

Inciting Networks Effects through Platform Authority: A Design Theory for Service Platforms

Completed Research Paper

Ulrich Scholten

Karlsruhe Institute of Technology
Englerstr. 11, 76131 Karlsruhe
ulrich.scholten@kit.edu

Christian Janiesch

Karlsruhe Institute of Technology
Englerstr. 11, 76131 Karlsruhe
christian.janiesch@kit.edu

Christoph Rosenkranz

Goethe University Frankfurt
Grüneburgweg 1, 60323 Frankfurt am Main
rosenkranz@wiwi.uni-frankfurt.de

Abstract

The provisioning of software has shifted towards a service model, where applications are hosted, metered, and billed through service platforms rather than being sold as on-premise software. These services can exhibit network effects, which are self-enforcing effects of value generation. They describe the interdependency of service or platform value, service consumption, and third party service provisioning. Service platforms need to be designed to incite these network effects. We propose a design theory which is based on the operator's platform authority as the key factor for controlling and influencing stakeholders in and around the service platform in different control modes. As a design method, we propose a conceptual modeling language which supports service platform design with staged areas of authority, process flows as well as control methods. We present the rationale behind the language, discuss its application, and propose testable hypotheses for an empirical evaluation.

Introduction

Software-as-a-service (SaaS) encompasses all applications running in the cloud that are accessible by end-users as metered services (Lenk et al. 2009; Mell and Grance 2009). In this context, a service is any kind of deployed software, provided by service platforms on demand. A *service platform*, in turn, is software which makes available these deployed applications as a service to end-users while taking care of elasticity, metering, and billing (Lenk et al. 2009; Mell and Grance 2009; Rimal et al. 2009). Factors that are often associated with SaaS and that drive the rapid growth of service platforms include shortened deployment times, reduced upfront implementation, and minimized long-term overheads compared to traditional on-premise software (Holt et al. 2011).

Active participation of consumers and external suppliers in the value creation process additionally accelerates SaaS's success by leveraging network effects (Chesbrough 2012). Network effects are self-enforcing effects of value generation, generated by causal loops of reciprocal interdependency between platform attractiveness and third party value provisioning (Rohlf's 1974; Shapiro and Varian 1998). Network effects do not respond to simple linear relationships. They are characterized by complex differential interdependencies of the various participants and activities in and around the service platform (Sterman 2000). After opening the platform to third party service provisioning, the platform operator has to give up control over their service quality to some degree and he has to accept a certain level of self-organization by the service providers (Lee et al. 2010).

To be able to incite and exploit network effects, platform operators need to be able to manage the flows of service provisioning and consumption. They can do so by using their platform authority. Platform authority describes the exercisable degree of control and influence of a platform operator over an ecosystem participant or activity. The design of a service platform which allows this degree of control and influence is challenging due to two major factors that need to be considered: the ecosystem's capacity for self-organization and the platform's capability to exhibit network attractiveness. Self-organization is the capability to act autonomously without central control, the sensitivity to changes, and the adaptiveness to these changes (Ashby 1962; De Wolf and Holvoet 2005). In the context of service platforms, on the one hand, the platform operator has to attribute its ecosystem certain rights to act and react in a self-paced way, for example, by granting the right to develop or deploy services within the platform. The platform operator also needs to prepare the terrain for sensitivity and adaptiveness. On the other hand, network attractiveness is an exponential function of an addressed target group's sensitivity to a value in the platform multiplied with its cardinality, divided by a limiting threshold. Only a small set of platform operators, offering selective cross-industry-applications, have up to now successfully mastered the viable development of such service platform (Keswani et al. 2012).

In this paper, our goal is to describe an information system (IS) design theory as a "systematic specification of design knowledge" (Gregor and Jones 2007). An appropriate design theory will help us to devise a method to build service platform designs, which allows us to incite the desired effects and instantiations of parameters for a successful balance between control and self-organization. We contribute to the IS body of knowledge by providing such a design theory which can guide both software developers and academic researchers. We have followed the Design Science approach (Hevner et al. 2004), a socio-technologist/developmentalist research paradigm (Gregg et al. 2001). As a basis we derived meta-requirements to guide our research process (Walls et al. 1992). The knowledge-building takes place through conceptualizing, designing, and building of artifacts (i.e., as proof-of-concept, proof-by-demonstration). Our core artifact is the developed methodology and language that builds on the design theory and supports the IS development process of designing a service platform which enables network effects and leads to the successful adoption of SaaS offerings.

In the following, we present the rationale behind the derivation of the required constructs and methods to graphically model participants and activities on and around service platforms, which can be controlled or influenced through the platform operator's platform authority to incite networks effects. First, we provide an overview of foundations, related work, and methodology of our research. Then, we elaborate on the solution design of a design theory which results into the design method: a graphical modeling language to capture areas of platform authority, process constructs, and control methods. We apply the artifact, discuss the outcome, and give an outlook on further empirical evaluation.

Background and Foundations

Service Platforms and Network Effects

The term *network effect* describes the phenomenon that products or services only become valuable when large numbers of people use them (Rohlf's 1974; Shapiro and Varian 1998). Many digital goods are subject to network effects. For example, Sterman (2000) shows that network effects can be strengthened or weakened by complementary network effects. Rochet and Tirole (2003) reveal that in many cases where network effects occur, two or more distinct participant groups benefit from each other. They term the underlying business model *two or multisided markets or platforms*. Eisenmann et al. speak of demand-sided network effects, finding that platforms, which are open for external supply, can encompass cross-sided network effects (Eisenmann et al. 2008; Eisenmann et al. 2011; Eisenmann et al. 2006). In these effects, supply side and demand side are interdependent.

A key notion of network effects is the self-organization of the concerned participants. The term *self-organization* originates from system theory and non-equilibrium physics (Ashby 1962). The concept entails that small changes can incite amplified network effects in self-organizing systems. Self-organizing systems show a set of features which favor network effects (De Wolf and Holvoet 2005; Nicolis and Prigogine 1977):

- *autonomy*, i.e. the parts of the system act without central control;
- *adaptability*, i.e. parts of the system are able to cope with changes autonomously and rapidly and align to a new temporal equilibrium; and
- *sensitivity* to change, i.e. the system is sensitive to a changing requirements and can react rapidly.

Viable and successful service platforms that use designs leveraging network effects should benefit from autonomy, adaptability, and sensitivity to change. The set of self-organizing, autonomous participants represent the platform ecosystem. Participants and ecosystem are in a reciprocal relationship with each other.

In the context of service platforms, the provision of information about activities on a platform to participants who are active on a platform is called *feedback*. The participants' reaction to this information causes reciprocity, meaning modified activity on the platform leading to renewed feedback. Feedback loops describe the reciprocity that a stock has on its own filling or depletion, when impacting on the flow through an auxiliary variable. We refer to the feedback loops of auxiliary variables, which are functions of stocks and which impact flows, as *causal loops*.

Hence, in our context, network effects describe the reciprocal relation between the value of a service platform and the quantity of involved service consumers and service providers. Network effects are driven by self-organization of the platform ecosystem.

Consequently, successful service platforms benefit from a suitably high quantity of managed services supplied by self-organized service and from high quantities of subscribed users. Unsuccessful platforms tend to not accomplish self-enforcing network effects (Scholten et al. 2009). We refer to the value propositions that are expected to incite network effects as *base value*. Base values need to exceed a minimal threshold called *critical mass*.

Service Platforms and Control Theory

Katz and Shapiro (1986), Schilling (2009), Boudreau (2010), Parker and Alstyn's (2010), as well as Hagiu and Lee (2011) investigate control in the context of open technical platforms, enabling cooperation of distinct supplier and user groups. These researchers consider control from perspectives of power through technology ownership, decision of technical evolution, and distribution rights. Being in control includes the rights to appropriate value from a technology. Parker and Alstyn's (2010) explicitly study service platforms. They discuss implications of platform openness on platform controllability. All these studies include network effects in their reasoning and conceptualizing. In contrast to differential equations that are used in system-theoretical approaches and allow for the considerations of dynamic

causal loops, these works use mathematical formulations based on either static-comparative or game-theoretical perspectives. These formulas are equation-based, describing points of equilibrium.

Closer to the system-theoretical consideration of causal loops is the theory of feedback loop control (Ashby 1964; Conant and Ashby 1970). It describes the concept of a (technical) system being regulated by a control device aligning a reference value with the fed back system output (cf. Figure 1).

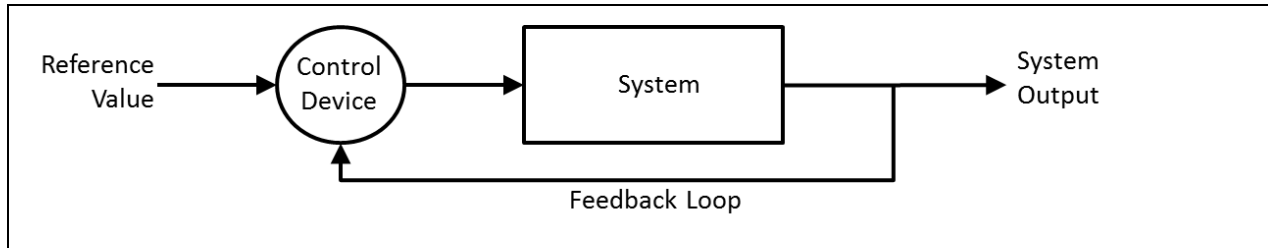


Figure 1. Feedback Controlled System

Consequently, in our context for service platforms, *control* describes service management actions of the platform operator, which change a set of parameters from a current status to a target status. Control in a platform context operates as a closed loop which uses feedback monitoring in the context of a regulatory process. The devices and methods which are used to control such a process are called *control modes*.

Table 1. Modes and Methods of Control based on Kirsch (1997)					
Control Mode	Key Characteristics	Antecedent Condition	Example Method	Group of Methods	Control Method
Behavior (formal)	Rules and procedures articulated	Behavior observability	Development methodology, work assignments, rules and procedures	Enforcement	Prescriptive control, sanctional control
	Rewards based on compliance to rules and procedures			Incentive	
Outcome (formal)	Outcomes and goals articulated	Outcome measurability	Comparison of outcome with the expected level of performance and successive rewards	Enforcement	Restrictive control
	Rewards based on producing outcome and goals			Incentive	Motivational control
Self (informal)	Individual sanctions himself	none	Individual empowerment, self-management and self-monitoring, and self-rewarding with respect to self-set goals	Enforcement	
	Individual defines task goals or procedures, Individual monitors and rewards himself, the rewards are based in parts on individuals self-control skills			Incentive	Informative control, market regulative control
Community (informal)	Identification and reinforcement of acceptable behaviors	none	Coalitions of individuals with share ideologies, socialization, hiring and training practices, implemented rituals and ceremonies	Enforcement	
	Common values and beliefs and problem solving philosophy			Incentive	Market regulative control, informative control, motivational control

Building on these insights, Kirsch (1997) develops a taxonomy of control modes. Table 1 summarizes the four control modes. The set of formal modes contains: (a) *behavior control*, characterized through articulated rules and procedures and (b) *outcome control*, defined by expressed project outcomes and goals. Formal control modes can be designed to be observable and are hence suitable in enforcement and reward-oriented approaches (Eisenhardt 1985; Kirsch 1997; Ouchi 1979). Informal modes include (c) *self-control* and (d) *clan control*. Instead of clans, we use the term *community*, as this term is established in the service platform context. Self-control relies fully on an individual's ability and competence to self-control. Community control modes are suitable where coalitions of individuals cluster around common values and beliefs (Kirsch 1997). Informal modes lack observability and, hence, their successful implementation is difficult to observe. However, an organization can benefit from such interpersonal, feedback-seeking dynamics in social structures that support self-regulating of social processes (Ashford and Tsui 1991).

Conceptual Modeling of Service Networks

There are several approaches to the design of service networks around service platforms. Eisenmann et al. (2006; 2007; 2010; 2011) for instance, have a specific service platform focus, considering aspects of control and network effects for business models and service innovation with little or no consideration of technical service management. However, none of these approaches do consider control.

In order to facilitate the development process of a service platform, any development group must agree on some shared representation forms. Based on this representation they can exchange and discuss ideas, thoughts, opinions, objectives, and beliefs about the object system (Hirschheim et al. 1995) (e.g., about the design and parameters of a service platform). One suitable means of representation is considered to be *conceptual modeling* (Frank 1999).

Conceptual models feature three main characteristics. They are mappings of representations, abstractions from originals, and in accordance with pragmatic requirements (Peckham and Maryanski 1988). Thus, multiple people's perceptions of a matter are typically condensed to a shared representation in the modeling process (Gupta and Sykes 2001). The conceptual models, which we consider, always result in an abstract account of reality expressed in a formal and described *graphical language* (Falkenberg et al. 1998). Such languages should be designed to meet a specific target group's way of thinking.

As-is conceptual modeling of networks has received some attention. These approaches extract models in a bottom-up approach through the access, retrieval, and combination of globally distributed information. Examples are the Open Semantic Service Relationship approach (OSSR) (Cardoso 2013) or the Open Semantic Service Networks approach (OSSN) (Cardoso et al. 2013; Cardoso et al. 2012). The graphical modeling of service networks in the context of service management, however, is not in focus. The Service Network Notation (SNN) focuses on the optimization of value in service networks (Bitsaki et al. 2008; Bitsaki et al. 2009). The Service Network Modeling Notation (SNMN) concentrates on the enablement of service offerings, requests and provision between and inside organizations (Danylevych et al. 2010). e3* (e3value/e3services/e3controls) focuses on qualitative analysis of value flows in networks (Kartseva et al. 2010). All three have in common that they model service networks as a set of nodes and edges in a *to-be* approach. These notations consider explicit relationships of value exchange. The more implicit approaches of ecosystem influence, as immanent in network effects, do not find consideration, nor do the control modes.

Research Methodology

We follow a Design Science procedure (Hevner et al. 2004) in order to develop our approach for designing service platforms. As our research progressed, we iteratively developed the configuration of the artifacts (Baskerville et al. 2009). The intended (and unintended) impact of the developed artifacts was evaluated to show their usefulness in solving the problem of designing a viable service platform. This involved comparing objectives and observed results in real-world settings (Hevner et al. 2004; Winter 2008). We make use of network effects (Rohlf's 1974; Shapiro and Varian 1998) and control modes (Kirsch 1997) as our kernel theories to derive a design theory with design requirements and a design rationale underlying

our approach (Simon 1996; Walls et al. 1992). The primary artifact of our research is a design method for designing service platforms in form of a conceptual modeling language.

We derived the following meta-requirements (Walls et al. 1992) to structure our research:

- (a) the design theory must be able to express all structural aspects of network effects such as base values as well as control modes within staged areas of platform authority,
- (b) the design theory must be able to express all procedural aspects of network effects, such as causal loops, as well as control modes on transactions and influences within staged areas of platform authority, and
- (c) the design theory must be able to express all enforcing and incentivizing aspects of network effects, in particular regarding control modes for behavior, outcome, community, and self-control within staged areas of platform authority.

Furthermore, we deem it necessary for the resulting design method to align with the following meta-requirements in order to be of relevance:

- (d) the method must be suitable to support the information system development process by providing an improved understanding of network effects and control modes, which in turn should result in better designs for service platforms, and
- (e) the derived design method must be a shared representation of ideas, thoughts, and opinions (Hirschheim et al. 1995), an abstraction from the original, retaining only the relevant parameters (Langer 1954) which improves the development process by either one or both, development speed and/ or cost.

The research process was an iterative process of building artifacts, intervention and learning, and enhancement, as described in Design Thinking (Meinel and Leifer 2010) and as suggested by Sein et al. (2011) in the Action Design Research approach. We compiled the first version of control methods through data elicitation, surveys, and field studies. They evolved over a period between 2009 and 2013 through a cyclical research process. The continuous application of the method lead to adaptations and enhancements and resulted in redesigns of the approach, a common occurrence in Design Science research (Davison et al. 2004; Vaishnavi and Kuechler 2008). This cyclic research process consisted, amongst others, of the following steps:

- Analysis of successful service platforms over a period of four years: We judged service platforms as “successful” according to two alternative criteria: First, if they were successfully in terms of financial success in the service platform domain, substantiated through prior published investigations (Holt et al. 2011; Keswani et al. 2012). Second, if they provided methods or structures that successfully support network effects (e.g., the platform Trello applies specific structures and methods to achieve network effects in the case of small numbers of users). In particular, we considered Appirio, BOINC, Dropbox, Google, Facebook, Intensify, Intuit, LongJump, Netsuite, Salesforce.com, SAP, S.Chand Edutech, and Trello.
- Comparative longitudinal study: We conducted a comparative longitudinal analysis of service intermediaries, analyzing and evaluating their methods, service quality, and service success (Scholten et al. 2009). Specifically, the study compared the intermediary operators SeekDa, WebServiceList, Xmethods, RemoteMethods, eSigma, and StrikeIron Market Place. This was extended by a longitudinal comparison of service quality for SeekDa and StrikeIron Market Place.
- Field studies on three selected service platforms: We analyzed in detail the platforms Force.com by Salesforce.com, SuiteApp by Netsuite, and Facebook Platform by Facebook. In the study design, we deployed an own sample service on the platforms to gain a deeper insight into the control and release methods as well as on the platforms’ configurations. We complemented the field studies’ findings by an analysis of the platforms’ terms and conditions.
- Field studies on the research platform and e-market pilot AGORA: Within the Theseus/ TEXO project, we analyzed the value and effect of explicit and implicit feedback methods to improve service quality (May et al. 2011). The monitoring of experimental test consumer behavior allowed retrieving necessary feedback on consumer self-organization that was incited by feedback.

Furthermore, we enhanced the results through an iterative process of discussion and reflection with researchers as well as with business analysts and platform architects from industry. Our research built on and benefited from continuous exchange with project partners from industry (e.g., within the Theseus/TEXO project). We conducted tests through modeling (intervention/action taking) of sample cases with the above described successful platforms to reveal whether all known control methods can be represented with our design method and language. We conducted an evaluation and reflection of the derived design theory with four platform operators (two at an intermediate as well as pre-final stage and with two at the end of their respective design processes).

In prior publications, we have presented an initial categorization on control methods (Scholten et al. 2010; Scholten et al. 2011). Critical discussion on an evolved solution (Scholten 2011) led to a correlation with the control modes suggested by Kirsch (1997). The categorization of control methods has also been discussed (Scholten et al. 2012).

Solution Design

In this section, we derive and formalize structural and procedural elements as well as groups of control methods of our design theory: First, structural elements embrace different areas of staged platform authority, areas with the capability to scale according to demand. Second, process elements allow for the modeling and analysis of causal loops and network effects. Finally, we introduce control methods to support the implementation of service management. Each of these subsections derives functional design requirements for a design theory, which are kept non-specific to any graphical notation or logical conceptual modeling language which – in turn – is the design method described in the following section.

Initial Modeling Constructs: Areas of Staged Platform Authority

Based on control theory and in specific the taxonomy of the control mode theory, we use the term *platform authority* to refer to the platform operator's ability to exert control (in any of the four modes) over the quality of offered services. Legner (2009) provides a rather functional, tripartite taxonomy consisting of infomediaries (a), e-hubs (b), and e-markets (c) to describe different levels of control and influence. Complementing this categorization with the concept of integrators (d) (Heinrich et al. 2011) results in three *areas of staged authority* for intermediaries to express the fact that the intermediary either has full, limited, or no authority over service consumer and/ or provider activities:

- a *control area* is the section where the platform operator can exert full platform authority through enforcement (the operator can observe and steer all events and structures within the control area);
- an *influence area* is the subsection of a platform ecosystem where the platform operator can only exert limited platform authority through incentives (the operator cannot observe but only inquire information on events within the influence area);
- a *noise area* is the subsection outside the platform ecosystem where the platform operator has no platform authority and no form of control.

The following examples with the different types of intermediaries illustrate this understanding (cf. Figure 2). The services, which are simply crawled by an infomediary (a), lie outside its control or influence area. E-hubs (b) do not have access to any data traffic, while federating a service. However, both consumer and service provider actively choose cooperation with the e-hub. Therefore both are located in the influence area. E-markets (c) represent a first supply concept with limited enforcing authority. E-markets can control and tab all traffic between the client and the service provider, as it is routed through the control area. The integrator (d) is omniscient to all traffic coming from and going to the consumer. He has enforcing power over the service as it is deployed within its control area. However, this omniscience and platform authority shrinks, once the service is of a composite nature with services outside the platform's control area. The consumer is always placed in the influence area.

In the *control area*, the platform operator has full platform authority. From a technical point of view, this means that it has the capability of enforcing its technical infrastructure on its physical servers or on its

virtual machines in an infrastructure-as-a-service environment and on all technically enabled activities which take place on the platform. Examples for the latter are service consumption and/ or service provisioning through third party providers. From an organizational point of view, this means that the platform operator can exert full platform authority over workforce (e.g., internal teams or external entities working on assignment). The allocation of control methods as points to exert service management is limited to the control area. Moreover, from the control area the platform operator exerts influence over the ecosystem participants. Hence, the control area is the area where the platform operator can exert control over activities and internal participants. It is also the area from where he influences ecosystem participants which are placed outside the control area. The following functional design requirements result from the above: First of all, the control area must be a structural element. All process constructs within this area must be equipped with control methods. Process constructs within the control area are allowed to be a source of influences, pointing at ecosystem participants in the influence area.

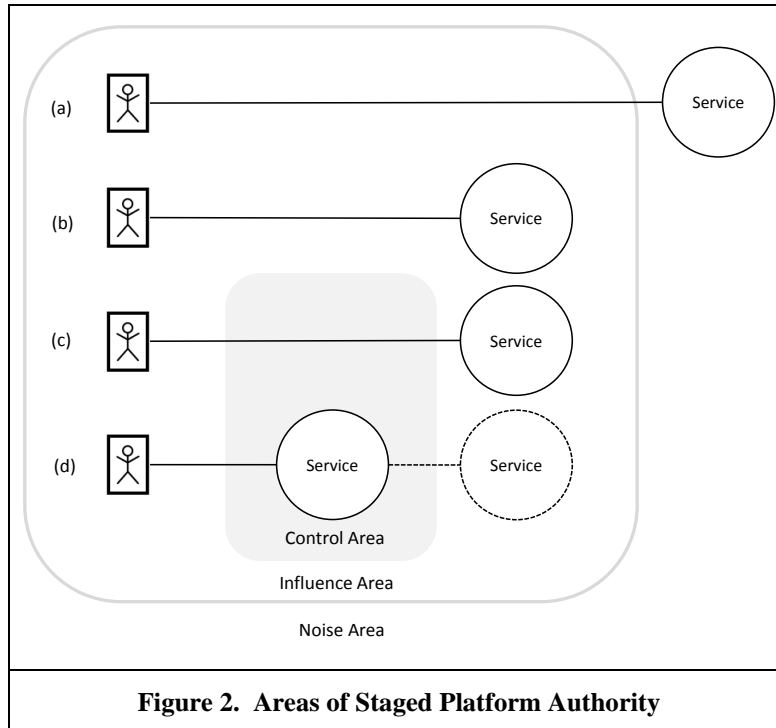


Figure 2. Areas of Staged Platform Authority

To better structure the control area, the construct of *division* can be used as a frame to group elements, which belong together (e.g., because they build a set of solutions or because they are in the same physical location). *Division groups* describe finite sets of similar technical environments, which could be servers at customer locations or native applications on client PCs or mobile phones, controlled by the platform operator. In areas where network effects apply, the platform operator has to be aware that the environment may require to scale.

The *influence area* is the structural area of the ecosystem around the control area. Ecosystem participants, who are in or may come into a value exchanging relationship with the service platform, are located in this area. Process constructs represent ecosystem participants as well as relationships. The influence area does not allow for control through enforcement, as it is outside the reach of the platform operator's platform authority. The influence area therefore requires indirectly operating methods of control, which the platform operator exerts from the control area, so called incentivizing methods. In the influence area, the ecosystem participants can also influence each other. Lastly, they are subject to influences of entities external to the ecosystem (e.g., competitors). The construct of influence area accommodates process constructs, representing ecosystem participants. Constructs are allowed to be a source of transactions into the control area. Furthermore, the platform is open for influence from ecosystem participants. Influence impacts through the ecosystem's activities on the platform and through the ecosystem-triggered market regulative control. Being outside the reach of the platform's full platform authority, the process constructs within the influence area must not carry control methods.

The *noise area* is the structural area around the ecosystem. In this area, the platform operator cannot exert any influence on the participants. It accommodates competitors and participants uninterested in becoming customers. Participants in the noise area are neither in a relationship of value exchange with the platform operator, nor can they exert influence on the platform operator. However, they may influence the ecosystem participants of the platform and may, thus, cause a backflow of value (e.g., cancellations). Hence, the noise area embraces all areas outside the platform ecosystem. Whereas the platform operator cannot exert any platform authority, the participants in this area can influence the ecosystem participants in the influence area. No value flow happens between noise area and control area as they have no direct relation with each other. The process constructs within the control area must not allow control methods in this area. Neither can any construct in this area be the target of influences from any other area, nor can any process constructs in this area be the source of influences, pointing into the influence area. Lastly, constructs in this area cannot be sources of transactions into the control area as they are not in a value exchanging relation with the platform.

Table 2 provides an overview of the solution design, elaborated in this section and substantiated in the subsequent subsections

Requirement	Solution Design	Attribute	Degree of Exercisable Platform Authority
Represent areas with full platform authority of the platform operator over all service provider and consumer activities	Control area	None	Control
Represent areas with limited platform authority	Influence area	None	Influence
Represent areas without platform authority	Noise area	None	None
Divider allowing to substructure the control area, able to depict technical environments which need to scale	Division	Scalable (Boolean)	Control
Divider group allowing to substructure the area of full platform authority	Division group	Scalable (Boolean)	Control

Process Constructs

From a process perspective, we require a representation of:

- *unspecific and specific participants* in and outside the platform and the platform ecosystem;
- *influences and transactions*; and
- *network effects*.

We formalize process constructs based on systems dynamics theory (De Wolf and Holvoet 2004; Nicolis and Prigogine 1977) and introduce *participant groups*, *participants*, *activities*, *influences*, and *transactions*. Causal loops, which incite network effects, consist of a concatenation of constructs and therefore do not require an element of their own.

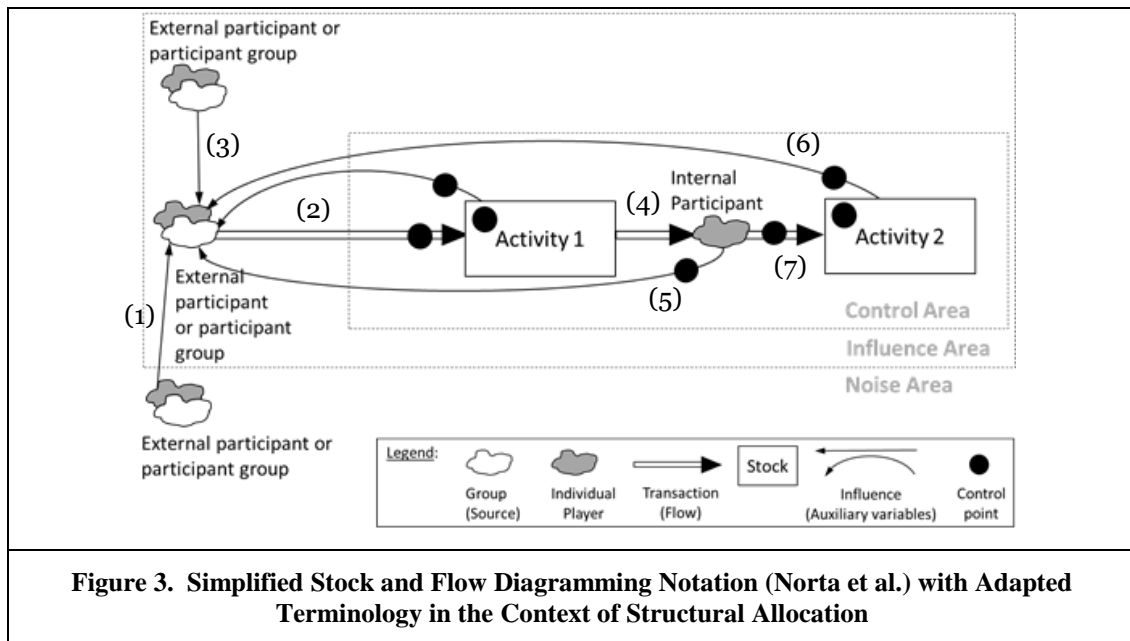
The construct *participant* describes specific entities with respect to the service platform in and outside its ecosystem. From a system dynamics point of view, participants are small stock, perceived to be static in capacity and accumulation. A participant may occur in any area. A group of specific entities of a large but finite size is called a *participant group*. Participant groups by definition only occur in the influence area and the noise area as potential new users or uninterested players. Participant groups do not move from one area to another as they represent a certain type of entities. A single participant would be a bottleneck within a causal loop because of its close to static stock behavior. Such loops rather require groups of participants, able to fill the accumulating activities within a platform. The construct of a participant can be equally applied for consumers and suppliers.

Figure 3 gives a schematic overview of the following discussion. Participants with no relation to the platform (i.e., participants in the noise area) cannot be sources of transactions. As the origin of an endogenous variable, these participants must be able to influence participants of participant groups in the

influence area (1). If a participant is an ecosystem participant, it may have a defined relationship with the platform and, thus, be the source of a transaction (2). It may also influence other ecosystem participants as a source of an endogenous variable (3). Participants cannot be the source of influence pointing at constructs within the control area.

Figure 3 also shows that the internal participant can be the origin of an auxiliary variable which stimulates a value flow into the platform. For example, an internal department could provide an existing (considered static) stock of services for deployment into activity 2. This provisioning may serve as base value. The influence pointing from the participant to the external participant (5) may stimulate a value flow to fill activity 1.

Activities, as depicted in Figure 3, correspond to stocks in system dynamics. They can accumulate and deplete. Activities represent the location for interaction of participants or participant groups in and with the platform. They are the target of participants or participant groups, addressed through transactions. Incoming transactions from participants and participant groups within the influence area (2) as well as from activities and participants within the control area (4) are allowed. Since they are IT-enabled tasks by the platform, activities are exclusive to the control area. Outgoing transactions to participants and other activities in the control area describe workflows. Any accumulation within an activity is considered an increase in value because value denotes positive effects on performance of actions, objects, and tasks. For example the quantity of services deployed might have value for service consumers (6). The more participants are active, the more the stock activity accumulates and vice versa. Activities can represent a base value to incite network effects. Such a base value within an activity is of explicit relevance due to its exponentially growing network attractiveness. Hence, activities can include a base value attribute.



Transactions are value flows into or within the control area. Their source is either in the influence area or in the control area. Participants and participant groups in the influence area are allowed to be the source of a transaction (2). In the control area, activities and participants can be the source of a transaction (4). Each transaction must have exactly one source and only one target. Constructs in the noise area cannot initiate a transaction. Constructs in the influence area and noise area cannot be target of a transaction. Value flows coming from the ecosystem can only target activities, as those are the only constructs inside the platform which exhibit stock characteristics.

Influence stimulates ecosystem participants or ecosystem participant groups to choose specific value flows into the service platform. From a system dynamics perspective, influences are auxiliary variables, which control the rate of flow of transactions through the stimulus of their sources. Influences must exclusively address participants or participant groups. Their targets are limited to the influence area (1, 3, 5, and 6). We consider influences as group of stimuli on ecosystem participants. In an uncontrolled way, it can be

any action without intervention of the platform operator and potentially without effect (e.g., exchange of information between ecosystem participants). When equipped with a control method, it means that the influence is amplified through the platform operator's incentives, i.e. market regulative control, informative control, and motivational control (cf. the following section).

Constructs for Control Methods

Control methods allow platform operators to intervene in managing services and service consumption by managed self-organization. They can be used to steer causal loops surrounding the platform (i.e., to generate network effects). Adding control methods to causal loops turns those loops into controlled feedback systems. Figure 3 visualizes the provision of control methods through so-called control points. Restricting full platform authority to the control area leads to the exclusive positioning of control methods within this area. Figure 3 shows three of the four possible positions:

- on activities, regulating their interaction (e.g., activity 1 and activity 2 in Figure 3);
- on transactions, regulating the inflow (resp. outflow) into an activity (e.g., transaction (2) and (7) in Figure 3);
- on influences, regulating the feedback into the ecosystem (e.g., influences (5) and (6) in Figure 3); and
- on internal participants, as they are in hierarchical subordination to the platform operator's authority (not shown in Figure 3)

We categorize the enforcing methods as *prescriptive control*, *sanctional control*, and *restrictive control*. Similarly, we categorize the incentivizing control methods as *market regulative control*, *informative control*, and *motivational control*.

Table 3 gives an overview of the applicable control methods within the areas of platform authority for all process constructs.

	Enforcing Methods			Incentivizing Methods		
	Prescriptive Control	Sanctional Control	Restrictive Control	Market Regulative Control	Informative Control	Motivational Control
Activity	x	x		x	x	x
Participant	x					
Transaction			x			
Influence				x	x	x

An antecedent condition for *prescriptive control* is the observability of behavior (Kirsch 1997). Platform operators can observe behavior of third party activities and participants in the control area since they have submitted to their prescriptions. Having allocated the activities and participants within the area of full platform authority, the platform operator can (a) observe and (b) steer the activities of external participants and modify their outcome. Hence, prescriptive control is the sequence of observing and steering a participant's set of actions within activities. Prescriptive control may further include subsequent corrective actions by the platform operator.

Activities in the control area of the service platform are subject to an additional method of control called *sanctional control*. The platform can either sanction deployed services or subscribed users. Sanctions are enforced in the moment of a policy breach (e.g., copyright infringement or SLA violations). The platform gathers information on policy breaches, for example, by automatic verification, service support, or complaint management systems. After the discovery of such an infringement, an escalation routine is initiated. The escalation routine can vary from defined time for correction or statement requested from the participant to immediate un-deployments, depending on aspects of safety, security, or the importance of the policy breach.

Restrictive control is a formal control method. It acts as a filter on transactions within the control area and verifies compliance with service platform policies (7). Restrictive control relies on outcome measurability as an antecedent condition.

Market regulative control categorizes control methods which are fully driven by the ecosystem and which are generated through explicit feedback. Market regulative control can address service consumers (e.g., through ranking) as well as service providers (e.g., through recommendations). Its objective is to communicate information on service quality. This is a self-organizing process; the ecosystem self-organizes when, in reciprocity, consumers adapt their consumption behavior and service providers amend service quality. This represents a causal loop within the control area as well as into the ecosystem. Figure 3 visualizes this causal loop by a black dot annotated in the upper left corner of activity 1. Another possibility of applying market regulatory control is on influences, pointing from internal participants to external participants or participant groups (5). For example, collaborative feedback systems may enable participants to recommend or advise against a value or base value contribution, provided by the participant. Market regulative control is located exclusively at influences and activities, within the control area.

Informative control stimulates creativity in the ecosystem, targeting individuals or communities and providing them with preprocessed information (e.g., on service requirements, preferences, or feedback on specific quality). It addresses the participants' intrinsic motivation (Frey and Oberholzer-Gee 1997). The respective informative control method consequently highlights opportunities or invitations to participate in activities. In contrast to the contributions in market regulative control, which are community-based, the platform operator manages informative control and addresses existing or potential participants or participant groups. The method's goal is to incite a self-organizing process of alignment in favor of the service platform. Like market regulative control, it is located on activities and influences that are directed from the control area into the platform ecosystem, for example, activity 2 (6). Informative control of activities explicitly addresses and supports existing participants. Informative control of influences addresses external participants (e.g., potential customers).

Motivational control comprises methods which explicitly set incentives and potential rewards for participants. The platform operator can incentivize activities in the control area. Moreover, motivational control can affect influences originating from the control area, pointing at participants or participant groups in the ecosystem (6). Motivational control for activities addresses existing participants (e.g., to financially motivate existing participants to produce services which are of strategic relevance to the platform in a certain segment). With respect to influences, motivational control addresses external participants (e.g., to motivate subscribed participants to invite new participants to win a reward).

Demonstration and Evaluation

In this section, we present the derivation of our design method, a graphical modeling language, from the design theory for service platforms elaborated on in the previous section. So far, we have described the rationale on how the meta-requirements from network effects theory and control mode theory have guided us in devising design requirements and constructs for a design theory to represent all relevant elements to the design of service platforms. In the following, we provide a description and an assessment of an instantiation of this design theory.

We implemented the notation's syntax and morphology through a stencil set, a plugged-in runtime constraint, and the layout processor within the Oryx framework (Decker et al. 2008). The editor includes a shape repository, accommodating the language's structural and procedural elements, a modeling canvas and a property configuration panel, allowing for configuring the control methods. Figure 4 presents a screenshot of the editor with a sample model. Given the prevalence of the Business Process Modeling Notation (BPMN) among targeted modelers, we chose to enforce the effectiveness of the language through allusion to BPMN conversations, thus improving users' familiarity with the symbolism (Object Management Group Inc. 2011). A full specification of the language can be found in Scholten (2013).

The sample model displays the control area (box with black line), influence area (box with dashed line) as well as the noise area (shaded canvas). There are two activities (hexagon) and one participant (box with rounded edges) in the control area and one participant and one participant group (three stacked boxes

with rounded edges) in the influence area. All value flows between the constructs are either labeled as an influence or as a transaction. All controls are marked as white or black dots on participants and/or value flows. Black dots signify active control mechanisms and white dots signify the possibility to enable control methods. The base value of the platform is displayed as the symbol β with the *Department for Own Value Contribution*. The base value could also be placed on the activity *deploy services*. Allocating the symbol to the participant emphasizes that the base value does not increase through loops.

This department now deploys services on the service platform, which serve as base value. This influences the *Target Group* to *subscribe*. The more subscribers the platform has and the more services the platform has, the more of the Target Group subscribe, that is, the stock is filled. This creates a causal loop, a demand-sided network effect. The amount of subscriptions eventually also has an influence on *Partner 1* to start deploying services on the service platform. This activity eventually influences more participants of the Target Group to subscribe and so forth. This second causal loop is a cross-sided network effect through service consumption and third party supply.

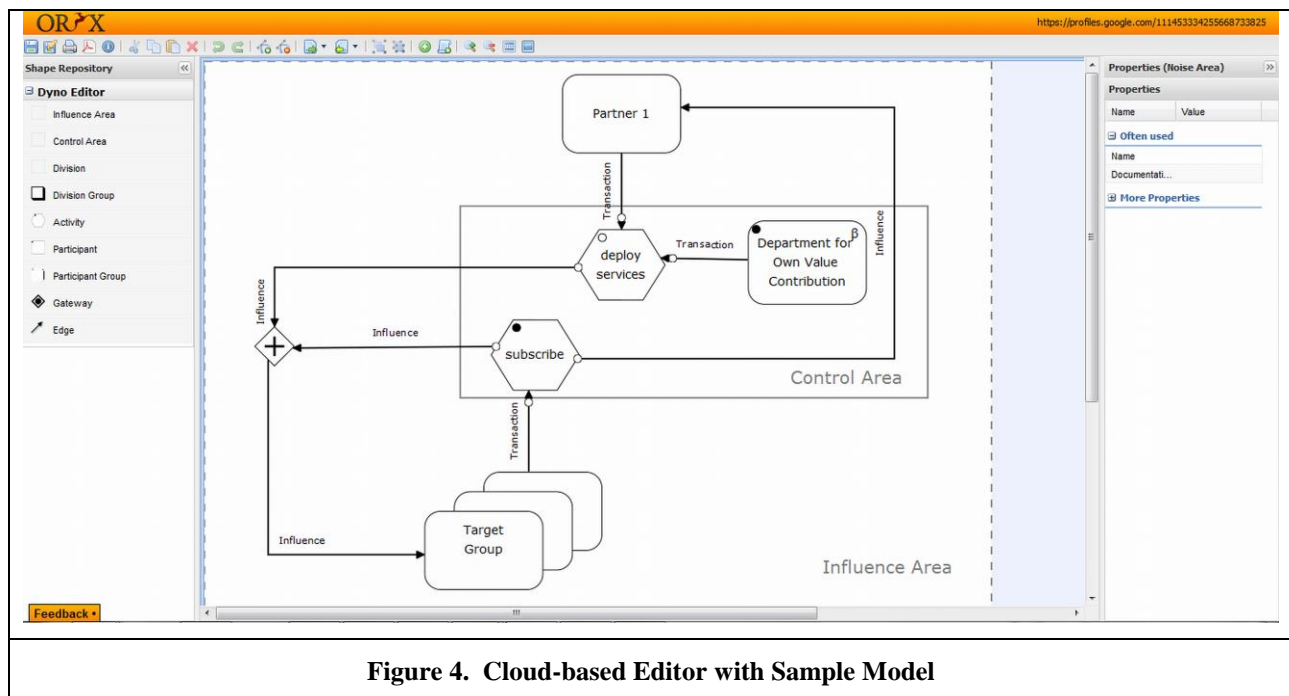


Figure 4. Cloud-based Editor with Sample Model

As an initial, first evaluation step, we modeled a set of existing service platforms which were part of our initial explorative analysis. Service platforms comprised Salesforce.com, Netsuite, Dropbox, and Google+. This helped exploring the language's expressiveness (i.e. its capability to express all relevant processes and control methods encountered in the real world in a semantically and syntactically correct way). However, conducting these tests solely on our own creates the risk to provide biased results (Zelkowitz and Wallace 1997). Therefore, we also conducted several initial field studies with platform operators and evaluated the outcomes using self-control surveys of participants. Users modeled and designed their service platforms and later evaluated their modeling experience and results. In a second step, we provided them with our graphical modeling language and software. They received training on the process constructs and control methods of the graphical modeling language as well as training on the concrete notation and the use of the Web-based editor. The users evaluated their modeling experience and their results against their previous experience. We are aware of the fact that this setup is not without bias. As we evaluated with business users having actual issues in platform design, we were subject to certain constraints and not able to devise proper test groups. We do elaborate on more controlled future empirical validations later in this section.

In the following, we exemplarily present the results achieved with M-Engineering, a company offering surveillance, control, and data acquisition solutions (SCADA) in automated processes. The company traditionally offered on-premise solutions and is a new entrant into the service platform business.

Question	Aggregate Score (n=10)
The design-time was shorter when using the software as compared to a design without the design method and software	2
I was quickly able to understand both design method and software	1,2
The design method gives me a better understanding of network effects and control possibilities in platform ecosystems	2
The design method and the software helped me to produce a better solution than without them	2
All relevant elements from the real-world scenario find application in the model-representation	2
The graphical user interface (GUI) was intuitive, and easy to follow	-0,6
Average of all 45 questions asked	1,60

As a result of this very preliminary evaluation, the modelers reaffirm our design decisions for our design theory based on our meta-requirements derived from network effects theory and control theory. Moreover, the design method as well as its notation was found to be suitable for the design of service platforms and not only shortened the design time but also improved the understanding of the effects at work. We have received less favorable results for the current usability of the implementation. We assume that this is mainly due to the fact that we limited ourselves to Web-based open source modeling frameworks, which are still in their infancy. Also, we have not yet focused on a wizard-like interface to guide the modelers but give them full access to all constructs of the design method.

So far, the design method and the software tool support service platform design in view of the exploitation of network effects, particularly referring to the control methods in the context of direct modes of action (transactions, activities) and more indirect modes (influences) and the staged areas of authority. With respect to a full-fledged empirical evaluation, we propose to conduct further controlled tests (e.g., laboratory experiment), as it is currently not feasible to judge and evaluate the design process and quality of actual, real-world service platforms designed with our design method. Neither is it feasible to evaluate the success of individual, real-world service platforms designed with our design method, as further factors are involved. Hence, we propose to derive testable hypotheses for controlled experiments from our meta-requirements, such as:

The use of systems that support both, the design of network effects and platform authority, will result (a) in an improved understanding of the possibilities to incite networks effects and exert control and, thus, (b) in better solutions than systems that only support the design of network effects.

Analogously:

The use of systems that support both, the design of network effects and platform authority, will result (a) in an improved understanding of the possibilities to incite networks effects and exert control and, thus, (b) in better solutions than system that only support the design of platform authority.

As a baseline, we deem it necessary to also test the following hypotheses as regards comparisons with approaches without support:

The use of systems that support both, the design of network effects and platform authority, will result (a) in an improved understanding of the possibilities to incite networks effects and exert control and, thus, (b) in better solutions than the use of “pen and paper” (i.e., no graphical modeling support).

One way to design a laboratory experiment for testing these hypotheses is to use a 1x4 factorial design with four groups of subjects, which will conduct the same task in one of the following four treatments: (a) pen and paper, (b) a system that supports modeling network effects, (c) a system that support platform authority, and (d) a system that supports both, network effects and platform authority. Using the same software (or at least notation) will contribute to the internal validity of the experiment results. The test subjects will be assigned at random to the four treatments. While we aim to select tasks that do not

require specific knowledge or training, it is difficult to do so within this confined context of service platform design - which as such is a task that requires expert knowledge. So all test subjects will need to receive the same training necessary to complete the task. We propose to have all subjects establish a baseline in a “pen and paper” test. Then, the subjects of the system treatments will receive software training and the next task will be handed out. While we can measure the duration to complete tasks within the experiment, we will need expert judges to compare and judge the resulting designs. In addition, the test subjects will be able to provide an evaluation of their perceived understanding factors affecting the platform ability to incite network effects.

Discussion and Conclusion

The emergence of service platforms and their openness to contributions of third parties as well as to self-organizing behavior on the consumer side give rise to new challenges for existing and future service providers. The increased autonomy of suppliers and consumers and the network effects resulting from their behavior increase the complexity of managing such platforms. Platform operators are well aware of the opportunities arising from leveraging network effects, particularly the possibility of exponential growth of service consumption and provisioning, which is certainly desirable. However, challenges also arise due to the loss of influence on service quality on the one hand and the possibility of unachieved growth or rapid collapse of the consumer base due to negative network effects on the other hand. Acting on these opportunities and challenges requires both theory and design support to enable the design of service platforms that are capable of harnessing desired network effects.

We proposed a graphical modeling language that is based on sound theoretical underpinnings and focuses on inciting network effects for service platforms. It uses, adapts, and structures existing knowledge originating from systems dynamics control theory, network theory, and dynamic market theory and applies it to service platform design. We understand network effects as instances of causal loops that are linked to both methods and structures on the platform and groups of potential participants. By including control methods into these loops, we allow for the design and creation of controlled feedback loops. Based on these insights, we have created a design method that allows us to position the different participants within and outside the platform’s control and influence according to the degree of authority that the platform operator can exert on them. Moreover, our approach allows us to describe a service platform’s various types of relationships and interactions, including causal loops. Lastly, we enable the placement of suitable control methods to ensure quality of service without suffocating self-organization.

In an iterative process, we have drafted, evaluated, and refined the language to its current state. We have described the rationale behind the constructs of the language (i.e., its staged areas of authority as well as its process support and control methods). We presented the application in a real-world case and discussed user feedback, which also supports our meta-requirements:

- (a) The initial evaluation of our design theory suggests that our proposed graphical modeling language is a suitable means to discuss and design all structural aspects of platform authority for service platforms inciting networks effects.
- (b) The initial evaluation of our design theory suggests that our proposed graphical modeling language is a suitable means to discuss and design all procedural aspects of platform authority for service platforms inciting networks effects.
- (c) The initial evaluation of our design theory suggests that our proposed graphical modeling language is a suitable means to discuss and design all enforcing and incentivizing control aspects of platform authority for service platforms inciting networks effects.

Furthermore, we presented the rationale behind our conceptualization of constructs for network effects and control modes:

- (d) We have presented a design method that incorporates all aspects which we deem relevant for the IS developer. In addition, we demonstrated that the resulting models improved the perceived understanding of the effects at work and the perceived quality of the service platform designs.

- (e) We have based the notation on the BPMN conversation diagram, which is an accepted design method for the abstract representation of collaboration. IS developers have noted the similarity and experienced a decrease in training and design time (a change in design cost was not measured).

Our research contributes to the existing body of knowledge on service platform design. For academics and researchers, we describe an IS design theory and design method for service platform design. The design method allows us to build service platform designs that foster the desired network effects and instantiations of parameters for a successful balance between control and self-organization. Our categorization of areas of staged platform authority (control area, influence area, and noise area) within our design theory for service platforms allows for pinpointing reasons for and causal loops of network effects in detail. Our design method allows for describing and visualizing with graphical models, on the one hand, direct exchange relationships with the service platform and, on the other hand, indirect relationships in terms of influences of the ecosystem by the platform provider. This allows illustrating and understanding these complex relationships using graphical models. As a result, our research provides a theoretical framework for the design of service platforms, taking into account their business models and leveraging the ecosystem of customers and suppliers. For practitioners from industry, we present a design method that is able to model and to identify network effects, allowing platform operators to manage the flows of service provisioning and consumption by fine-tuning their methods of platform authority. Moreover, methods for steering and controlling network effects can be visualized and designed, allowing steering and manipulating as well as potentially simulating these relationships. Consequently, the proposed design method can act as a tool for supporting the shared modeling, discussions, and decisions of solution managers and platform architects.

In addition, we have proposed a preliminary research design for further empirical evaluation. Apart from this lack of empirical evaluation in our methodology, we are aware of further limitations of the designed artifact. First, the graphical modeling language is targeted at system architects and decision makers; therefore it is rather abstract and it is not possible to (semi-) automatically derive any deployment architecture or even program code. So far, it is designed to be a means of communication to support early stages of system design. Nevertheless, our ultimate goal is a transformation of its logical models into program code. Following the model-driven architecture paradigm, modeling should not be an end in itself. Second, despite the language's rather high level of abstraction, our evaluations so far have shown that not all language constructs are self-explanatory. In addition, the current implementation does not offer any form of guidance for the creation of new models because we have not yet developed a procedure model. In some respect the same limitations also apply to the understanding of models by users. In contrast to typical process models, there is no single point of entry, and models (i.e., the represented service platforms and their ecosystems) have to be understood as a whole.

While we have presented initial evidence and results which suggest that the proposed graphical modeling language is a suitable tool to improve the understanding and, thus, design of network effects in service platforms, it is practically impossible to relate this to the economic and financial success of a service platform. Whether or not a service platform is successful in the market also depends on many additional factors which currently cannot be expressed with our graphical modeling language (e.g., pricing, branding, or the general market development). Moreover, our current transactions and influences represent a generic flow of value but do not attribute concrete valuations to allow for simulation and/or prediction.

Future research can focus on two areas: (1) extending the set of constructs and methods; (2) propagating the models to subsequent stages of IS development. All developed control methods could be further extended and refined through additional layers. Each new layer could offer additional sets of methods. Furthermore, the language in its present form focuses only on control methods. Attributes with other foci and aspects might be useful (e.g., monitoring or security). At this stage, our language provides a high-level view of interactions of service platforms and their ecosystems, with the focus on harnessing network effects. In order to be used in later stages of the systems development process, elements of the language also have to be passed on to subordinate layers of modeling languages other than BPMN conversation diagrams such as BPMN choreography diagrams and BPMN collaboration diagrams. These are executable as program code at the lowest level.

References

- Ashby, W.R. "Principles of the Self-organizing System," in *Principles of Self-Organization: Transactions of the University of Illinois Symposium*, H. von Foerster and G.W.J. Zopf (eds.), Pergamon Press, London, 1962, pp. 255-278.
- Ashby, W.R. *An Introduction to Cybernetics University*, Paperbacks, London, 1964.
- Ashford, S.J. and Tsui, A.S. "Self-regulation for Managerial Effectiveness: The Role of Active Feedback Seeking," *Academy of Management Journal* (34:2), 1991, pp. 251-280.
- Baskerville, R., Pries-Heje, J., and Venable, J. "Soft Design Science Methodology," in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST)*, Philadelphia, PA, 2009, pp. 1-11.
- Bitsaki, M., Danylevych, O., van den Heuvel, W., Koutras, G., Leymann, F., Mancioffi, M., Nikolaou, C., and Papazoglou, M. "An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network," in *Proceedings of the 1st European Conference ServiceWave. Lecture Notes in Computer Science Vol. 5377*, P. Mähönen, K. Pohl, and T. Priol (eds.), Madrid, 2008, pp. 196-207.
- Bitsaki, M., Danylevych, O., van den Heuvel, W.J.A.M., Koutras, G.D., Leymann, F., Mancioffi, M., Nikolaou, C.N., and Papazoglou, M.P. "Model Transformations to Leverage Service Networks," in *Proceedings of the 4th International Workshop on Engineering Service-Oriented Applications (WESOA). Lecture Notes in Computer Science Vol. 5472*, G. Feuerlicht and W. Lamersdorf (eds.), Sydney, 2009, pp. 103-117.
- Boudreau, K. "Open platform Strategies and Innovation: Granting Access vs. Devolving Control," *Management Science* (56:10), 2010, pp. 1849-1872.
- Cardoso, J. "Modeling Service Relationships for Service Networks," in *Proceedings of the 3rd International Conference on Exploring Services Sciences (IESS). Lecture Notes in Business Information Processing Volume Vol. 143*, J. Falcão e Cunha, M. Snene, and H. Nóvoa (eds.), Porto, 2013, pp. 114-128.
- Cardoso, J., Pedrinaci, C., and Leenheer, P. "Open Semantic Service Networks: Modeling and Analysis," in *Proceedings of the 3rd International Conference on Exploring Services Sciences (IESS). Lecture Notes in Business Information Processing Volume Vol. 143*, J. Falcão e Cunha, M. Snene, and H. Nóvoa (eds.), Porto, 2013, pp. 141-154.
- Cardoso, J., Pedrinaci, C., Leidig, T., Rupino, P., and De Leenheer, P. "Open Semantic Service Networks," in *Proceedings of the 4th International Symposium on Services Science (ISSS)*, K. Meyer and N. Abdelkafi (eds.), Leipzig, 2012.
- Chesbrough, H. "Open Innovation: Where Weve Been and Where Were Going," *Research-Technology Management* (55:4), 2012, pp. 20-27.
- Conant, R.C. and Ashby, W.R. "Every Good Regulator of a System Must Be a Model of That System," *International Journal of Systems Science* (1:2), 1970, pp. 89-97.
- Danylevych, O., Karastoyanova, D., and Leymann, F. "Service Networks Modelling: An SOA & BPM Standpoint," *Journal of Universal Computer Science* (16:13), 2010, pp. 1668-1693.
- Davison, R.M., Martinsons, M.G., and Kock, N. "Principles of Canonical Action Research," *Information Systems Journal* (14:1), 2004, pp. 65-86.
- De Wolf, T. and Holvoet, T. "Emergence Versus Self-organisation: Different Concepts but Promising When Combined," in *Proceedings of the 3rd International Workshop on Engineering Self-organising Systems (ESOA). Lecture Notes in Computer Science Vol. 3464*, S. Brueckner, G.D.M. Serugendo, A. Karageorgos, and R. Nagpal (eds.), New York, NY, 2004, pp. 1-15.
- De Wolf, T. and Holvoet, T. "Towards a Methodology for Engineering Self-organising Emergent Systems," in *Self-Organization and Autonomic Informatics (I)*, H. Czap, R. Unland, C. Branki, and H. Tianfield (eds.), IOS Press, Amsterdam, 2005, pp. 18-34.
- Decker, G., Overdick, H., and Weske, M. "Oryx: Sharing Conceptual Models on the Web," in *Proceedings of the 27th International Conference on Conceptual Modeling (2008). Lecture Notes in Computer Science Vol. 5231*, Q. Li, S. Spaccapietr, E. Yu, and A. Olivé (eds.), Barcelona, 2008, pp. 536-537.
- Eisenhardt, K.M. "Control: Organizational and Economic Approaches," *Management Science* (31:2), 1985, pp. 134-149.
- Eisenmann, T., Parker, G., and Van Alstyne, M. "Opening Platforms: How, When and Why?," in *Harvard Business School Working Paper*, Boston, MA, 2008.

- Eisenmann, T., Parker, G., and Van Alstyne, M. "Platform Envelopment," *Strategic Management Journal* (32:12), 2011, pp. 1270-1285.
- Eisenmann, T., Parker, G., and Van Alstyne, M.W. "Strategies for Two-sided Markets," *Harvard Business Review* (84:10), 2006, p. 92.
- Falkenberg, E.D., Hesse, W., Lindgreen, P., Nilsson, B.E., Oei, J.L.H., Rolland, C., Stamper, R.K., Assche, F.J.M.V., Verrijn-Stuart, A.A., and Voss, K. *A Framework of Information Systems Concepts. The FRISCO Report*, 1998, available at <http://www.mathematik.uni-marburg.de/~hesse/papers/fri-full.pdf>, accessed on 2013-08-19.
- Frank, U. "Conceptual Modelling as the Core of the Information Systems Discipline: Perspectives and Epistemological Challenges," in *Proceedings of the 5th Americas Conference on Information Systems (AMCIS)*, D.W. Haseman, D.L. Nazareth, and D.L. Goodhue (eds.), Milwaukee, WI, 1999, pp. 695-697.
- Frey, B.S. and Oberholzer-Gee, F. "The Cost of Price Incentives: An Empirical Analysis of Motivation Crowding-out," *The American Economic Review* (87:4), 1997, pp. 746-755.
- Gregg, D.G., Kulkarni, U.R., and Vinz , A.S. "Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems," *Information Systems Frontiers* (3:2), 2001, pp. 169-183.
- Gregor, S. and Jones, D. "The Anatomy of a Design Theory," *Journal of the AIS* (8:5), 2007, pp. 312-355.
- Gupta, D. and Sykes, J.A. "Conceptual Modelling Process and the Notion of a Concept," in *Information Modeling in the New Millenium*, M. Rossi and K. Siau (eds.), IDEA Group Hershey, PA, 2001, pp. 53-69.
- Hagiu, A. and Lee, R.S. "Exclusivity and Control," *Journal of Economics & Management Strategy* (20:3), 2011, pp. 679-708.
- Heinrich, B., Leist, S., and Zellner, G. "Service Integrators in Business Networks: The Importance of Relationship Values," *Electronic Markets* (21:4), 2011, pp. 215-235.
- Henderson, R., Podd, J., Smith, M., and Varela-Alvarez, H. "An Examination of Four User-based Software Evaluation Methods," *Interacting with Computers* (7:4), 1995, pp. 412-432.
- Hevner, A.R., March, S.T., Park, J., and Ram, S. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), 2004, pp. 75-105.
- Hirschheim, R., Klein, H.K., and Lyytinen, K. *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*, Cambridge University Press, Cambridge, 1995.
- Holt, A., Weiss, K., Huberty, K., Gelblum, E., Flanner, S., Devgan, S., Malik, A., Rozof, N., Wood, A., Standaert, P., Meunier, F., Lu, J., Chen, G., Lu, B., Han, K., Khare, V., and Miyachi, M. *Cloud Computing Takes Off: Market Set to Boom as Migration Accelerates*, 2011, available at http://www.morganstanley.com/views/perspectives/cloud_computing.pdf, accessed on 2013-08-19.
- Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. "Control Patterns in a Health-care Network," *European Journal of Information Systems* (19:3), 2010, pp. 320-343.
- Katz, M.L. and Shapiro, C. "Technology Adoption in the Presence of Network Externalities," *The Journal of Political Economy* (94:4), 1986, pp. 822-841.
- Keswani, S., Krans, A., Henlin, E.H., Mirandi, J., Casey, M., and Gagnon, K. *4Q11 Public Cloud Market Landscape. Cloud Business Quarterly*, 2012, available at <http://www.tbri.com/samples/download/index.cfm?sample=RS&docid=7691>, accessed on 2013-08-19.
- Kirsch, L.S. "Portfolios of Control Modes and IS Project Management," *Information Systems Research* (8:3), 1997, pp. 215-239.
- Langer, S.K. "Feeling and Form: A Theory of Art Developed from Philosophy in a New Key," *The Musical Quarterly* (40:1), 1954, pp. 102-106.
- Lee, S.M., Kim, T., Noh, Y., and Lee, B. "Success Factors of Platform Leadership in Web 2.0 Service Business," *Service Business* (4:2), 2010, pp. 89-103.
- Legner, C. "Do Web Services Foster Specialization? An Analysis of Commercial Web Service Directories," in *Proceedings of the 10. Internationale Tagung Wirtschaftsinformatik (WI)*, Vol. 1, H.R. Hansen, D. Karagiannis, and H.-G. Fill (eds.), Wien, 2009, pp. 67-76.
- Lenk, A., Klems, M., Nimis, J., Tai, S., and Sandholm, T. "What's Inside the Cloud? An Architectural Map of the Cloud Landscape," in *2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, IEEE, Vancouver, 2009, pp. 23-31.
- May, N., Scholten, U., and Fischer, R. "Towards an Automated Gap Analysis for E-service Portfolios," in *Proceedings of the 8th International Conference on Services Computing (SCC)*, Washington, DC, 2011, pp. 274-281.

- Meinel, C. and Leifer, L. (Eds.) *Design Thinking: Understand - Improve - Apply*, Springer, Heidelberg, 2010.
- Mell, P. and Grance, T. *The NIST Definition of Cloud Computing*, 2009, available at <http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc>, accessed on 2013-02-08.
- Nicolis, G. and Prigogine, I. *Self-organization in Nonequilibrium Systems: From Dissipative Structures to Order Through Fluctuations*, John Wiley and Sons, New York, NY, 1977.
- Norta, A., Hendrix, M., and Grefen, P. "A Pattern-Knowledge Base Supported Establishment of Inter-organizational Business Processes " in *Proceedings of the International OTM Workshop on Modeling Inter-Organizational Systems focusing on Collaboration and Interoperability, Architectures and Ontologies (MIOS-CIAO). Lecture Notes in Computer Science Vol. 4277*, R. Meersman, Z. Tari, and P. Herrero (eds.), Montpellier, 2006, pp. 834-843.
- Object Management Group Inc. *Business Process Model and Notation (BPMN) Version 2.0*, 2011, available at <http://www.omg.org/spec/BPMN/1.2/PDF>, accessed on 2011-12-21.
- Ouchi, W.G. "A Conceptual Framework for the Design of Organizational Control Mechanisms," *Management Science* (25:9), 1979, pp. 833-848.
- Parker, A. and Pohlmann, T. *The Emerging IT Ecosystem: The Line between Technology and Service Will Blur at a Faster Pace*, Forrester Research, Cambridge, MA, 2007.
- Parker, G. and Alstyne, M.V. "Innovation, Openness and Platform Control," in *11th ACM Conference on Electronic Commerce (EC)*, ACM, Cambridge, MA, 2010, pp. 95-96.
- Peckham, J. and Maryanski, F. "Semantic Data Models," *ACM Computing Surveys* (20:3), 1988, pp. 153-189.
- Rimal, B.P., Choi, E., and Lumb, I. "A Taxonomy and Survey of Cloud Computing Systems," in *Proceedings of the 5th International Joint Conference on INC, IMS, and IDC*, Seoul, 2009, pp. 44-51.
- Rochet, J.C. and Tirole, J. "Platform Competition in Two-sided Markets," *Journal of the European Economic Association* (1:4), 2003, pp. 990-1029.
- Rohlf, J. "A Theory of Interdependent Demand for a Communications Service," *The Bell Journal of Economics and Management Science* (5:1), 1974, pp. 16-37.
- Schilling, M.A. "Protecting or Diffusing a Technology Platform: Tradeoffs in Appropriability, Network Externalities, and Architectural Control," in *Platforms, Markets and Innovation*, A. Gawer (ed.), Edward Elgar, Cheltenham, 2009, pp. 192-218.
- Scholten, S., Scholten, U., and Fischer, R. "Composite Solutions for Consumer-driven Supply Chains: How to Control the Service-enabling Ecosystem?," in *Supply Management Research*, R. Bogaschewsky, M. Eßig, R. Lasch, and W. Stölzle (eds.), Gabler, Wiesbaden, 2010, pp. 277-301.
- Scholten, U. "Service Level Management in Platform Ecosystems," in *Proceedings of the Informatik 2010, Vol. 1*, Leipzig, 2011.
- Scholten, U. *Dynamic Network Notation: A Graphical Modeling Language to Support the Visualization and Management of Network Effects in Service Platforms*, Dissertation, Karlsruhe Institute of Technology, Karlsruhe, 2013.
- Scholten, U., Fischer, R., Bojkov, D., and May, N. "Supply Chain Control building on Emergent Self-Organizing Effects," in *Supply Management Research*, R. Bogaschewsky, M. Eßig, R. Lasch, and W. Stölzle (eds.), Gabler, Wiesbaden, 2011, pp. 311-335.
- Scholten, U., Fischer, R., and Zirpins, C. "Perspectives for Web Service Intermediaries: How Influence on Quality Makes the Difference," in *Proceedings of the 10th International Conference on Electronic Commerce and Web Technologies (EC-Web). Lecture Notes in Computer Science Vol. 5692*, T. Di Noia and F. Buccafurri (eds.), Linz, 2009, pp. 145-156.
- Scholten, U., Schuster, N., and Tai, S. "A Pattern Language and Repository for Service Network Management," in *Proceedings of the 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, Taipei, 2012, pp. 1-9.
- Sein, M., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. "Action Design Research," *MIS Quarterly* (35:1), 2011, pp. 37-56.
- Shapiro, C. and Varian, H.R. "Versioning: The Smart Way to Sell Information," *Harvard Business Review* (76:6), 1998, pp. 106-114.
- Simon, H.A. *The Sciences of the Artificial*, (3rd edn.), MIT Press, Cambridge, MA, 1996.
- Sterman, J.D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*, McGraw Hill Higher Education, Boston, MA, 2000.
- Vaishnavi, V.K. and Kuechler, W. *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*, Auerbach, Boca Raton, FL, 2008.

- Walls, J.G., Widmeyer, G.R., and El Sawy, O.A. "Building an information system design theory for vigilant EIS," *Information Systems Research* (3:1), 1992, pp. 36-59.
- Winter, R. "Design Science Research in Europe," *European Journal of Information Systems* (17:5), 2008, pp. 470-475.
- Zelkowitz, M.V. and Wallace, D. "Experimental Validation in Software Engineering," *Information and Software Technology* (39:11), 1997, pp. 735-743.