The Dynamic Network Notation: Harnessing Network Effects in PaaS-Ecosystems

Ulrich Scholten
Karlsruhe Institute of Technology
Karlsruhe, Germany
Ulrich.Scholten@KIT.edu

Robin Fischer
Karlsruhe Institute of Technology
Karlsruhe, Germany
Robin.Fischer@KIT.edu

Christian Zirpins
Karlsruhe Institute of Technology
Karlsruhe, Germany
Christian.Zirpins@KIT.edu

ABSTRACT

Web applications complement the Platform-as-a-Service (PaaS) value by satisfying widespread and rapidly changing consumer requirements within limited time and budget. Successful PaaS providers excel in governing their market performance by leveraging complex network effects, which implicitly control PaaS-ecosystems. There is currently no methodically sound and easy to use tool available to business analysts and software engineers of PaaS-offerings that addresses challenges and opportunities in launching and governing such highly dynamic networks. In this paper, we capture network behavior through elements of complex system and control theory. Our dynamic network notation (DYNO) builds upon these theories. In more detail, DYNO models PaaS offerings with a focus on identifying and shaping network effects towards a sufficient user-base and an optimized portfolio of Web applications, all while maintaining a high quality of service.

Categories and Subject Descriptors

H.1.1 [**Models and Principles**]: Systems and Information Theory – *general systems theory.*

General Terms

Network Science, Network Effect, Dynamics, Design,

Keywords

PaaS, Web application, software ecosystem, network effects, causal loops, base value, control, notation.

1. INTRODUCTION

Successful platform operators have leveraged external development resources and creativity to extend their service portfolio by opening up to third party providers. New entrants, however, face difficulties in starting off an interconnected network of application providers and consumers ('PaaS-ecosystem') [17, 20]. Those networks are highly responsive and dynamic, while network players are fully self-organized. It is challenging to initiate and govern high quality Web application offerings in such a context. Future platform providers need to find the right network design, where the own value contribution can leverage network dynamics [19].

State-of-the-art in the domain of network science and software design falls short in providing tools that support business analysts and service engineers in modeling PaaS ecosystems and their immanent network dynamics. Tools used in the field of network dynamics (e.g. VenSim^{TM}, cp. fig. 1) lack a procedural dimension and remain rather explanatory. Harnessing dynamic networks around PaaS implies the need to incorporate system or complex network effects. In the dynamic context of Cloud Computing, those effects are originating in rather indirect (i.e., implicit) patterns and relationships. These cannot be directly modeled through service choreographies or process orchestration and require a tailored approach. We therefore introduce a notation for dynamic network effects, empowering business analysts and service engineers to govern network effects in PaaS-ecosystems.

We ground our research on dynamic network optimization theory [4, 5], system theory [14], control theory [3] and dynamic markets [7, 15, 21] in pursuit of creating a bridge between network science and application oriented modeling. We gained understanding on dynamic processes and base value through explorative analysis of several successful platform providers in [16, 17]. Second, we compiled the technical requirements through market analysis in conjunction with laboratory experiments [2, 8]. Third, we captured the relevance of protagonists' control on quality of service and the respective designs of distributed control settings have been gathered through a longitudinal analysis of service intermediaries [20].

DYNO aims to address the following design challenges of PaaS-ecosystems: *DC1:* Where are dynamic processes around the protagonist's value proposition located? *DC2:* What are the ignition factors (*base value*) for these dynamic processes? *DC3:* Where are service-providing IT-systems exposed to network effects and, thus, required to scale quickly to maintain high quality of service? *DC4:* How can quality of service be controlled in the PaaS ecosystem?

The remainder of the paper is structured as follows: We start with an analysis of network effects in PaaS in section 2. Building on that, we shed light on the design challenges formulated above, leading to a set of governance tasks (section 3). These tasks set the frame for a subsequent derivation of a meta-model and a notation (section 4). In section 5, we experimentally evaluate DYNO, followed by brief summary of related work and close with a conclusion and outlook.

2. NETWORK EFFECTS IN PAAS

Platforms-as-a-Service with their consumers on the one side and Web applications on the other side can be described as two-sided markets. Two-sided markets are subject to various network effects (same side, cross side and two-sided network effects) [7, 15, 21]. Those network effects are - from a system theoretic perspective - feedback loops, where the magnitude of a stock amplifies the

SIMPLEX '12, April 17 2012, Lyon, France

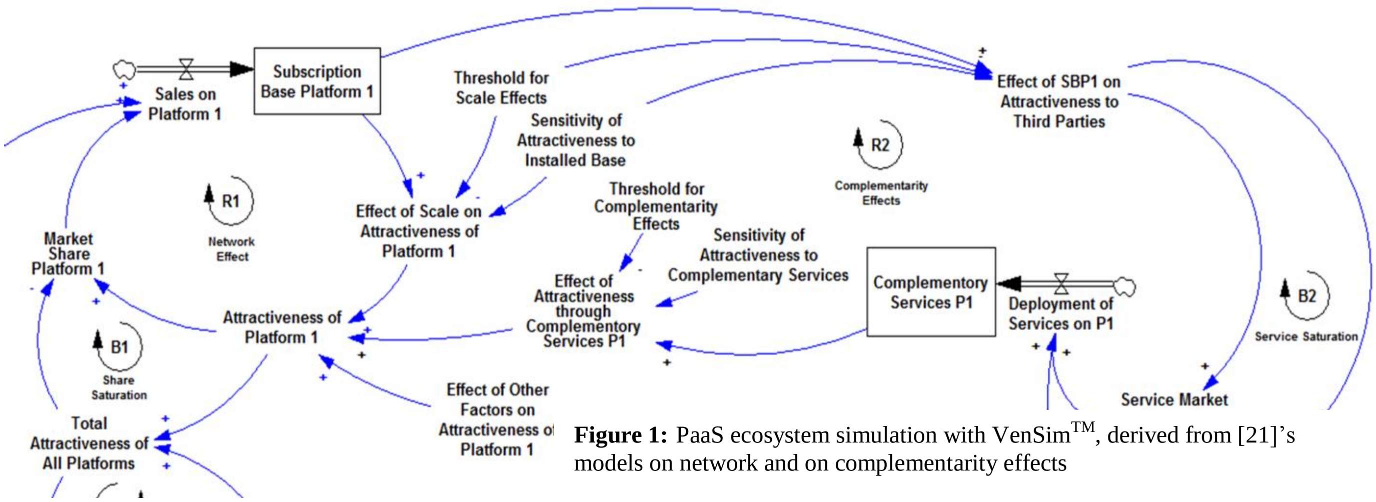

Figure 1: PaaS ecosystem simulation with VenSim™, derived from [21]'s models on network and on complementarity effects

flow, which in reciprocity increases the stock again. Imagine the inscription base of a platform as the stock. The higher it is, the more it attracts potential customers to subscribe. This subscription rate would be the flow. And it also is plausible that this resulting increased subscription base (stock) further amplifies the flow (i.e. additional subscriptions of new platform users).

In complex system theory, this effect is described as the exponential function

$$B = \exp(gt) * B_0 , \quad (1)$$

where B describes the subscription base, B_0 denominates the subscription base at the time t_0, g the fractional growth and t the time. In reality, the exponential behavior is limited to:

- s-shaped behavior defined in the logistic curve due to saturation effects and
- non-linear behavior due to interactions with other players, e.g. competitors.

In pursuit of designing a tool for practitioners and in search of applicable results, we chose causal loop diagrams as a first iteration (see fig. 1) to model complexity and non-linear behavior of platform ecosystems. A challenge is that the holistic data on competitors and consumers, required to qualitatively model such a complex environment is highly volatile and hardly attainable. For us, a simplifying factor is that we only want to identify and trigger network effects. We therefore need to qualitatively identify the causal loops within a PaaS environment, responsible for exponential growth behavior, or those which make market success unrealistic in the first place. Fig. 1 shows a fraction of a causal loop diagram around a PaaS. It builds on models for network and complementarity effects [21], which we integrated and applied on the PaaS ecosystem. It shows a same-side network effect on the left, where new customers are increasingly motivated to subscribe as function of the quantity of existing customers (causal loop R1).

We will now provide a mathematical model to understand network effects. Based on [21], we can describe total attractiveness A_{P1} of a PaaS as the product of various variables of attractiveness Aj, e.g. price, reliability, but also attractiveness resulting from network effects, e.g. on the number of subscribed users.

$$A_{P1} = \prod_{j=1}^{n} A_j \quad (2)$$

The market share of a platform can be described as

$$M_{S1} = \frac{A_{P1}}{A_{P1}+A_C} \quad (3),$$

where A_{P1} stands for platform P1's attractiveness and A_C for the aggregation of all other platforms' attractiveness.

The attractiveness resulting from the discussed network effect can be described as [21]:

$$A_i = \exp(s_{S0} * \beta_N) \quad (4),$$

where s_{S0} is the sensitivity to the subscription base and the normed subscription base β_N (which is the subscription base relative to the threshold). The threshold is the critical mass of subscribed customers above the subscription base which has impact on attractiveness. Fig.1 illustrates a major advantage of network effect related attractiveness: Through the subscription base, it scales up with the causal loop of the network effect. In our simulation, the effects of *other Factors on Attractiveness on Platform 1* are kept constant ('c'). The relative subscription base (meaning: relative to the total subscriptions in the market) of platform 1 can be defined as

$$\beta_{rP1} = \frac{\beta_{NP1}}{\beta_{NP1}+\beta_{NP2}} \quad (5),$$

Substituting attractiveness in (3) with (4) and (5) leads to

$$M_{S1} = \frac{1}{1+c*\exp\left(s_{S0}*\beta_{NP2}*\left(1-\frac{1}{\beta_{rP1}{}^{-1}-1}\right)\right)} \quad (6),$$

which displays the market share of platform 1 as a phase-plot of relative subscription, with the aggregated non-network related attractiveness 'c' as amplifier. Fig. 2 shows this phase plot. In a first step we focus on the network impact and set c=1. On the 45° line, the system is in equilibrium, as the current market share is equal to the relative subscription base. In cases where the slope is >1, small changes in the relative subscription might cause significant changes in market share (*instable equilibrium*). On the other side, areas where the slope is strongly <1 do not give much scope of influence (*stable equilibrium*).

The network dynamics described above is the consequence of the network effect R1 (fig. 1) in conjunction with the counter-acting loop of share saturation (B1), caused by the activity of competition. However, there are more loops in this example. An important one is that cross-sided network effects add on through providers of Web applications in a complementarity effect (indicated through a complementarity loop R2 displayed partly in fig.1). The PaaS' attractiveness to the Web application provider is also exponentially dependent of the subscription base. The igniting effect to the ecosystem happens, when the platform attractiveness to customers is further increased through an increased provision of Web applications. In return, the increased customer-base will again increase platform attractiveness to providers. It is plausible that the more cross-supporting loops are created, the stronger is the effect.

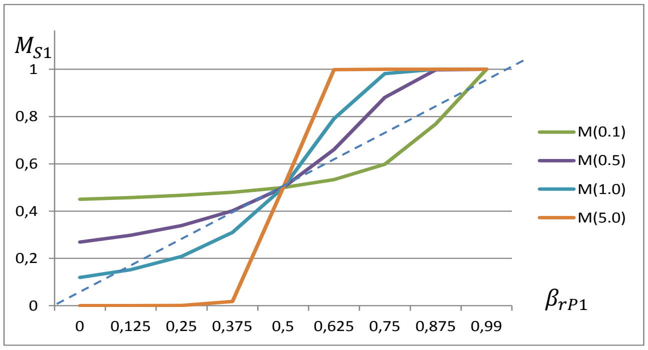

Figure 2: Market share of platform 1 (for sensitivities 0.1, 0.5, 1, 5) of as a phase-plot of relative subscription base based on [22]

Once, a dynamic process is set off, loops may self-fertilize and potentially grow towards a market dominating position (*lock-in situations*) over time (e.g. the stable position of Apple's App Store, making it particularly difficult for challengers to successfully introduce competitive solutions). However, the same inertia is faced when trying to initiate system dynamics. Prerequisite to success is an initial value proposition (*base value*) to attract a critical mass of first movers among Web application providers and consumers that eventually ignites a dynamic loop.

3. GOVERNANCE IN PAAS ECOSYSTEMS

In order to exercise control over the PaaS ecosystem and the Web application provision we need to derive governance tasks from those design challenges. It is to be noted that our governance definition is broader than in the traditional SOA governance view. In our understanding, the term embraces exercising control and influence over services, activities and ecosystem participants (e.g. Web application providers, consumers). Subsection 3.1 suggests governance tasks resulting from network-related aspects in design challenges DC1-4. In the subsequent subsection, we suggest a control-related governance task derived from design challenge 4.

3.1 Major network governance tasks

As consequence of the above described network effects, the following three major network governance tasks arise: *(a) Governing Dynamic Loops:* The PaaS/Web application scenario offers a multitude of possibilities for causal loop design. The notation shall allow for easy conception of such causal loops. The meta-model needs to be conceived in a way that editors, designed around the notation shall be able to offer an automated loop discovery function. *(b) Identifying and positioning Base Value:* The previous section showed that base values of sufficient magnitude are required to set off the loops. The notation needs to give support for discovering and dimensioning them. *(c) Allowing for Scalability:* Once a dynamic loop is set off, its proliferation requires technical environments, able to quickly adjust to increasing demand. At the same time occurring drop-offs demand the facility to quickly release resources. These capabilities of 'scalability' need to be designed into the system, where required.

3.2 Major control tasks

Given the inter-organizational characteristics of Cloud-based service compositions and the autonomy of its participants, the power of a protagonist (the participant, whose view point is modeled, e.g. the platform provider) to control or influence other participants is limited. We define the power, which embraces power of control and power of influence as *stakeholding power*.

Delimitation of zones of controllability: The protagonist's stakeholding power diminishes with every additional level of indirection as it steps further away from its proprietary domain. For a clear delimitation of the protagonist's stakeholding power, we introduce 3 zones with different levels of authority [19, 20]: (a) The center domain (*Control Area*) allows highest observability and direct control on all transactions and activities. Quality can be assured, either through directives or through enforcement in this area. (b) The consecutive zone (*Influence Area*) describes the area where the protagonist can only indirectly influence activities through targeted information and incentives. (c) The outer zone (*Noise Area*) embraces those participants who are indifferent or opposed to the protagonists activities.

Handling Self-Organization: We know from system theory that autonomous participants self-organize over time. In our context they are SaaS-providers, adapting their value in pursuit of long-term profit optimization. This causes the governance task of influencing 'self-organization' in the Influence Area. The necessary approach is to nourish external creativity for the price of reduced observability and controllability compared to their dependent counterparts within the boundaries of the Control Area. This governance task will be accomplished through enforcing and incentivizing control mechanisms within a suitable system design. Applied control mechanisms can be understood as the operationalization of governance tasks.

4. META MODEL AND NOTATION

The design goal of DYNO is to enable business analysts and service engineers to layout their PaaS infrastructure in accordance with the context of the inter-organizational characteristics of the PaaS ecosystem. We therefore develop a notation that is able to circumscribe weaknesses (e.g. lack of base value) and leverages in opportunities (e.g. network loops) in a dynamic network context.

We formulate the basic concepts of abstraction (CoA) for DYNO, building on section 3: *CoA1:* All PaaS ecosystem participants (service providers, service consumers) and their respective activities must find a representation in DYNO models; *CoA2*: Models must incorporate relationships that reflect both transactions and influences between PaaS ecosystem participants; *CoA3:* Models must be able to incorporate points to exert control and respective control mechanisms for PaaS providers; *CoA4:* Each ecosystem participant (providers, consumers) must be ascribed to a clearly defined zone of controllability; *CoA5:* All ecosystem participants may be ordered in groups.

We integrate all abstraction rules in one meta-model (fig. 3) based on UML class diagrams [10]. The underlying dependencies are accomplished through constraints, modeled in OCL [11]. To fulfill the given concepts of abstraction (CoA1-5), we suggest a meta-model that consists of five core elements. They are *Protagonist Control, Controlled Element, Directed Relationship, Location* and *Division*. All resulting, instantiable DYNO-elements are depicted in fig. 4.

DYNO models will not explicitly depict one central element with control center functionality. It uses *Protagonist Control*, which is to be understood as a set of decentralized toeholds of control and influence, exerted onto the respective Controlled Elements and Directed Relationships. Controllability of a Controlled Element or a Directed Relationship depends on the respective position: The Control Area assigns full hierarchical control to the protagonist and can be segmented into divisions. All Controlled Elements (apart from the Gateway) and parts of Directed Relationships (source and/or target-side) within the control area are controllable. The existence of the attribute *controllable* is depicted through a screw head. In cases, where the control functionality is activated

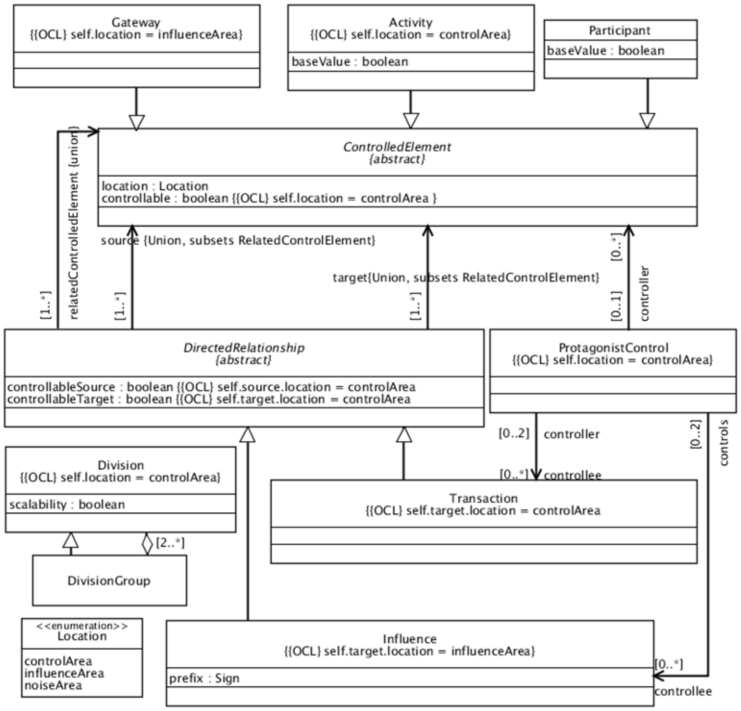

Figure 3: Simplified Meta-model for DYNO

by the Protagonist Control, the screw head is filled black. *Controlled Element* is an abstract class, refined through inheritance into the node-subclasses *Gateway, Activity, Participant* as well as into the hierarchical elements *Activity Group* and *Participant Group*.

All but the *Gateway* can represent a base value, aimed at setting off causal loops (represented through the symbol *ß*). The *Gateway* aggregates influences in the Influence Area into one resulting outgoing influence. It is depicted through a diamond symbol with included '+'. An *Activity* stands for any kind of service orchestration and is represented through a hexagon. The *Participant* subclass (symbolized through a rounded rectangle) embraces any possible participant in a PaaS ecosystem scenario, e.g. external or internal service providers and consumers, but also standardization bodies or competitors. Depending on its position, a *Participant* can be directly controlled, influenced, or remains unaffected by the Protagonist Control. Our intent is not to only build on explicit collaborations, but to leverage on clusters of Participants, which are high in number and often not explicitly known. This leads to the requirement of further refining a controlled element into *Activity Group* and *Participant Group*. All relationships within the DYNO meta-model are directed. The reason behind this is that tangible transfer of a value happens either in form of service (*transaction*) or information exchange (*influence*).

In previous work and based on market studies, we categorized different control mechanisms, which can be exerted through Protagonist Control, when a control functionality is activated. In the extended Meta-model, they are aggregated into the Protagonist Control. There is (a) *Restrictive Control*, embracing access regulations and limitations. (b) *Sanctional Control* puts together all enforcing actions, exerted after service deployment. (c) In contrast to those enforcing mechanisms, *Motivational Control* is incentivizing, operating through development support, community building, funding, etc. (d) *Informative Control* follows an incentivizing approach. Information, e.g. consumer behavior,

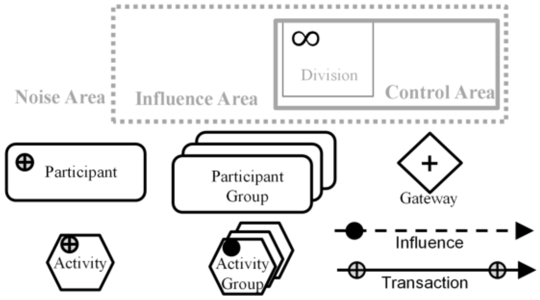

Figure 4: DYNO Elements

platform evolution, value creation opportunities is passed from a service receiving Participant to a service supplying Participant for the sake of empowering the latter for service optimization. (e) Market Regulative Control comprises consumer-based service ratings. (f) For the sake of completeness we introduce *Prescriptive Control*, describing direct control, exerted on structurally dependent elements. For substantiation and case studies on control elements, we refer to our respective publications [17, 18, 19, 20].

5. EXPERIMENTAL EVALUATION

An evaluation of our approach to modeling PaaS network effects with DYNO needs to address the four fundamental design goals. Therefore, we have followed a strategy of *experimental evaluation* building on a *case study experiment*. We consider this a good compromise between the crucial ability to *observe* a real world PaaS network design activity with *local control* over the experiment on the one hand and the given limitations of *replication* on the other.

Subjects of this experiment were the members of a project team faced with the design of a platform concept for e-Learning-as-a-Service for and in cooperation with S. Chand Group [22]. We were interested in *attributes* that correlate with answers to our fundamental questions, i.e. identification and optimization of dynamic loops, base values, points requiring scalability and means of governance. The experiment was conducted with two different *treatments* with variations in the *factor* of design and analysis methods: first the team utilized their conventional methods and second they were equipped with DYNO. In order to evaluate DYNO, we have compared the design artifacts resulting of the first treatment against those of the second. Furthermore, we have interviewed the members of the project team and discussed about their experiences within the experiment.

5.1 Context of the Case Study

S. Chand Group created the subsidiary S .Chand Edutech (SCE) with the mission to design a platform that (a) provides e-Learning-as-a-Service, (b) scales with growing quantities of users and providers as well as seasonal effects, without the need of stockpiling of IT infrastructure; (c) leverages on SCE's existing position in the academic text book sector in terms of width and depth of portfolio and in terms of distribution channel; (d) benefits from the existing multitude of e-Learning developing companies without a distribution channel;(e) offers content that responds to a diversity of academic classes;(f) works around shortcomings in national telecommunications infrastructure; (g) assures suitable quality levels.

5.2 Modeling & Analyzing the SCE platform

The case study is about designing the SCE PaaS offering from a network perspective. In one of the treatments of our case study

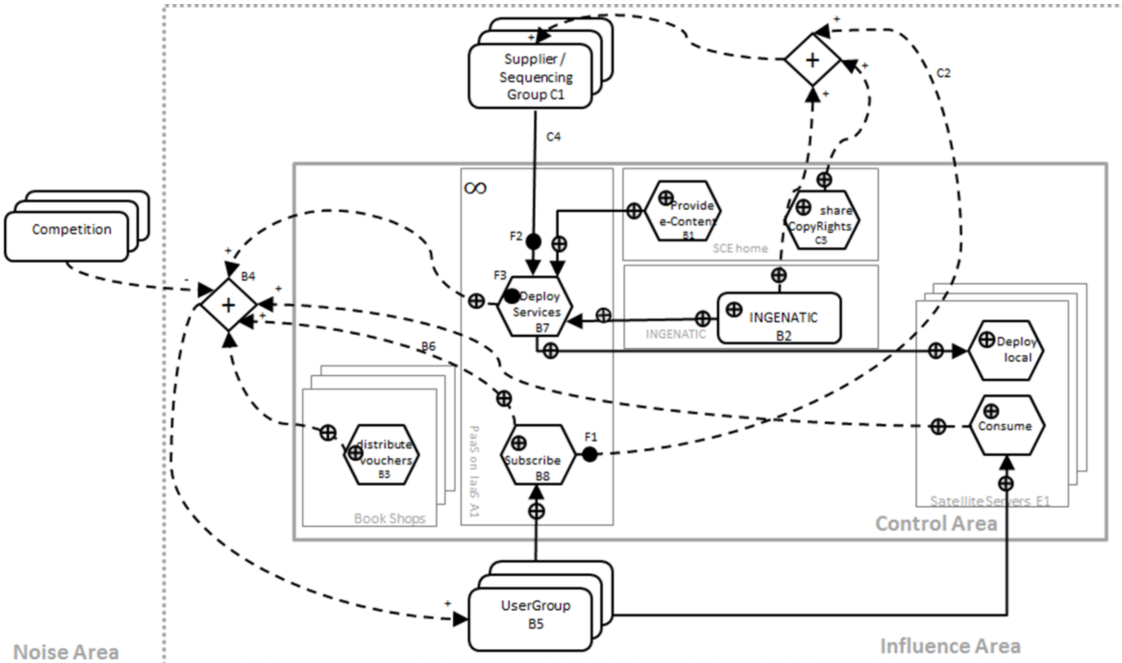

Figure 5: DYNO Model for S.Chand Edutech's PaaS and ecosystem

experiments, we have equipped the project team with a simple implementation of the DYNO notation in order to accomplish the design task. The outcome of this experiment was a concrete DYNO model that is shown in fig. 5. In the following we describe the artifacts of this model in more detail.

Demand-side Base Values: Several base values have been identified and modeled to set off this initial loop on the demand-side. Most important are electronic content provision through digitalized versions of S. Chand's academic books (fig. 5, Activity B1) and the take-over of an established e-Learning developer to further stimulate the network effect through increased attractiveness (fig. 5, Participant B2). The positive influences of all value contributions are aggregated within the Gateway B4 and take effect on potential users (fig. 5, Gateway B4). Competitors weaken this accumulated positive impact. As those users are receptive to influence, they are positioned in the Influence Area. The symbol for Participant Groups is applied to emphasize that not an explicit group of users is addressed, but a non-quantified group. Once the first users subscribe, the initial loop – a network effect (fig. 5, Influence B6) – is started off, as the number of inscribed users further stimulates a platform's attractiveness. However, the aggregated attractiveness, impacting on the potential users needs to be strong enough to overcome initial inertia and get the initial loop going (fig. 2). Control mechanisms need to be placed to steer and amplify this effect.

Supply-side Network Effects and Base Values: SCE choose to delegate content creation into a platform ecosystem: Instead of providing ready-made e-Learning courses, the Web application providers (fig. 5, C1) provide fine-granulated Sharable Content Objects (SCOs), which can be sequenced into courses following Sharable Content Object Reference Model [6]. Having set off the dynamics on the demand-side through own base value, the suppliers are already attracted by the volume of users, promising potential for turnover (fig. 5, Influence C2). This is strengthened through SCE's granting permission to use copyrighted material (fig. 5, Activity C3) and a multitude of design templates, provided by INGENATIC (fig. 5, B2). Once the suppliers deploy their services (see fig. 6, Transaction C4), the complementarity loop is closed. As this stimulates the network effect, the designers have initialized an ongoing effect of two loops in positive – meaning self-enforcing – reciprocity.

Scalability Requirements: SCE decided to deploy the e-Learning PaaS on a scalable Infrastructure-as-a-Service (IaaS), provided by a third party provider (fig. 5, Division A1). Indian Internet connectivity is of good quality in the metropolitan areas, but many university locations still suffer from a limited bandwidth. Working around this shortcoming and benefiting from the existing University-WLANs, SCE places proprietary 'satellite servers' into the Intranets of partner universities (fig. 5, E1, Division 'Satellite Servers'). The servers are synchronized through delta-coding with the central server. The lacking infinity symbol pinpoints that the servers are a non-scalable segment within a loop.

5.3 Governing the SCE Ecosystem

Working with external provisions implies a completely new paradigm of quality control and of enforcing one's goals. There are many possible control points and corresponding sets of control mechanisms in the model – all should be tooled up. In the following we exemplify some examples: (a) SCE applied informative control on the Influence (fig. 5, e.g. Influence F1) to amplify its impact of the existing user base, i.e. they provide customized information per Web application provider on user preferences help providers to tune their offers on the user requirements. (b) Through transaction C4 (see fig. 5), the provision of e-Learning content is restricted. SCE applies restrictive control mechanisms that verify compliance with the stipulated programming model (SCORM, LOM).

5.4 Evaluating the DYNO notation

The initial design results of the project team significantly differed from the results after the implementation of DYNO. The differences can be grouped into those related to base value contribution, to specific causal loops, to control and to scalability. Initially, the only base value planned in was the INGENATIC content. Based on the DYNO-model, the distribution of vouchers in book shops was added (fig. 5, B3). The supply side, which was originally planned as a pure outsource concept was redesigned into a complementarity loop, complemented with a sharing of copyrights as another base value (fig. 5, C3). In order to further diversify the provisions from the supply side, the original static content provision was split into the supply of sequencing (IMS-manifests) and of sharable content objects (SCOs). Control in the previous solution was unstructured and mainly focused on controlling the outsourced development. SCE speaks now of a holistic Control-Concept. Discussions with the project team confirmed that DYNO's emphasis on loops and base values makes sense with respect to platform effectiveness. Furthermore, DYNO's supporting set of control mechanisms was appreciated as well as the fact that a structured approach to governance was made possible, allowing to go beyond of what is normally done.

6. RELATED WORK

Ongoing related research related to network notations has several directions. Two network notations resulting from the European S-Cube Project aim at 'developing, monitoring and optimizing SOA-enabled business processes in service networks" [23]: (a) the Service Network Notation (SNN) to model and describe service networks and (b) the Graphical Service Network Modeling Language helping solve optimization problems for process-based KPI and SLA [1, 13]. The e3value group models value flows in service networks. The group proposes generic solutions, so called *control patterns* [9]. The underlying transactional design is helpful to describe and depict direct control relationships. Both related research streams, however, lack the ability to model control complications with respect to dynamic network effects.

7. CONCLUSION

This paper introduces a notation to support business analysts and service engineers in the design of competitive PaaS ecosystems consisting of Web application providers and customers. We thereby seek to introduce network complexity theory into the domain of software design, i.e. software platform design. In doing so, the DYNO notation attributes explicit attention to the governance of network effects and to the placement of base values. Its goal is to set off initial causal loops and to continuously govern the ecosystem. Providers and consumers are designed as a self-organizing, but governed dynamic network around the PaaS. DYNO depicts the available toeholds for system control and suggests a set of suitable incentivizing and enforcing control mechanisms. A mapping of ecosystem participants and activities to specific areas illustrates the platform's specific power of control or influence. In further research we want to shed light on sensitivity and thresholds related to network-attractiveness, building on recent findings in dynamic network science [5, 19].

8. REFERENCES

[1] Danylevych, O., Karastoyanova, D., Leymann, F. 2010. *Service networks modeling: A SOA & BPM standpoint*. JUCS, 26 (13), pp.1668 – 93.

[2] Fischer, R., Scholten, U., Scholten, S. 2010. *A reference architecture for feedback-based control of service ecosystems,* Proc. 4th International , IEEE Conference on Digital Ecosystems and Technologies, Dubai, pp. 1 – 6.

[3] Föllinger, O. 1990. *Regelungstechnik*. Hüthig-Verlag.

[4] Friesz, T.L. 2007. *Network Science, Nonlinear Science and Infrastructure Systems*, Springer, New York.

[5] Friesz, T.L., Taeil, K., Kwon, C. 2011. *Approximate Network Loading and Dual-Time-Scale Dynamic User Equilibrium*. In: Transportation Research Part B,45, (1), Jan. 2011, pp. 176–207.

[6] IMS standard, http://www.imsglobal.org/content/packaging/, retrieved Aug. 2011.

[7] Katz M., Shapiro C. 1985. *Network externalities,competition, and compatibility*. American Economic Review 75: 424-440.

[8] May, N., Scholten, U., Fischer, R. 2011. *Towards an automated gap analysis for e-Service portfolios*. Proc. 8th International Conference on Services Computing, Washington.

[9] Kartseva, V., Gordijn,J., Tan, Y.H. 2007. *Value-based design of Networked enterprises using e3control patterns*, 15th IEEE International Requirements Engineering Conference, pp. 91 – 100.

[10] Object Management Group, 'Documents associated with UML 2.3, May 2010, http://www.omg.org/spec/UML.

[11] Object Management Group, 'Documents Associated with Object Constraint Language, v2.3 - Beta 2," Mar 2011, http://www.omg.org/spec/OCL/2.3/Beta2.

[12] Object Management Group, 'Documents Associated with Business Process Model and Notation (BPMN) v 2.0," Jan 2011, http://www.omg.org/spec/BPMN/2.0.

[13] Papazoglou, M. , Bitsaki, M., Danylevych, O., van den Heuvel, W-J., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C. 2008. An architecture for managing the lifecycle of business goals for partners in a service network, Towards a service-based Internet. Springer, pp. 196 – 207.

[14] Prigogine, I., Nicolis, G. 1977. *Self-Organization in nonequilibrium systems: From dissipative structures to order through fluctuations*. Wiley, New York, 1977.

[15] Shapiro, C., Varian, H.R. 1999. *Information rules, A strategic guide to the network economy.* Harvard Business School Press.

[16] Scholten, S., Scholten, U. 2010. *Platform-based innovation management: Directing external innovational efforts in complex self-organizing platform ecosystems*. Proc. PICMET '10, IEEE, Bangkok, pp. 1 – 12.

[17] Scholten, S., Scholten, U., Fischer, R. 2010. *Composite solutions for consumer-driven supply chains*. Proc. Supply Management Symposium'10, Supply Management Research, Springer, Würzburg, 2010, pp. 277 – 301.

[18] Scholten, U. 2010.*Service Level Management in Platform Ecosystems.* Proceedings of the INFORMATIK 2010 conference on Service Science, Leipzig

[19] Scholten, U., Fischer, R., Bojkov, D., May, N. 2011. *Supply chain control building on emergent self-organizing effects*. Proc. 4th Academic Symposium on Supply Management, Springer, Würzburg, pp. 311 - 335.

[20] Scholten, U., Fischer, R.; Zirpins, C. 2009. *Perspectives for Web service intermediaries: How influence on quality makes the difference.* Proc. 10th International Conference on Electronic Commerce and Web Technologies (EC-Web), (Linz, Austria, 2009), pp. 145-156.

[21] Sterman, J.D. 2000. *Business dynamics, systems thinking and modeling for a complex world.* McGraw-Hill.

[22] S.Chand Group Homepage, http://www.schandgroup.com/, retrieved Aug. 2011.

[23] S-Cube, www.s-cube-network.eu, retrieved Aug. 2011.