may not appear immediately but only after some weeks/ months. Third, while service networks are in fact dynamic systems of systems, it is only natural to conceptualize them in terms of flow processes (end-to-end processes, service processes, resource management) like in SD. These three reasons makes simply relying on discrete event simulation –as done by many process modelling simulators- largely inappropriate to measure and tune performance that permeates service networks at the logical and physical level.

A key stage in the development of a (system dynamics) simulation model is validation of the model. We did structural validation (does the model reflect reality and are key feed-back mechanisms included?) and numerical validation (do simulation runs with the model produce output values similar to known data sets?), following guidelines by Peck (2010).

3.1 Automobile Industry

We apply simulation analysis on a car repair service network in the automobile industry. This service network case is taken from Caswell et al (2008) and Saimaresh et al (2004). We use their descriptions of the structure and the quantitative properties of the (Chrysler) service network. We focus on the car repair and car parts delivery end-to-end process as shown in Figure 3. At the business network level, business services fulfil business transactions like customer support service, parts and repair service, and catalogue management services. At the firm level, service processes are designed and executed to diagnose a car problem, to order parts or to repair a car. One level deeper, at the business process level, atomic services are designed and executed to fulfil the firm level service processes. Atomic services are implemented as software services (P/IaaS or SaaS) or as human services (by repair technicians, parts managers, etc.).

Note that performance of the service network can be determined (with various indicators) at all levels. Also, key decisions on service design are taken on all levels, like ‘which human services will be automated’ and ‘which fine grained services will be combined into course-grained aggregated services and end-to-end business processes at the business network level’.

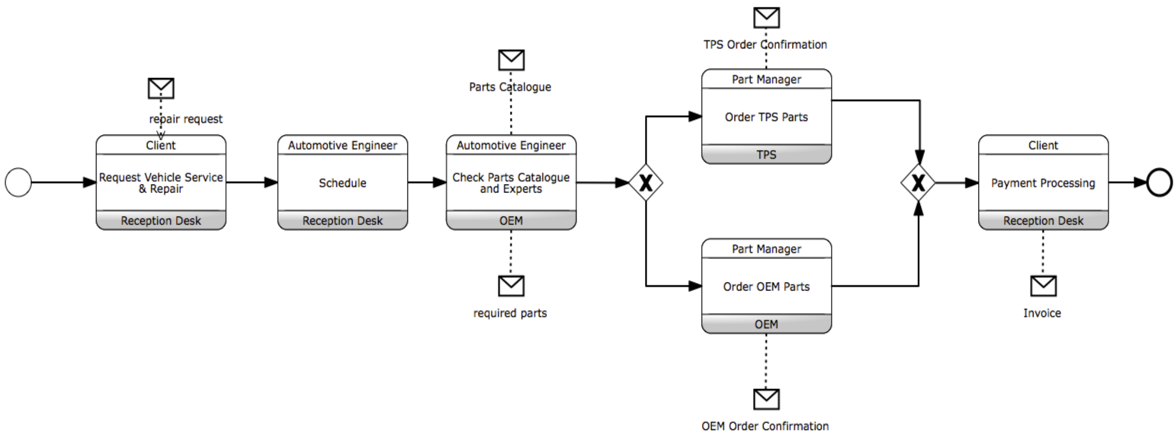


Figuur 3. Overview of the car repair service network.

4 The Car Repair Case

The car repair network basically links four types of participants: an Original Equipment Manufacturer (e.g. Volvo), OEM-franchised Car Dealers, Third Party Parts Suppliers (TPS), and Clients. OEM-franchised car dealers may service and repair cars for their clients. Both activities require a car parts catalogue to ensure that repairs can be performed efficiently either in the replacement of parts or repairing after accidents. The car catalogue facilitates efficient installation, operation and lifecycle maintenance of intricate products describing detailed part information that can be fully integrated with other service applications supporting customer support processes, human resource management, and other service provisions. The quality of the OEM parts, the OEM car catalogues, and OEM support services influences the fraction of parts from OEM or TPS. Figure 4 depicts a simplified version of the choreography for car repair in BPMN 2.0 syntax. Note that this model does not allow simulations.

Vehicles are booked in 15-minute time intervals to allow customers to discuss their needs with fully trained automotive engineers. The engineers then inspect the car on the hoist and diagnose and report the car service requirements that may include replacing teardowns, warranty replacements and collision repairs. On the basis of the car diagnosis, a cost estimate will be computed and communicated to the client for authorization. Once authorized the automotive engineer will scrutinize failure symptoms, detect faulty parts, order parts and perform the repair. Ordering parts is a complex process that involves asking advice from expert technicians from the OEM, including acquiring information about parts under warranty, and getting approval from the dealer's part manager. The part manager then checks local inventory for the required part, and if necessary checks the stock at the OEM or other supplier stocks, and eventually places an order. Based on the quality of the parts catalogue and the expert advice, the parts manager may either use third-party suppliers or suppliers from certified supply-chain suppliers. Also, the quality of these services may also influence the price, speed, and other properties of the car repair service to the end customer.



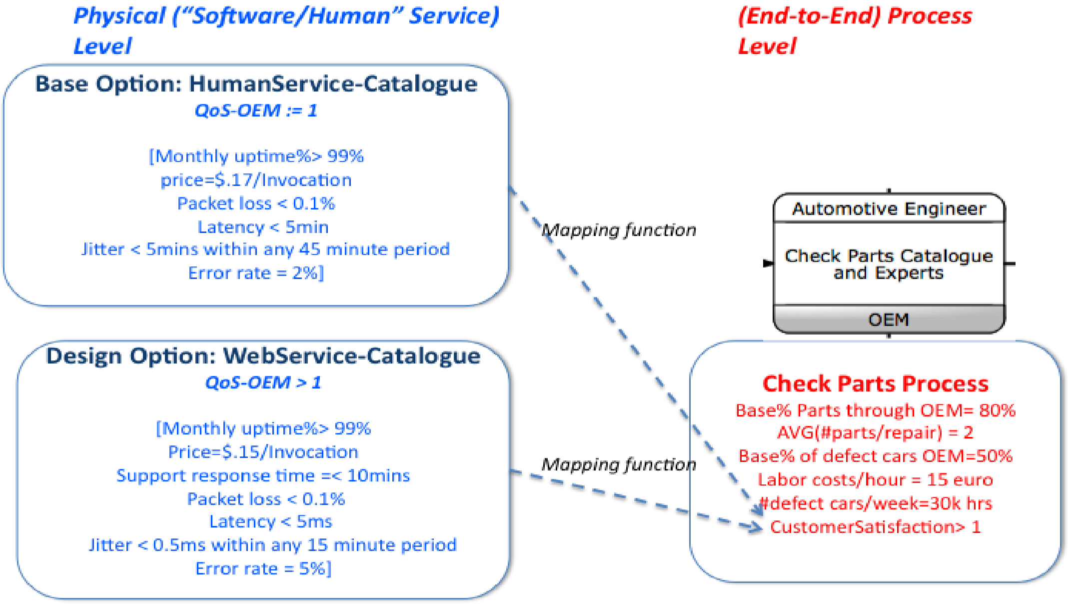
Figuur 4 Overview of the end-to-end car repair process (BPMN choreography model)

The automotive engineers spend on average one hour/day determining which parts to order, whilst the part manager loses roughly 30 minutes per day checking local inventory and ordering parts that are out of stock. While logically designing the car repair service network, the service engineer faces many design challenges about which network partners to involve, how to abstractly choreograph their services into end-to-end processes, and which IT or human services to utilize. Each design decision may affect performance at the level of the (end-to-end) processes as well as the level of the service resources.

4.1 Service Design Decisions

In the context of the running example, the service engineer needs to decide (for example) on which design option yields the optimal customer satisfaction (performance) for the service network: reinforcing human (experts) help desk services of OEM, e.g., through additional human resources (base option), or automating the OEM parts catalogue with Web services for faster and more accurate information about car parts (web service option).

Both design options, and their ramifications for the performance at the level of the service resources, the firm level, and the (end-to-end) service network level have been illustrated in Figure 3. As Figure 2 illustrates the mapping between performances at both levels, the mapping function- signifies the key challenge. In particular, the dotted directed lines in Figure 3 depict mapping between the Quality of Service of the two design options to the aggregated Customer Satisfaction metric associated to the check parts process. The mapping function aggregates quality of service of operational software and human-operated services, into strategic, process-related customer satisfaction



Figuur 5 Two design options in the CheckParts process in the Car Repair Network

4.2 Business Network Simulation Model

Figure 5 shows an overview of the system dynamics simulation model. The main building blocks of SD models are *stocks*, represented as rectangles, *inflows and outflows* (incoming and outgoing directed pipes), *valves* that control flows, and *clouds* that render sources and sinks. Behind this graphical model lie 50+ mathematical functions, specifying how valves influence flow and stock variables.

The SD model consists of four sections defined around the four stocks in the service network: OEM value (upper left in Figure 5), OEM dealer value (upper right), OEM dealer satisfaction on OEM (lower left), and customer satisfaction on OEM dealers (lower right). Note that, like in a supply chain, the upper sections indicate financial flows (costs and revenues) from right to left and the bottom sections the flow of services from left to right to the end customer. The four sections are logically chained into three processes: the Catalogue Management Process (CMP), the Parts and Repair Process (PRP, cf. Figure 2) and the Customer Support Process (CSP).

OEM value is a stock that accumulates OEM total revenues per week (inflow rate) and OEM total costs per week (outflow rate). OEM value will rise if the OEM total costs decrease or OEM total revenues increase. The model shows that OEM total costs are influenced by four variables: investments in parts catalogue services and human services, production costs per OEM part and the total number of OEM parts. OEM total revenues are influenced by number of OEM parts used per car repair and OEM price per part and number of car repairs per week.

Dealers' satisfaction on OEM is a stock that increases if OEM parts catalogue services and human services together outperform the quality of TPS services. The quality of OEM services is influenced by the decision whether to leverage catalogue service quality with Web services (new design option) or that of helpdesk services (base option).

OEM quality of service and dealer satisfaction on OEM influence the parts and repair process (PRP) through its impact on the total repair time per car and the number of OEM parts used per car repair.

Customer satisfaction on the car repairs of OEM dealers constitutes the third stock in the model. This stock increases if customer satisfaction per OEM repair (inflow) outperforms customer satisfaction per TPS repair (outflow). Customer satisfaction per OEM repair is an index value, calculated from total repair time (versus expected total repair time), total price for repair service per car (versus expected price), and service hours per client (versus norm service hours per client).

Lastly, OEM dealers' value makes up the fourth stock in the model. This stock accumulates OEM dealers' total revenues per week (inflow) and OEM dealer total costs per week (outflow). OEM dealers' total revenues depend on number of car repairs per week and the total price for repair services per car. OEM dealer total costs depend completely on two control variables (labour costs per hour and personnel hours available per week).

Validity of the simulation model was checked by performing structural validation (by comparing the model structure with the descriptions of Caswell et al (2008)) and numerical validation by running the model with Caswell input data and checking the output with results given by Caswell et al (2008).

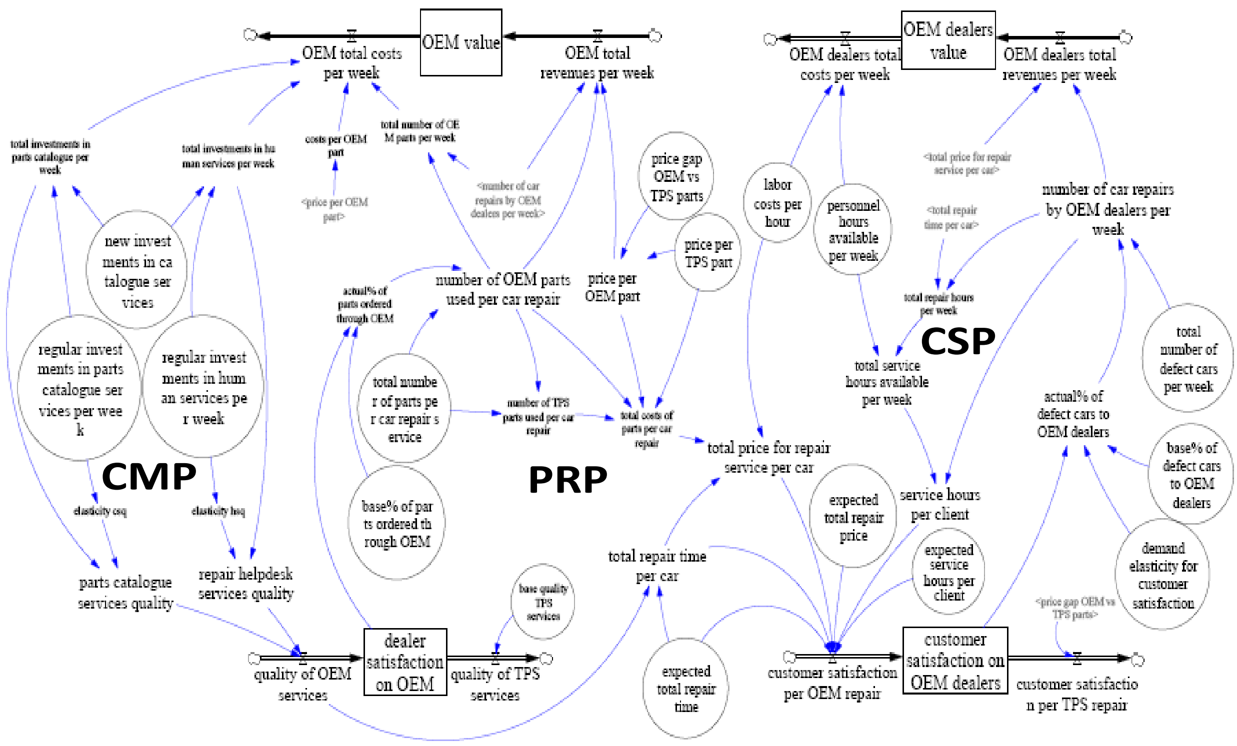


Figure 5. System Dynamics Model of the Car Repair Service Network

4.3 Service Performance

The circled variables in the SD model signify the 17 control variables (independent variables), the main inputs in the SD model. These inputs are listed in Table 1, including the (single) design decision for the service engineer as specified in Figure 4. For reasons of simplicity, the current simulation only considers one design variable that can be adjusted by the service engineer: the decision whether or not to design a web service catalogue (extra investment in software catalogue services per week) that improves Quality of Service of the OEM. Also, the simulation considers only one decision by the business manager: to keep total investments in catalogue plus human services constant. The base investment for catalogue services is 21k euro per week and 26k euro for human services per week. In the new design situation, 10k euro additional investment are made in software catalogue services (total 31k euro) and investments in human services are reduced to 16k euro.

Actor level	Independent (control) variables	Base value	New design
OEM	Extra investment in software catalogue services pr wk	0 euro	10000 euro
	Regular investment in human services p wk	26000 euro (*)	16000 euro
	Regular investments in catalogue services p wk	21000 euro (*)	No change
	Purchase costs per OEM part (% of sales price)	60% (*)	No change
	Price gap for OEM parts (relative to TPS parts)	2 (*)	No change
Car Repair Network	Total number of parts per car repair service	2 (*)	No change
	Base% parts ordered through OEM	80% (*)	No change
	Price per TPS part	15 euro (*)	No change
	Base% of defect cars to OEM dealers	50%	No change
	Total number of defect cars per week	25000 (*)	No change
OEM dealer	Labor costs per hour	50 euro (*)	No change
	Personnel hours available per week	30000 hrs (*)	No change
	Base repair time per car (expected repair time)	2 hours (*)	No change
Customer	Expected total price for OEM repair service	154 euro (*)	No change
	Expected repair time (= base repair time)	2 hours (*)	No change
	Expected service hours per client	0.35 hours	No change
	Demand elasticity for satisfaction	1	No change

Table 1. Control variables in the car service network on four actor levels. (*) values based on Caswell et al (2008).

Running the SD model with these inputs results in table 2, showing the 12 main outputs (dependent variables) for performance indicators at different actor levels (four indicators for OEM performance, four indicators for OEM dealer performance, and five for service network performance). Column BSC (Balanced Scorecard) identifies the indicator type (F = financial, C = customer, and P = process).

Actor level	Dependent variables	BSC	Base value	New design
OEM	Quality of OEM services (index variable)	P	1	1.045
	Dealer satisfaction on OEM (index value)	C	1	5.578
	OEM total revenues per week (euro)	F	612000 (*)	799980
	OEM total costs per week (euro)	F	414200 (*)	526988
Car Repair Network	Total price for repair service per car	F	154 (*)	156
	Total repair time per car (hours)	P	2 (*)	1.9
	Customer satisfaction per car repair (index)	C	3	3
	Customer satisfaction on OEM dealers (index)	C	1	3.3
	Total revenues (dealers +OEM) (euro)	F	66.385.000	83.433.000
OEM dealer	Number of car repairs by OEM dealers p week	P	12750 (*)	13333
	Total repair hours per week (hours)	P	25500 (*)	25498
	OEM dealer total revenues per week (euro)	F	1.965.000 (*)	2.074.000
	OEM dealer total costs per week (euro)	F	1.500.000 (*)	1.500.000

Table 2. Dependent variables in the car service network on three actor levels. (*) values based on Caswell et al (2008). Column BSC (Balanced Score Card) denotes the indicator type.

Table 2 shows that the additional investments in Web service catalogue leverage 10 out of 12 outputs. For instance, OEM QoS raises from the normalized base index value 1 to 1.045, leading to higher customer satisfaction (from the normalized base value 1 to 3.3), and aggregated network value (revenue), pointing towards a Go-decision for the service engineer.

The current simulation with new design values for 2 out of 17 control variables supports hypotheses H1 (service engineer decision affects performance indicators at all levels and dimensions), H2 (business manager decision affects performance indicators at all levels and dimensions), and H3 (service engineer decision and business manager decision interact). Additional analysis of the impact of design decisions (changing control variables in Table 1) may include more combinations of design decisions by service engineers and business managers. Future research will include designed experiments with such system dynamics models.

5 Conclusions and Outlook

Performance analysis of service networks will become increasingly important where service resources and processes can continually morph themselves to respond to environmental demands and changes. In this paper, we have investigated the relations between service design decisions and service network performance using a system dynamics simulation approach. We simulated the design decision described by Caswell et al (2008) and determined the effects of the design decision on 12 dependent (Balanced Scorecard) performance indicators. We found effects on 10 indicators at the firm level as well as on service network level.

The aim of this research was to improve our understanding of service design decisions on service network performance. In particular, we have introduced an analytical model that entails predicting the long-term impact of design decision on network performance as well as service resource allocations. In contrast to existing service engineering methods, we present a method that revolves around an analytical model to take into account performance of end-to-end processes and supporting service resources (human and software services) in service networks.

Clearly, the results presented in this article are first results in nature. Improvements and extensions of the model and analysis of design impacts are needed in various directions. First, we plan to further harness the analytical model using several other case studies. Currently, we are studying three cases from the domain of healthcare and telecom to reinforce the model using actual data. Second, we will formalize the mapping of abstract models of service networks in BPMN 2.0 Choreography and Collaboration diagrams to system dynamics models. Third, we envision developing techniques and tools for runtime performance analytics. This will imply an integration of our approach with Web service monitors and process mining tools, such as ProIM (Santillo et al, 2007). Finally, this new method needs to be validated against reality, which ought to be part of the new research agenda.

References

An L and Jeng JJ (2005): Web Service Management using System Dynamics. Proceedings of IEEE International Conference on Web Services (ICWS'05), 347-354, IEEE Press.

Arsanjani A (2004): Service-Oriented Modeling and Architecture (SOMA), IBM developer Works, www.ibm.com/developerworks/webservices/library/ws-soa-design.

Basole RC and Rouse WB (2008): Complexity of service value networks: conceptualization and empirical investigation, IBM Systems Journal 47 (1) 53-70.

Bell M (2008): Service oriented modelling (SOA): Service analysis, design, and architecture, Wiley, 2008.

Bostrom RP, Heinen JS (1977): MIS problems and failures: a socio-technical perspective. MISQ 1 (3) 17-32.

Brailsford SC (2008). System Dynamics: what's in it for health care simulation modelers? In: Mason SJ, Hill R, M¨onch L and Rose O (eds). Proceedings of the 2008 Winter Simulation Conference, Miami, FL, pp 1478–1483.
Caswell NS et al. (2008): Estimating value in service systems, IBM System Journal 47 (1) 87-100.
Chenhall RH (2005): Integrative strategic performance measurement systems, strategic alignment of manufacturing, learning and strategic outcomes: an exploratory study. Accounting, Organizations and Society 30: 395-422.
Forrester JW (1994). System dynamics, systems thinking, and soft OR. System Dynamics Review 10(2–3): 245–256.
Forrester, JW (1961): Industrial Dynamics, Pegasus Communications.
Gadrey J and Gallouj F (2002): Productivity, innovation and knowledge in services: New economic and socio-economic approaches. Cheltenham, UK: Edward Elgar.
Jin LJ, Machiraju V, Sahai A (2003): Analysis on Service Level Agreement of Web Services, HP Technical Report.
Kaplan R and Norton D (1992): The balanced scorecard: Measures that drive performance, Harvard Business Review, 71—79, January/February.
Kleijnen J and Smits M (2003): Performance metrics in supply chain management, Journal of Operational Research Society (54) 507-514.
Langley A (1999): Strategies for theorizing from process data. Ac. Of Management. 24 (4) 691-710)
Maglio PP and Spohrer J (2008): Fundamentals of service science. J. of the Acad Mark. Sci. (36) 18-20.
Maglio PP, Srinivasan S, Kreulen JT, Spohrer J (2007): Service systems, service scientists, SSME, and innovation. Communications of the ACM (49) 81-85.
Mumford E (2000): A socio-technical approach to systems design. Requirements Eng 5: 125-133.
Peck C (2010): The Value of Critical Thinking Skills: Modeling Price and Inventory Dynamics. In: Tracing Connections: Voices of Systems Thinkers. Isee Systems, Acton (MA)
Sairamesh J et al (2004): Dealer collaboration: transforming the value chain through integration and relationships, Proceedings of International Conference on E-Commerce Technology, 325-239.
Santillo L (007): Seizing and Sizing SOA Applications with COSMIC Function Points, Proceedings of the Software Measurement European Forum (SMEF), 156 – 165, Rome.
SLA Management Handbook (2004), volume 4, The Open Group, Enterprise Perspective.
SOA guide (2006): SOA Practitioners' Guide Part 3: Introduction to Service Lifecycle. www.healthmgttech.com/editorial_whitepages/SOAPGPart3.pdf.
Sterman JD (2000): Business Dynamics: Systems Thinking and Modeling for a Complex World, McGraw-Hill.
Taylor H et al. (2009): Event-driven architecture: How SOA enables the real-time enterprise, Addison-Wesley.
Van den Heuvel WJ (2009): Changing the Face of the Global Digital Economy, Smart service systems as a Catalyst for Innovation. Tilburg University Press..
Van den Heuvel WJ and Papazoglou MP (2010): Toward Business Transaction Management in Smart Service Networks. IEEE Internet Computing. Pp. 70 – 75. IEEE Press.
Zur Muehlen M. and Shapiro R. (2009): Business Process Analytics. In: Handbook on Business Process Management (eds Rosemann,M. et al.) Vol. 2, Springer Verlag, Berlin.