

Information-based Control of Service-enabling Ecosystems

Robin Fischer¹, Ulrich Scholten¹, Simone Scholten², Stefan Tai¹

¹Universität Karlsruhe (TH), Karlsruhe Services Research Institute (KSRI),
Englerstraße 11, 76131 Karlsruhe, Germany
{Ulrich.Scholten, Robin.Fischer, Stefan.Tai}@ksri.uni-karlsruhe.de

²SAP Research,
Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
Simone.Scholten@sap.com

Abstract. Continuous optimization of value co-creation in networks of consumers and autonomous service enablers describes a major challenge to mediating platform operators. In analogy to systems theory, we propose to introduce customizable feedback loops from the service-enabling ecosystem to the service enablers via the platform operator. Relevant feedback information can be derived from analysis of network structure, service interactions, and service consumer preferences. In using our method, optimization of individual service offerings and of the network as a whole is facilitated through the platform operator, while retaining the autonomy of each service provider in the network.

Keywords: Service-enabling Ecosystem, Service Value Net, Consumer Preference Cluster, Feedback Loop

1 Introduction

Software platform operators aim at flexibly providing a high variety of product and service configurations to satisfy individual consumer demand. Within this context, a platform is conceived as “a building block, which can be a product, a technology, or a service that acts as a foundation upon which other firms can develop complementary products, technologies or services” [1]. Software platforms build upon modular software architectures such as service-oriented architecture (SOA), Web application programming interfaces, and corresponding design rules that allow independent third parties to autonomously innovate and develop complementary software applications or electronic services on top of the platform. Examples are Facebook, StrikeIron, Salesforce or the SAP-Ecohub.

In consequence, business ecosystems co-evolve with the core platform offerings. These ecosystems have become a major source of competitive advantage. Their market success depends on consumer satisfaction with the “whole product” [2], delivered by the platform operator and created conjointly with third party service enablers, complementing the platform’s offering.

As a result, this holds the platform operator responsible for a continuous, efficient platform evolution and for ensuring that services provided by the ecosystem's service enablers, as an essential complement, evolve simultaneously with the platform. This constellation leads to a problem of information asymmetry between the platform operator and independent service enablers [12]: While the platform provider has access to market information allowing for a market-conform adaptation of the platform service portfolio, the platform provider does not hold the stakeholding power to implement change [3]. Independent services enablers can implement change, but lack access to market information. Thus, platform providers are not able to optimize their service portfolio in function of the actual consumer behavior.

The paper at hand responds to this void by introducing a method that supports the platform operator to effectively empower service enablers to optimize their services and the platform's service portfolio in a self-organized manner. Our approach is based upon system theory and follows principles as applied in control engineering. In the next section, we will provide general thoughts on the importance for co-evolution in business ecosystems. The illustrative example provided in section 3 will thereafter motivate our further analysis of the relation between business ecosystems, service-enabling ecosystems, intermediating platforms, and consumer preference clusters. The highlighted shortcomings in terms of ecosystem controllability and information asymmetry will lead to the introduction and description of our feedback-based approach in section 4. Before we summarize our work and point to future research in section 6, we will relate our work to state-of-the-art research in section 5.

2 Co-evolution in Business Ecosystems

Successful platform operators depend on a robust, highly productive business ecosystem of complementary third party companies to co-create the platform's overall value proposition and to support its market adoption [4]. Therein, "the performance of a firm is a function not only of its own capabilities or of its static position with respect to its competitors, customer, partners, and suppliers, but of its dynamic interactions with the ecosystem as a whole" [5]. Business ecosystems are understood as an "economic community supported by a foundation of interacting organizations and individuals [...]. This economic community produces goods and services of value to customers, who are themselves members of the ecosystem" [6]. The participants of the business ecosystem – embracing service enabler, lead producers, competitors, consumers and other stakeholders – "work co-operatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations" [6]. They co-evolve their capabilities and roles over time, and tend to align themselves with the directions set by one or more central companies, the platform operator. These dependencies, however, evoke particular indirect network effects, determining the platform's market success and profitability: The more external companies join the value net in order to create complementary innovations, the more valuable the platform becomes. This dynamic causes more users to adopt the platform and more complementors to enter the ecosystem [7].

Once having adopted a platform strategy, its long-term success depends on continuous innovation and renewal [8], embracing continuous services portfolio optimization and provision of superior customer value. Therefore, the platform operator is obliged to control the business ecosystem, characterized as complex self-organizing web of direct and indirect relationships between independent actors to co-create and deliver value, whereas the value of the total offering is determined and driven by the consumer. However, given the non-linear and autonomous behavior of independent services enablers, the platform operator cannot simply demand innovation or optimization of services by relying on top-down efforts to implement change. On the contrary, he has to encourage service enablers to keep on optimizing the complementary offerings and, therefore, the complete offer. This implies empowering and stimulating service enablers to invest in optimizing their service offerings by providing them with key market information. However, the participants still face the problem of information asymmetry [9]: Neoclassical theory postulates total disposability of market information to the vendor allowing for market-conform adaptation of his service portfolio [10; 11]. Being positioned in a dyadic relation with the platform operator (or the next tier service enabler) constitutes a significant limitation of accessible information (information asymmetry). The consequence is services out of phase with the actual market demand.

In summary, the platform operator's challenge with respect to its ecosystem relates to its lack of active control mechanisms as the complementary services are in the service enabler's ownership. The resulting question is how can they be influenced? Potential dominator-based control strategies are not subject of this paper and are hence neglected. Rather, our approach builds on the value creating emergent characteristics of networks and shared value. Thus, the platform operator's influence should leverage the common source of revenue: consumer satisfaction. They therefore need to enable the ecosystem to better respond on the consumers' preference patterns. We suggest influencing the ecosystem by providing key market information to service enablers. In the following, we will discuss the prominent role of platform operators in collecting relevant information on service consumption. Based on concepts of control theory, we will further describe how key market information can be fed back in a customized way.

3 Illustrative Example

At the outset, it is useful to shed light on the state-of-the art. In [3] we analyzed a range of platform operators for Web services based on longitudinal data. As an illustrative example, in the following we will discuss StrikeIron (www.strikeiron.com), which showed the best performance in the selection group. The company works with mechanisms similar to other platforms like Salesforce (www.force.com) in the Web application segment or the Apple App Store (www.apple.com) for iPhone applications.

Since 2008, StrikeIron operates a two-sided Web service delivery platform named IronCloud. IronCloud facilitates data distribution for Web service enablers and Web service consumption for commercial users. The service-enabling ecosystem is consti-

tuted through companies or independent developers, who provide software or online services and who look for a platform to broadly market their services. Consumers of StrikeIron's Web services are enterprises, who look for cost-effective standardized solutions for managing data distribution, including metering, monitoring and authentication.

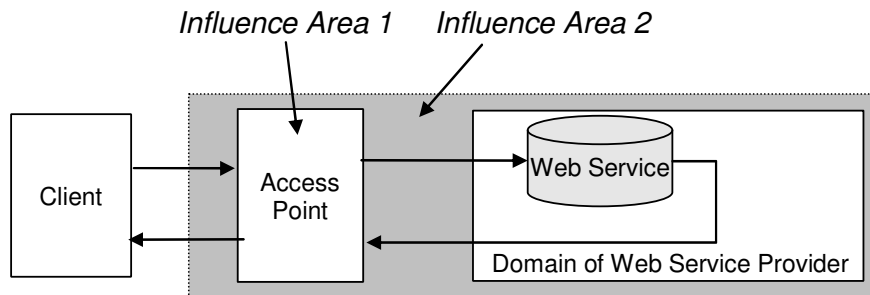


Fig. 1: Influence Area on Quality of Web Services

In contrast to classical e-Markets that simply syndicate services (influence area 1 of fig. 1), StrikeIron operates all services on its own domain, having influence on the transaction flow (influence area 1) as well as on the complete runtime environment (influence area 2). This allows for a proactive optimization of runtime related quality factors before initiating feedback-based optimization [3] of Web services.

To allow and to ensure interoperability, StrikeIron provides predefined Web application programming interfaces. Also, several directives for purposes such as authentication are imposed on the service-enabling ecosystem to achieve a basic level of manageability and security. In a verification routine before Web service publishing, StrikeIron automatically verifies whether the basic directives are followed. If not, feedback is given to the respective service enabler, who has to mend and to rerun the routine.

StrikeIron's consumption-based feedback is mainly focused on service level management quality (availability, accessibility, successability, response time). According to the company's own information (www.strikeiron.com), multiple consumption agents from remote geographic locations constantly ping, invoke and inspect the Web services and their responses on a minute-by-minute-basis and thus create a real-time picture of the respective quality parameters of their Web services. This information is compiled and reported to the service enablers in real-time to motivate performance optimization. Furthermore, an escalation routine is in place to alarm service engineers at StrikeIron or the respective service enablers to rapidly resolve problems at an early stage. If necessary, StrikeIron excludes underperforming services from its platform.

Feedback like publicly displayed usage information is missing. Also, StrikeIron does not provide consumer-based mechanisms for reputation measurement or a disclosure of a service's relevance. This complementary feedback information is applied e.g. by Salesforce to provide orientation to the consumers and to motivate its service enablers to retouch a specific service in case of consumer-perceived underperformance.

With the focus on service level management quality, StrikeIron limits itself to the optimization of purely transactional parameters. Feedback-based methods to measure the services' business value or extra-functional qualities related to security (e.g. trace management) and business process management are missing. Also, no analysis of consumption behavior is provided.

In conclusion, we believe that an enhancement of StrikeIron's existing feedback mechanisms would allow its service-enabling ecosystem to not only improve basic Web service quality, but to also lead to a better alignment of the overall product-mix in function of the consumer requirements. Lacking more enhanced feedback methods, the StrikeIron platform currently shows a bias towards elementary services like address verification or currency conversion, which are of limited value.

4 Closing the Loop

In the following, we introduce a method to develop more sophisticated feedback for advanced guidance of service enablers. We will also outline steps for its technical realization.

As depicted in fig. 2, business ecosystems embrace the economic community including service enablers, the platform and the consumers, while service-enabling ecosystems take account of the service enablers only. *Service Value Nets (SVN)*, in turn, are instances of business ecosystems, consisting of service enablers, the intermediary, the consumers as well as their respective relations within one period of composite service generation and consumption. The intermediary of the SVNs in our case is a software platform that mediates between service consumers and service enablers.

Consider a setting that consists of a software platform, which mediates between a first tier of services and a set of service consumers: A first tier service (S1.1, S2.1, S3.1) may be a basic service (S2.1), or a complex service (S1.1, S3.1) that aggregates other services (e.g., S1.2). We consider service-enabling ecosystems to be open pools of services. Thus, service-enabling ecosystems comprise both services that are frequently consumed in various types of SVNs (e.g., S2.3) and services that actually never participate in any type of business transaction (e.g., S2.2). We furthermore distinguish different types of SVNs by means of classifying mediated service consumers into consumer preference clusters (CPCs) that subsume service consumers according to their preference patterns (see section 4.2) and structural network information (see section 4.3). CPCs allow us to locate service subsets of the ecosystem that are involved in the value generation for distinct groups of similar consumption – in other words, services that target the same CPC. For the scenario shown in fig. 2, we identify three types of SVNs: SVN1 provides value for CPC1, which is created by S1.1, S1.2, S1.3, and S2.3 and mediated through the platform. Similarly, SVN2 provides value for CPC3 by use of S3.1, S3.2, S3.3, S2.3 and the platform. S2.1 is directly mediated over the platform and consumed by customers of CPC2, which makes up SVN3.

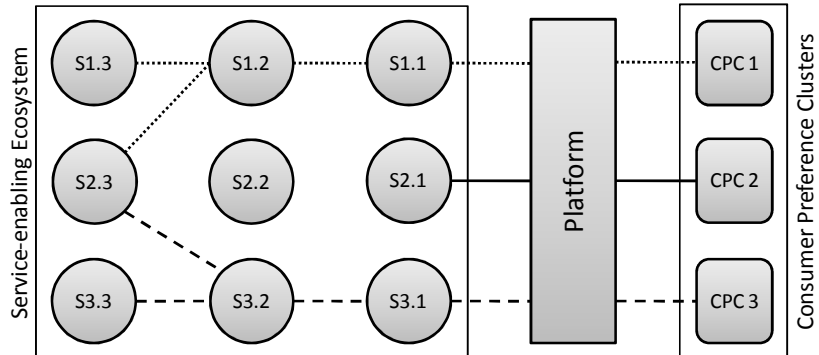


Fig. 2: Business Ecosystem

4.1 Feedback in Business Ecosystems

Our feedback mechanism follows ideas as applied in control engineering [13]. These kinds of feedback control systems aim at leveling the control path by monitoring the actual value tapped at the end of the control path (see fig. 3). An active regulator compares the actual value to a given set value (in fact, negated actual value and set value are summarized) resulting in a modified control value. Based on the newly adjusted control value the new actual value is tapped again, which is where the feedback loop is closed. Since disturbances may influence the control path by random noise or a steady change in the external environment, the actual value may be disturbed. In these cases, the feedback to the regulator allows for re-leveling actual and set value.

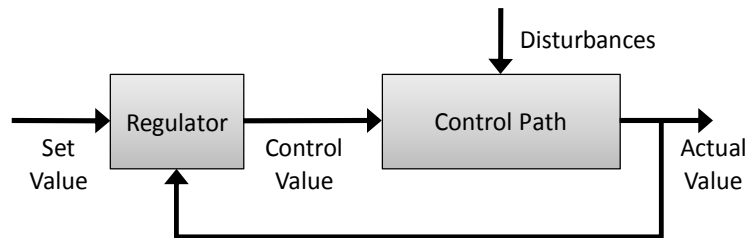


Fig. 3: Feedback Control System in System Theory

Given these introductory considerations on feedback control systems, we are now able to formulate a feedback control system for business ecosystems that is likewise made up of the interconnected entities of a control path and a regulator (see fig. 4). SVNs represent the control path of the feedback control system. Similar to control engineering, we monitor the output (actual value) of individual SVNs (e.g., value, performance, dynamic consumer behavior) as raw data to generate feedback for each service enabler.

Based on the received feedback on actual performance and in alignment with commercial goals (set value), each service enabler now may readjust his service prop-

osition. Concordant with control engineering, the set screw for service parameter modification is depicted in a detached imaginary component, called regulator. Hence, the network self-organizes. Service enablers may modify their value proposition by internal “process transformation” [14] or through a selective replacement of their respective service enablers. This then causes a “system transformation” [14].

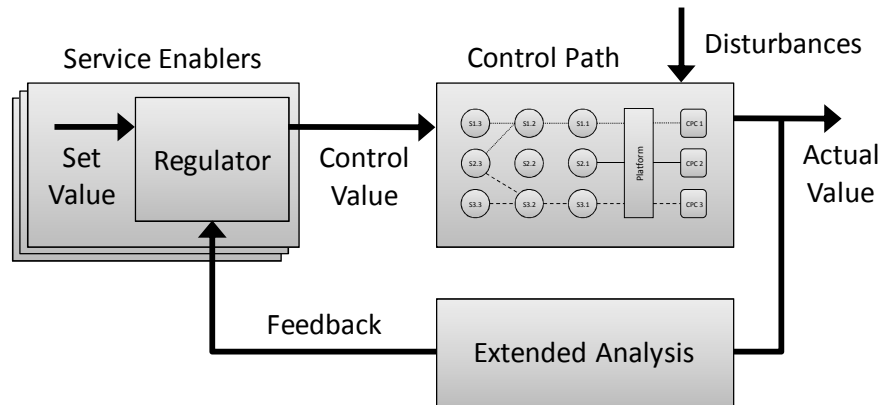


Fig. 4: Feedback Control System for Business Ecosystems

The system is characterized by a small cause, large effect principle through non-linear interactivity: The interaction of the service enablers, platform operator and consumers causes a so-called macroscopic effect – meaning a coherent behavior of the SVN. Feedback incites enforced reverse adaptation on the micro-level (the service enablers’ offerings). The results of a first optimization (e.g., reaction on shifting consumer preferences) again will trigger feedback. Over time, the service-enabling ecosystem will line up to a temporary equilibrium: Once there is no deviation anymore between the service enabler’s commercial goals (set value) and the actual value, the configuration stops growing until the next adaptation is initiated [15, 16].

In general, we consider two types of aggregated feedback: *preference feedback* and *structural feedback*. In our approach to provide business level feedback we furthermore propose *interpreted feedback*, which combines information of both types of aggregated feedback. In the following, we will outline our understanding of feedback information to illustrate the concept of feedback systems for service-enabling ecosystems.

4.2 Requirements for Preference Feedback

In order to analyze the consumers’ service preferences based on their actual consumption (consumption feedback), we propose to identify typical consumption patterns within the extended analysis component. Such a component could make use of the OASIS quality model WSQM (see fig. 5) to assess and aggregate individual consumer preferences. On the topmost level, the WSQM is grouped into *Business Value Quality*, *Service Level Measurement Quality*, *Business Process Quality*, *Suitability for*

Standards, Security Quality, and Manageability Quality. Based on [3], we further divide those categories into 21 subgroups, leading to a service’s specific quality bundle $Q = \{q_1, \dots, q_{21}\}$ on the provider side and to weighted preference bundles $P = \{\omega_1 q_1, \dots, \omega_{21} q_{21}\}$ on the consumers’ side. Thereby, the weighting factor ω_i describes the relative importance given to each quality parameter of a specific service by a particular group of service consumers.

Aggregated feedback on service quality and respective consumption preferences (preference feedback) can be read directly or indirectly derived from observable data. In the following, we provide requirements on the abilities to observe respective quality parameters.

Business Value Quality		Service Level Measurement Quality			Business Process Quality		Suitability for Standards		Security Quality		Manageability Quality									
Service Cost	Service Sustainability	Service Aftereffect	Service Recognition	Service Reputation	Response Time	Maximum Throughput	Availability	Accessibility	Successability	Reliable Messaging	Transaction Processing	Collaborability	Conformability	Interoperability	Authorization Management	Trace Management	Distributed Authorization	Management Information Offerability	Observability	Controllability

Fig. 5: Quality Parameters

4.2.1 Directly Observable Quality Parameters

Out of all quality parameters, *Service Level Management Quality* related parameters are potentially the easiest to observe. *Availability*, for instance, could be tracked by pinging mechanisms as performed in the example of StrikeIron. Due to this simplicity, it is the most frequently applied feedback approach and can be found in most of the service intermediaries and platform operators today. *Response Time*, *Maximum Throughput*, *Accessibility* and *Successability* on the other hand require observing service interactions. Solutions ranging from network-level monitoring to automated agents exist and may be applied here. Functionalities to observe, log and analyze basic service interactions are fundamental to an extended analysis.

Among the directly observable *Business Value Quality* parameters, we consider *Service Cost* to be critical. We thus require functionality to identify cost information, for instance from SLAs or service descriptions. *Service Recognition* can either be received from external sources like the Google Relevance or similar indicators. *Service Reputation* may also be surveyed through consumer-based ratings, as implemented by platform operators like Ebay or Amazon. *Service Reputation* can be either score-based (quantitative) or descriptive (qualitative). Thus, features such as tracking, inspecting and external sourcing are also required in the extended analysis.

The parameters of *Business Process Quality*, *Suitability for Standards*, *Security Quality* and *Manageability Quality* can only be monitored if the services are of native structure, that is, services are programmed and designed to conform to the platform operator's specifications [3]. If so, the services can be inspected through automated scanning routines. Figure 6 summarizes and correlates directly observable parameters with functionalities for their respective observation.

Quality	Quality Parameter	Monitoring methods					
		ping	invoke	inspect	track	ext. source	native only
Business Value Quality	Service Cost			v			
	Service Recognition					v	
	Service Reputation				v		
Service Level Management Quality	Response Time		v				
	Maximum Throughput		v		(v)		
	Availability	v					
	Accessibility		v				
	Successability		v				
Business Process Quality	Reliable Messaging			v			v
	Transaction Processing			v			v
	Collaborability			v			v
Suitability for Standards	Conformability			v			v
	Interoperability			v			v
Security Quality	Authorization Management			v			v
	Trace Management			v			v
	Distributed Authorization			v			v
Manageability Quality	Manag. Information Offerability			v			v
	Observability			v			v
	Controllability			v			v

Fig. 6: Directly Observable Quality Parameters

4.2.2 Indirectly Observable Quality Parameters

According to our study, two parameters cannot be observed directly (see fig. 7). However, as these parameters are of high importance, an indirect approach needs to be derived. In the following, we identify the steps allowing for their observation.

The *Service Aftereffect* describes the direct impact of a service on the consumer's system. The term "system" here stands for the service's application context (e.g., the targeted software system or organizational unit). However, we do not know the consumer's system nor do we know the individual return-on-investment (ROI) of a particular service. But we can measure a service's magnitude to create aftereffect indirectly by analyzing whether a particular consumer sources his demanded service from the service enablers. If yes, we can assume that he was satisfied. If he sources a service of similar functionality in later purchases from a new supplier, we can suppose that those new supplier provides higher utility. According to sampling theory, similar

tendencies created from a statistically relevant amount of homogeneous users will indicate that the decision is not an arbitrary individual case, but satisfies the requirement of *representativity* [22, 23].

In a similar approach, long-term consumer behavior can be considered: Over what period of time do consumers invoke specific services? By statistically answering this question, we can conclude a service's sustainability. Utilization patterns of bigger user groups will then allow for generalization.

We can conclude that for indirectly observing *Service Aftereffect* and *Service Sustainability*, information on individual service interactions is required to be collected by means of extended analysis. However, the focus is less on timeliness, frequency or erroneous cases, but on the identification of value co-creating service instances, i.e. collaborating service enablers.

Quality	Quality Parameter
Business Value	Service Aftereffect
Quality	Service Sustainability

Fig. 7: Indirectly Observable Quality Parameters

4.3 Requirements for Structural Feedback

Structural analysis will create the bridge to a more enhanced analysis, which correlates preference and economic data with the respective SVNs. To do so, we need to collect information on those. In the following, we outline an approach to observe structural aspects of SVNs.

As indicated above, the quality parameters *Service Aftereffect* and *Service Sustainability* can be only observed indirectly by correlating recurring SVNs with positive impact on *Service Aftereffect* and *Service Sustainability*. The prerequisite for indirectly observing these quality factors, though, is the ability to longitudinally observe the whole structure of SVNs over time. We separate structural changes into *process transitions* and *system transitions* [14].

Monitoring process transitions within an SVN requires tracking and comparison of service offerings provided by service enablers over time. Extended analysis providing structural feedback therefore requires the functionality to track versions of service offerings as well as the capability to identify evolution between two versions. Identifying system transitions on the other hand requires the tracking of individual service interactions that are to be performed for value creation within a particular type of an SVN. Feedback on systems transitions is required to allow for analyzing and defining SVNs according to particular sets of required service interactions. Required information includes data on constituting service enablers and their respective interaction patterns. In order to relate type level information (definition) to instance level data (execution) both designing and monitoring features are required.

In the pursuit of our goal to provide vital management information to the service enablers, we can now correlate preference and structural feedback.

4.4 Interpreted Feedback

The fact that one service consumer sources its desired service from service enabler A during a particular period of time, but sources from service enabler B afterwards is an important observation. Potentially, there are plenty of reasons for the change such as a change in the consumer's preference pattern, market exit of a service enabler or optimization of individual service offerings. Identification of these kinds of roots, though, requires additional information that needs to be interpreted by correlating aggregated feedback, i.e. correlation of preference feedback and structural feedback.

As described in section 2, the overall goal of this work is to design feedback loops that empower the service-enabling ecosystem to better respond to effects like evolving consumer preference patterns or market changes. The generation of CPCs for business ecosystems based on structural and preference analysis provides information on what service is chosen for which preference pattern and which one is omitted. After correlating preference clusters with their respective financial relevance on the platform (e.g. cumulated turnover during a specific time period within a particular preference cluster), the platform operator is in a position to guide each service enabler towards a specific optimization of quality parameters. The service enablers should be inclined to heed this advice, if they want to further attain yet unexploited, but financially interesting consumption clusters. In an example scenario, an enabler of data storage services may be provided with the information that he could access a profitable cluster of financial transactions, if only his services had sufficient security quality.

5 Related Work

The ideas presented in this paper build upon a variety of existing research. In the following, we position our approach to related work from system theory and managing quality of service.

The application of control theory in the field of software systems is not a new approach. Already 20 years ago, principles of control theory have been applied to the area of computer networks [17]. They mainly focus on system identification for finding optimal convergence and fairness policy for congestion avoidance. Thus, they aim at deriving a transfer function that allows the modeling of a system's behavior. Other applications of systems theory can be found in the area of scheduling in operating systems [18] or distributed environments such as distributed multimedia systems [19].

These approaches, however, rely on very detailed models of linear closed-loop systems and require time-consuming analyses. They are building upon a set of non-trivial assumptions that can only be identified through expert knowledge. These assumptions do not hold for the field of dynamic ecosystems and their respective value nets. Furthermore, the complexity of SVNs [12] makes a modeling of transfer functions difficult, if not impossible. As we seek to derive the evolving states of the system from our feedback information, the availability of ex-ante knowledge on the structure of SVNs is ruled out. Additionally, as the service enablers and consumers in such a context are self-organized players in a system of permeable borders, we also need to consider it as non-linear. Transfer functions of linear nature, as applied [17] are only

valid during periods of temporary equilibrium. However, transitions from one equilibrium to the subsequent cannot be captured at all. In addition, we aim at monitoring and influencing business ecosystems in at least near to real-time. To do that, we cannot rely on expert interviews in order to identify the right set of assumptions.

In [20] an approach is proposed that appears to be more reasonable for our application domain. The authors identify systems, based on the aggregation of observable statistical data. In that, they treat the system under control as black boxes to reduce influences of system complexity and lack of expert knowledge. We go similar ways by incorporating a regulator into an existing target system (in our case: business ecosystems). However, their approach on regulator-design is based on the assumption of system linearity. This is contradicting our perception of dynamically changing SVNs. Furthermore, our approach differs with respect to providing individual feedback for each service enabler while in [20] only a particular software system is considered.

In the field of service quality, there exists a variety of work that seeks for providing a quality definition for software systems and services. However, there is only little work that focuses on managing the quality of services in business ecosystems or SVNs. [21] propose an approach to “increase a software vendor’s flexibility, and responsiveness to changes in performance and usage of its service-based, online software, at specific customers and concerning its software in general” [21]. In their “Service Knowledge Utilization (SKU)” approach, they propose metrics for both individual services and service methods. These basic metrics are then aggregated to indices that provide an overview of a customer’s perceived service performance, usability and client utilization. The authors furthermore propose and implement a prototype that provides functionality to track metrics by weaving trace functionality into existing services. They claim that their systems may be introduced into arbitrary service environments by following an aspect-oriented approach. Although we share the same basic idea of using service usage information – or more broadly: feedback – and providing this information to individual service enablers, we see the following shortcomings in comparison to our approach. First, SKU is based on the assumption of a dyadic relationship between service provider and individual customers. Our concept of business ecosystems, however, comprises platform providers that mediate between service enablers and consumers. Therefore, generalizing SKU to our domain of SVNs is constrained. Furthermore, we consider an SVN to be provided through service-enabling ecosystems, having at least one tier of service enablers. Therefore, we anticipate further difficulties and open questions by recursively applying SKU for each transaction that occurs between service enablers of different tiers.

6 Summary and Outlook

Modern service systems are highly dynamic business ecosystems that typically comprise multiple consumers, autonomous service enablers, and a mediating platform operator. The continuous optimization of value co-creation in this setting requires novel mechanisms for all three roles. To this end, we propose to group consumers into CPCs, to model the service-enabling ecosystem in terms of the SVNs it supports, and to introduce customizable feedback loops from the service-enabling ecosystem back

to the service enablers via the platform operator. We distinguish and classify different kinds of feedback information – preference feedback, structural feedback, and interpreted feedback – and outline how feedback loops can be implemented by the platform. Our method capacitates platform operators to share critical market and consumer information with service enablers, and thus to improve quality and value of both individual service offerings and the network as a whole.

This paper describes novel ideas and fundamental concepts in support of a larger effort to study strategic (service) value networks. In future work, we plan to exemplify and tailor our method for specific application domains and case studies, to (continue to) develop and implement platform feedback mechanisms, and to provide experimental evaluation results.

7 References

- [1] Gawer, A.: Platform Dynamics and Strategies: From Products to Services. In: Gawer, A. (ed.): *Platforms, Markets and Innovation*, Cheltenham (2009).
- [2] Moore, G. (1999) *Crossing the Chasm*, Harper, NY.
- [3] Scholten, U.; Fischer, R.; Zirpins, C.: Perspectives for Web Service Intermediaries: How Influence on Quality Makes the Difference. *Proceedings of the 10th International Conference on Electronic Commerce and Web Technologies (EC-Web'09)*, 2009, pp. 145-156.
- [4] Gawer, A.; Cusumano, M. A. (2002) *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, Harvard Business School Press, Boston.
- [5] Iansiti, M.; Levien, R. (2002) *The New Operational Dynamics of Business Ecosystems: Implications for Policy, Operations and Technology Strategy*.
- [6] Moore, J. F.: *Predators and Prey: A New Ecology of Competition*, Harvard Business Review, (1993), 71, pp. 75-86.
- [7] Cusumano, M.: *The Changing Software Business: Moving from Products to Services*, IEEE Computer, (2008), 41 (1), 78-85.
- [8] Meyer, M. H.; Lehnerd, A. P. (1997) *The Power of Product Platforms: Building Value and Cost Leadership*, Free Press, NY.
- [9] Williamson, O. E.: *The modern corporation*. *Journal of Economic Literature*, (1981), 19 (4), pp. 1537-1568.
- [10] Liening, A. (1998) *Komplexe Systeme zwischen Ordnung und Chaos*, LIT, Münster.
- [11] Kleine, A. (1995) *Entscheidungstheoretischer Aspekte der Principle-Agent Theorie*, Physica, Heidelberg.
- [12] Basole, R. C.; Rouse, W. B.: *Complexity of Service Value Networks: Conceptualization and Empirical Investigation*, IBM System Journal, (2008), 47 (1), pp. 53-70.
- [13] Franklin, G. F.; Powell, D. J.; Abbas, E.-N. (2009) *Feedback Control of Dynamic Systems*, Paerson Prentice Hall, Upper Saddle River.
- [14] Bernet, B.: *Technologie an der Schwelle des 21. Jh.* In: Belz, C.; Bieger, T. (eds.): *Dienstleistungskompetenz und innovative Geschäftsmodelle*; Thexis, St. Gallen (2000).
- [15] De Wolf, T.; Holvoet, T.: *Adaptive Behavior based on Evolving Thresholds with Feedback*. *Proceedings of the Conference on Artificial Intelligence and Simulation of Behaviour (AISB'03)*, 2003, pp. 91-96.
- [16] Nicolis, G.; Prigogine, I. (1977) *Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order through Fluctuations*, Wiley, NY.
- [17] Chiu, D.-M.; Jain, R.: *Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks*. *Computer Networks and ISDN systems*, (1989), 17 (1), pp.1-14.

- [18] Steere, D. C.; Goel, A.; Gruenberg, J.; McNamee, D.; Pu, C.; Walpole, J.: A Feedback-driven Proportion Allocator for Real-rate Scheduling. Proceedings of the third Symposium on Operating Systems Design and Implementation (OSDI'99), 1999, pp. 145-158.
- [19] Li, B.; Nahrstedt, K.: Control-based Middleware Framework for Quality of Service Applications. IEEE Journal on Selected Areas in Communication, (1990), 17 (9), pp. 1632-1650.
- [20] Parekh, S.; Gandhi, N.; Hellerstein, H.; Tilbury, D.; Jayram, T.; Bigus, J.: Using Control Theory to Achieve Service Level Objectives in Performance Management. Real-Time Systems, (2002), 23 (1), pp. 127-141.
- [21] van der Schuur, H.; Jansen, S.; Brinkkemper, S.: Becoming Responsive to Service Usage and Performance Changes by Applying Service Feedback Metrics to Software Maintenance. Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering – Workshops, 2008, pp. 53-62.
- [22] Kaya, M.; Himme, A.: Möglichkeiten der Stichprobenbildung. In: Albers, S.; Klapper, D.; Konradt, U.; Walter, A.; Wolf, J. (eds.): Methodik der Empirischen Forschung (2007), pp. 79-88.
- [23] von der Lippe, P.; Kladroba, A.: Repräsentativität von Stichproben, Marketing: Zeitschrift für Forschung und Praxis, (2002), 24, pp. 227-238.