3-5-2015

Using System Dynamics Models to Understand and Improve Application Landscape Design

Alexander Schneider

Anna Gschwendtner

Florian Matthes

Using System Dynamics Models to Understand and Improve Application Landscape Design

Alexander W. Schneider, Anna Gschwendtner, and Florian Matthes

Technische Universität München, Munich, Germany
{schneial,gschwena,matthes}@in.tum.de

Abstract. Application landscape design has become a key challenge for enterprises. For further exploration of related enterprise architecture benefits establishing shared mental models among all application landscape designers is required, i.e. architectural thinking. Thus, to complement existing approaches by modeling human behavior and decision effects which form implicit application landscape evolution principles, we propose the use of System Dynamics. We derive five guidelines from literature for developing a corresponding method. To exemplify the approach, a concrete causal loop diagram on the topic of technological standardization is presented. A subsequent evaluation based on expert interviews demonstrates the model content validity as well as the modeling method's suitability to foster communication among different communities of practice.

Keywords: Enterprise Architecture, Application Landscape Design, Standardization, System Dynamics, Causal Loop Diagram

1 Introduction

For many years, information systems (IS) within companies have been growing uncontrolled resulting in proliferation, i.e. an application silo architecture [1]. One reason for implementing enterprise architecture management (EAM) within an organization is to control the evolution of its application landscape. For example, EAM can increase the overall transparency by rigorously modeling and documenting the as-is architecture [2]. In addition, enacting architectural principles can constrain the future development of the application landscape towards aligned goals [3]. Furthermore, by modeling concrete to-be architectures, concrete roadmaps for the application landscape evolution can be developed to constrain IT projects [4]. However, these approaches focus on the symptoms of uncontrolled application landscape evolution and assume that a single designer or architect is in place.

Regarding organizations as complex adaptive systems has become prevalent in several research areas [5–7]. Although there exists no agreed-upon definition of complexity [8], several properties are frequently attributed to complex systems, including emergence [6, 9], non-linear behavior [10] and path-dependence [11] and they are often made up by autonomous agents without any central control [12]. Therefore, in a

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

Schneider, A.; Gschwendtner, A.; Matthes, F. (2015): Using System Dynamics Models to Understand and Improve Application Landscape Design, in: Thomas. O.; Teuteberg, F. (Hrsg.): Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015), Osnabrück, S. 348-362

complex system – such as in an organization – understanding causal dependencies between the decisions of different actors, i.e. architectural thinking [13], is essential. Knowledge about how actors, such as business unit managers, enterprise architects and software developers, influence each other's decisions leads to improved alignment and coordination of the actors' decisions. Hence, when considering principles guiding architectural evolution, we need to account for both explicit principles defined by enterprise architects [3, 14] as well as implicit principles emerging within the complex system. To cover also such implicit principles and model the causal relationships influencing an enterprise architecture, several authors already proposed to extend enterprise architecture models with System Dynamics [15–17] and social aspects [18].

To get one step further towards a method enabling enterprise architects to develop System Dynamics (SD) models [19, 20], we apply a four step research approach according to the design science paradigm [21, 22] following the method described by [22]. First, we analyze the state-of-the-art in SD modeling within the context of EAM by performing a structured literature review to motivate the research and describe the problem. Second, based on existing work on SD modeling, we derive design guidelines for a method enabling enterprise architects to develop SD models to define the objectives of the solution. Third, to demonstrate the guidelines' applicability we develop a concrete CLD describing standardization activities occurring within companies based on input gathered from literature. Fourth, we evaluate the prototypical CLD based on a series of expert interviews. Fifth, we derive additional design guidelines accounting for EAM specifics based on the analyzed interview results. We conclude this paper by describing possible future research activities.

2 Foundations and Related Work

2.1 System Dynamics Modeling

"Without modeling, we might think we are learning to think holistically when we are actually learning to jump to conclusions" [23]. System Dynamics can serve hereby as powerful tool by integrating both mathematical and methodological aspects. Initially proposed by Forrester [19], System Dynamics is an approach to understanding the behavior of complex systems over time. To support understanding, two popular model types are offered: Causal Loop Diagrams (CLDs) and Stock-and-Flow Diagrams (SFDs) [24]. Depicting the structure of a system, CLDs provide a macroscopic view on causalities within a system. To get a more detailed and quantitative view on a system, a CLD can be transformed to a simulation-enabling SFD [25].

A CLD composes of relevant system elements and their corresponding causal system structures or feedback theories which are denoted by arrows (see e.g. Fig. 2). A plus sign indicates a positive relation between two variables, i.e. they change in the same direction. A minus sign indicates a negative relation, i.e. they change in the opposite direction. Consequently, a closed loop of such relations has either a balancing or an unbalancing, i.e. reinforcing or diminishing, effect on involved system components. The loop type is indicated by the corresponding sign at its center. As behavioral effects might not occur immediately, a feedback arrow can be complemented by a time delay symbol, i.e. two parallel lines, if necessary [20]. To ease the diagram's readability, feedback loops can also be named. Subsequently, the modeling process requires an iterative verification of the respective hypotheses on causal system structures and feedback theories [26].

2.2 Related Work on System Dynamics in the Context of EAM

In order to identify related work, we systematically searched a number of scientific databases in accordance with [27]. Querying IS journals (AIS senior scholars basket), academic databases (ScienceDirect, IEEE Xplore, ACM DL, GoogleScholar, SpringerLink) and IS conferences (AMCIS, ICIS, ECIS), we got 23 initial hits on combinations of the search terms "Enterprise Architecture" or "Enterprise Architecting" and "System Dynamics", "Causal Loop" or "Causal Model", limited to title, keywords and abstract. After removing both irrelevant articles and duplicates, seven articles remained. To avoid omitting relevant references, we subsequently conducted a forward and backward search.

Assimakopoulos et al. [28] use SD and problem structuring methodologies to analyze a virtual enterprise architecture constructing wireless payments. After the conceptual planning, different situations of the virtual enterprise life cycle are simulated. Furthermore, the development of a concrete SFD is presented.

Stressing the importance "to explicate and analyze the whys" of an enterprise architecture, i.e. causal and intentional aspects, Sunkle et al. [29] propose an approach integrating the use of the i* language and SD models, in particular SFDs.

Applying and extending Systems of Systems Engineering, Wojcik and Hoffmann [30] developed a framework representing complex enterprise characteristics. Describing the acquisition process at a very abstract level, they introduce a hybrid approach based on game theory and highly optimized tolerance.

Golnam et al. [16] present an approach to support choice making by integrating SD and Systemic Enterprise Architecture Methodology. Based on an as-is-architecture value network conceptualization they develop a SFD for supply chain management, respective scenarios and simulations. Furthermore, the iterative development of a to-be architecture design is demonstrated.

The modeling methodology described by Rashid et al. [17] integrates enterprise modeling based on CIMOSA, System Dynamics modeling based on CLDs and simulation modeling. Based on the most important organizational characteristics a CLD is developed to capture dynamic causal aspects. Finally, simulation models are used to quantify identified effects with respect to organizational performance.

Concluding, current literature focuses on problem identification and offers special purpose SFDs, some CLDs and respective simulations. Qualitative aspects as represented in causal loop diagrams are underrepresented and a consistent method of SD model development within the context of EA management is missing.

2.3 Shared Understanding Impacts on Application Landscape Design

Assuming that application landscape design is a wicked problem [31–33] within a complex adaptive system for which only good (not true) solutions exist which have to be collaboratively developed by a variety of stakeholders, shared understanding and consensus building become vital. In fact, "consensus building among stakeholders is increasingly common as a way to search for feasible strategies to deal with uncertain, complex, and controversial planning and policy tasks" [34]. There is empirical evidence that, after consensus building, stakeholders build shared understanding. Hereby, shared definitions of the problem and agreement on data and models support the coordination action and reduce areas of conflict [35]. Furthermore, there is strong evidence that stakeholders build up trust which has recently been identified as a key to more effective architectural thinking [13].

Shared mental models allow people to make predictions, understand phenomena and more quickly decide upon an appropriate course of action [36]. This also holds true for the context of enterprise architecting, since cognitive coordination of tasks, i.e. having shared mental models, is important for effective enterprise architecting by coordinating stakeholders implicitly [37]. Although there exist a variety of tools and techniques to facilitate the emergence of shared mental models among application landscape designers SD models are promising for this context. As visualized in Fig. 1, decisions are based on information feedback from the real world but individual strategies and rules which stem from the individual mental model of the world determine how one uses this feedback. By creating virtual worlds SD models allow decision makers to experiment, learn and enhance their mental models [20]. Gaining such feedback from the real world is more difficult and time-consuming. Furthermore, they model the actual behavior of people as well as "side-effects" of their actions. By making tacit knowledge about relevant phenomena influencing application landscapes explicit, this knowledge is available in concrete decision situations whereas otherwise people would, instead of systematically considering all causes and effects, tend to think mono-causally [38].

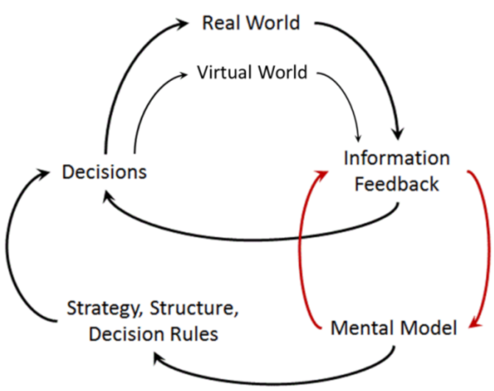

Fig. 1. Mental model building through virtual worlds [20]

3 Design Guidelines for System Dynamics Modeling

Although System Dynamics (SD) modeling has a long tradition in science [19] and already proved to be valuable in practice [20], practitioners within the EA discipline have not yet adopted this technique. One reason therefore might be the absence of a concrete method for developing SD models within companies. Existing methods [20, 26] are very general in nature, do not account for the specifics within companies and assume the presence of experienced modelers and workshop facilitators. Designing a method consisting of concrete activities, their sequence of execution, a role model, a meta-model and appropriate techniques is a difficult task. In order to get one step further in achieving this goal, we derived five design guidelines for such method grounded in general problem solving techniques, EA context as well as EA artefact requirements.

3.1 Guideline 1: Distinguish a Divergent and a Convergent Creation Phase

Several complex problem solving techniques suggest to start with a divergent phase followed by a convergent phase [39, 40]. Within the divergent phase the goal is to identify as many sensible solutions as possible and to hereby avoid any bias. In the subsequent convergent phase, the number of potential solutions is systematically analyzed and inappropriate solutions are rejected. In order to account for the divergent phase, input for SD models should be collected decentralized, i.e. each modeler should provide his first input independently and unbiased from other inputs. Although appropriate for the convergent phase, conducting a workshop to model collaboratively without an experienced facilitator is consequently not sufficient.

3.2 Guideline 2: Gather Input from Heterogeneous People

There is empirical evidence that solutions of difficult problems become better the more people try to solve it due to their different views implied by their diverse backgrounds [41]. In addition, the more people are involved in building shared mental models the more effective these mental models are and the more people are able to benefit from their existence. Application landscape design is about coordinating the architectural development across levels and departments in an organization [13], hence, a multitude of stakeholders is involved. Therefore, to get an accurate and holistic overview, SD models have to include as many diverse viewpoints as possible.

3.3 Guideline 3: Model for the Purpose of Learning

Since each model has a clear purpose, it cannot include all aspects present in real world [42]. To ensure a feasible scope and timely results [20], this holds true also for SD models. As a consequence, a SD model cannot be complete [43] and therefore modelers have to identify and maintain the model boundary. For enterprise architecture artefacts, accurate granularity and consistency are more important than actuality and completeness [44]. Thus, SD models should focus on feedback generally unaccounted for in respective mental models. In addition, if used as communication tool with the intention of teaching, SD models have to respect their related cognitive load [45]. Hence, they should limit both intrinsic and extraneous cognitive load by limiting modeled phenomena as well as refrain from adding irrelevant information, repetitions, numerous references or visual distractions.

3.4 Guideline 4: Ensure Transparency

Correctness has been identified as an important property of any EA artifact and used data sources as one potential reason for erroneous products [46]. As mental models are the primary data source for SD models their correctness cannot be proven. Instead, traceability of the origin of model elements [47], i.e. causal effects, needs to be provided to foster general comprehensibility and trust. Therefore, the origin of each causal effect has to be documented to enable traceability. This includes at least the technique of elicitation, date and role of the modeler.

3.5 Guideline 5: Validate with Data

As pointed out by [20], validating and enhancing qualitative models by simulation is essential to foster learning. Although system thinking techniques [48] and soft systems analysis approaches [49] are able to enhance our intuition about complex situations, they do not allow conducting experiments. Especially if experimenting is infeasible in the real world – as it is for application landscape design – simulations become a suitable tool to understand complex system mechanisms without relying on feedback from real world, thus, enabling faster learning.

4 Exemplary CLD Modeling

To exemplify the approach described in the previous sections we developed a CLD for the area of technology standardization. Despite the existence of various definitions of the term standardization, we consider a technology (e.g. database management system, web application framework, operating system) to be a standard technology, if the technology has been defined explicitly to be standard in a given organization.

4.1 Modeling Approach

In accordance with [24], we started the modeling process by deriving a set of dynamic hypotheses. While there exists a broad range of suitable data-gathering techniques [26], we derived the initial set of dynamic hypotheses from literature and personal experiences gained from several action research endeavors [31, 50]. The initial brainstorming session of the author team was done without interaction. The identified hypotheses were then discussed and harmonized. By this we adhered to guideline 1. Each hypothesis was then substantiated by respective literature (see Sec. 4.2). Subsequently, we structured these hypotheses and created a visual representation of the gathered information (see Sec. 4.3). By this we adhered to guideline 4 and ensured transparency. The hypothesis generation process has been stopped after identifying five concise and substantiated hypotheses in order to allow readers to learn from the model (guideline 3) without overwhelming. Since each hypothesis can be mapped to a different role within the organization we ensured their heterogeneity (guideline 2).

4.2 Dynamic Hypotheses

Network effect. People focusing on one activity, e.g. operating a specific database product, increase their skill disproportional to those focusing on a set of diverse activities [51]. Therefore, if an organization employs highly qualified IT operators, the respective products receive an increasing attractiveness for both solution and enterprise architects. As a consequence, these products are preferred over non-standard products and hereby increase the standardization rate over time (see Fig. 2).

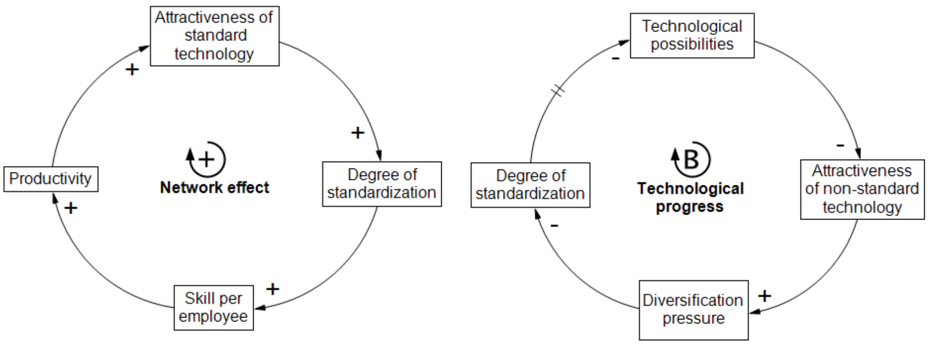

Fig. 2. Causal loop diagrams modeling the network effect and technological progress

Technological progress. A high degree of standardization has a declining effect on the scope of technical possibilities [52]. As technology diffusion is a slow process and its effects are likely to materialize after considerable delay [53], the notifiable impact of standardization on technical possibilities is also delayed. The narrowing technological scope enforces the attractiveness of non-standard technologies [54]. Thus, the diversification pressure increases accordingly. Hence, the technological progress relations identified create a balancing feedback loop (see Fig. 2**Fig. 2.** Causal loop diagrams modeling the network effect and technological progress).

Shadow IT. Aside from the direct effect on standardization, diversification pressure also impacts the incentive of departments to develop local solutions which are not corresponding to defined standards. Reasons therefore include increasingly tech-savvy users, easy access to web-based solutions and available end user computing tools [55], but also misalignment of business demands and provided IT solutions [56].

Hence, with a rising number of local, i.e. not centrally managed, solutions the actual standardization degree within the application landscape declines.

IT cost cutting. Standardization can help to reduce IT costs [14, 54], which is an incentive for an increased standardization degree [57]. An increasing request for standardization enforces constraints and hereby the strictness of project requirements, while the latter positively influences the standardization degree.

Maintaining the decision-scope. Due to high switching costs and network effects, e.g., increased employee productivity and incompatible technologies, lock-in effects hinder enterprises from changing suppliers in response to both predictable and unpredictable changes in efficiency [58]. The augmenting dependency thus limits the respective decision scope. As this scope limitation is a long and slow process, the actual perception of the limited decision scope occurs after a certain delay. The limit perceiving subsequently increases the pressure on diversification [54].

4.3 Integrated Model Construction

As exemplified in Fig. 2, each of the aforementioned phenomena can be represented by a CLD. Accordingly, as visualized in Fig. 3 an integrated CLD can be developed by unifying recurring variables. The integrated model shows how the different phenomena interrelate and simultaneously affect the degree of technological standardization within application landscapes.

5 Model Evaluation and Additional Guidelines

To ascertain if the phenomena modeled in the previously introduced causal loop diagram (CLD) can be observed in practice, we conducted interviews with experienced practitioners. In addition to the model verification, its suitability to serve as means of communication was assessed.

5.1 Evaluation Setting

Table 1 provides an overview about the background of the experts participating in our evaluation. None of the interviewees was familiar with the System Dynamics in general or CLDs in particular. Instead, all interviewees were familiar with standardization efforts currently taking place in their organization. While we cannot claim our sample to be representative, we can expect statements of high quality as respondents have diverse backgrounds [41] and are faced with standardization issues during their daily work. Structured interviews have been conducted based on a questionnaire consisting of three major sections: An assessment of the interviewee´s general point of view on standardization, a stepwise presentation and evaluation of the individual dynamic hypotheses (see Sec. 4.2), and an evaluation of the modeling notation.

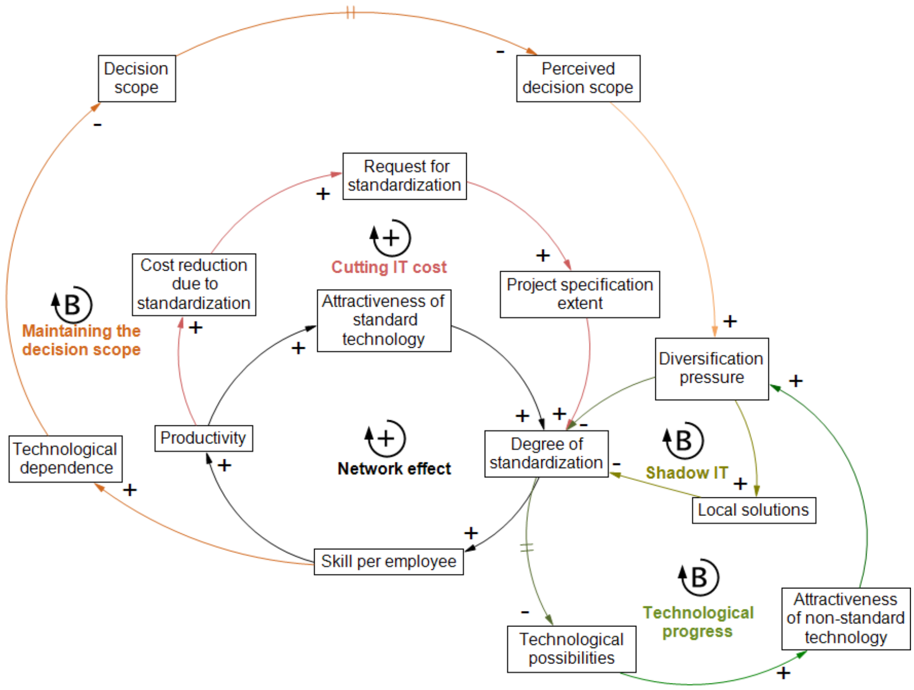

Fig. 3. Integrated causal loop diagram for technological standardization

We conducted the interviews face-to-face or via video conference tools and asked the interviewees to focus on their own experience while trying to identify individual examples for explanations. Due to the broad topic under investigation, the last part of the interviews was done in conversation-style [59] to get a deeper understanding of difficult aspects.

Id	Role	Industry	Experience
1	Enterprise architecture consultant	Insurance	3 years
2	Project manager (business)	Automotive	>10 years
3	Project manager (IT)	Gas Industry	6 years
4	Project manager (business)	Automotive	3 years
5	Head of Sales & Marketing Analytics	Pharma	>10 years
6	Enterprise architect	Automotive	2 years
7	IT revision	Service industry	1 year
8	Solution architect	Automotive	>10 years

Table 1. Overview about interviewed experts from industry

5.2 Evaluation Results

The structured part of the interviews revealed the following insights:

- All interviewees demonstrated their understanding of the used notation. They were able to answer questions about dynamic hypotheses, positive and negative impacts as well as transitive impacts correctly. Furthermore, all of them were able to transfer the presented hypotheses to their individual experiences. However, in all cases a verbal explanation was necessary to guide the interviewee through the presented model. In some cases interviewees did not consider time delay symbols correctly.
- The interviewees by themselves speculated about the role mostly concerned about the presented phenomena. Every expert used his personal experience and role understanding for interpreting the presented models.
- It was striking that all interviewees agreed with every single CLD. For some cases, e.g. productivity increase due to standardization, additional effects were mentioned, such as the fact that employees can get retired and educating the new employees reduces productivity despite a high degree of standardization. In addition, interviewees stated that the impact of the different phenomena is not equally high. The influence factors they considered to have more impact strongly correlated with their role, i.e. their mental model of the system. The same could be observed for the time needed to understand a specific model.
- In two cases, a change of the interviewee's mental model could be observed. When asked for effects related to standardization before presenting the CLDs, they mentioned one or two. When asked the same question after presenting the models both stated that they would consider the presented additional effects as well without having to consult the models again. In both cases, the work experience was lower than five years.
- When confronted with the integrated model (see Fig. 3) most of the interviewees were overwhelmed at first glance. After associating the already presented model parts with the integrated model, all of them understood the latter as well. As benefits of the integrated model, getting a general overview was mentioned as well as a consolidated view on a single variable displaying all identified impacts and effects.
- When asked for potential applications of the presented model, all interviewees mentioned the largest benefits when communicating within diverse groups. Nevertheless, three experts limited potential target groups by excluding top managers.

Based on the insights gained from the presented interviews, the following design guidelines have been derived extending the set of literature-based guidelines.

Guideline 6: Explicitly Include Roles. When using SD models, the respective roles focusing on the specific parts, i.e. feedback loops, should be made explicit to illustrate role conflicts and increase transparency. Therefore, we recommend an additional layer including color coding or icons.

Guideline 7: Use Consistent Terminology. As pointed out by four interviewees, when used as a means of communication among different language communities, ensuring beforehand the equal understanding of the terms used in the model is necessary. To achieve this, a glossary can be used.

Guideline 8: Limit the Number of Modelling Elements for Presentation. Although modeling elements like the delay symbol are necessary to indicate concrete system behavior and foster formal modeling, some of them should be excluded from model visualizations if used for communication. This includes the delay symbol as well as symbols indicating loop direction and type. As could be observed during the interviews, some people struggle with these elements when trying to understand the model. Therefore, a configurable visualization should be used instead of presenting a complete or formal model.

6 Discussion and Outlook

6.1 Summary and Critical Reflection

The paper at hand motivated the use of System Dynamics (SD) models to establish architectural thinking within an enterprise in addition to structural representations commonly created by enterprise architecture teams. Using SD models can create shared understanding among all application landscape designers and therefore enhance decision making processes by establishing trust and coherence. Based on existing literature about SD model creation in general and in particular within the enterprise context, five design guidelines for a method enabling enterprise architects to develop their own SD models have been derived. In order to exemplify the approach and evaluate the guidelines in practice a concrete causal loop diagram (CLD) representing common phenomena in the context of technology standardization has been developed based on hypotheses derived from respective literature. Based on this CLD, eight interviews with industry experts have been conducted to evaluate the model itself as well as CLDs as means of communication. The interview results indicate that CLDs add a new perspective to a controversially discussed topic and might be able to facilitate communication among people with heterogeneous background. While literature offers various guidelines for drawing CLDs [60], we found that not all of them, e.g. an explicit emphasis on delay modeling, are suitable for the EA context when using CLDs as means of communication. We expect that scientist build upon these guidelines to develop an appropriate method for CLD development in the context of enterprise architecture management. In addition, practitioners can evaluate these guidelines in their specific context and use them, for example, to derive requirements for tool support.

6.2 Critical Reflection and Future Research

Although the design guidelines are underpinned by findings presented in related literature, the practical evaluation of the CLD created according to these guidelines is limited by the number of interviews. Furthermore, the phenomena modeled might not completely cover all system dynamics influencing the degree of technology standardization, but within eight interviews no other important and recurring phenomenon could be identified.

Beside a more extensive evaluation and case studies, future research activities should focus on developing a concrete method for SD modeling to facilitate architectural thinking within enterprises. Such method would allow researchers and practitioners to develop more causal loop as well as stock-and-flow diagrams and perform respective evaluations in more detail. Furthermore, researchers should analyze to what extend phenomena modeled by causal loop diagrams are common or company-specific. The identification of archetypes [48] might lead to new insights relevant for decision makers as well as the scientific community. In addition, tool support implementing the presented guidelines is necessary to facilitate evaluation and diffusion in practice.

References

1. Ross, J.W.: Creating a Strategic IT Architecture Competency: Learning in Stages. MIS Quarterly Executive 2, 31–43 (2003)
2. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. Journal of Enterprise Architecture (2007)
3. Greefhorst, D., Proper, H.A.: Architecture Principles. Springer (2011)
4. The Open Group: TOGAF Version 9.1, http://pubs.opengroup.org/architecture/togaf9-doc/arch/ (2011)
5. Schneider, M., Somers, M.: Organizations as Complex Adaptive Systems: Implications of Complexity Theory for Leadership Research. The Leadership Quarterly 17, 351–365 (2006)
6. Morel, B., Ramanujam, R.: Through the looking glass of complexity: The dynamics of organizations as adaptive and evolving systems. Organization Science 10, 278–293 (1999)
7. Brown, S.L., Eisenhardt, K.M.: The Art of Continuous Change: Linking Complexity Theory and Time-Paced Evolution in Relentlessly Shifting Organizations. Administrative science quarterly 42, 1–34 (1997)
8. Johnson, N.F.: Two's Company, Three is Complexity. A simple guide to the science of all sciences. Oneworld, Oxford (2007)
9. Goldstein, J.: Emergence as a construct: History and issues. Emergence 1, 49–72 (1999)
10. Poincaré, H.: Science and method. Dover Publications, Mineola, N.Y (2003)
11. David, P.A.: Clio and the Economics of QWERTY. The American economic review, 332–337 (1985)
12. Mitchell, M.: Complexity. A guided tour. Oxford University Press, Oxford, New York (2011)

13. Weiss, S., Aier, S., Winter, R.: Institutionalization and the Effectiveness of Enterprise Architecture Management. In: Baskerville, R., Chau, M. (eds.) Proceedings of the 34th International Conference on Information Systems (2013)
14. Richardson, G.L., Jackson, B.M., Dickson, G.W.: A principles-based enterprise architecture: Lessons from Texaco and Star Enterprise. MIS Quarterly, 385–403 (1990)
15. Kaisler, S.H., Armour, F., Valivullah, M.: Enterprise Architecting: Critical Problems. In: 38th Annual Hawaii International Conference on System Sciences (2005)
16. Golnam, A., van Ackere, A., Wegmann, A.: Integrating system dynamics and enterprise modeling to address dynamic and structural complexities of choice situations. In: Moon, T.-H. (ed.) Proceedings of The 28th International Conference of The System Dynamics Society, vol. EPFL-CONF-153207vol. . Wiley (2010)
17. Rashid, S., Masood, T., Weston, R.H.: Unified modelling in support of organization design and change. Journal of Engineering Manufacture 223, 1055–1079 (2009)
18. Molnar, W.A., Korhonen, J.J.: Research paradigms and topics in Enterprise Engineering analysis of recent conferences and workshops. Eighth International Conference on Research Challenges in Information Science (RCIS) (2014)
19. Forrester, J.W.: Industrial dynamics. MIT press Cambridge, MA (1961)
20. Sterman, J.: Business dynamics. Systems thinking and modeling for a complex world. Irwin/McGraw-Hill, Boston (2000)
21. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28, 75–105 (2004)
22. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of management information systems 24, 45–77 (2007)
23. Sterman, J.D.: Learning in and about complex systems. System Dynamics Review 10, 291–330 (1994)
24. Homer, J., Oliva, R.: Maps and models in system dynamics: a response to Coyle. System Dynamics Review 17, 347–355 (2001)
25. Agyapong-Kodua, K., Weston, R.H., Ratchev, S.: The integrated use of enterprise and system dynamics modelling techniques in support of business decisions. Advances in Decision Sciences 2012 (2012)
26. Luna-Reyes, L.F., Andersen, D.L.: Collecting and analyzing qualitative data for system dynamics: methods and models. System Dynamics Review 19, 271–296 (2003)
27. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26, 13–23 (2002)
28. Assimakopoulos, N.A., Riggas, A.N.: Designing a virtual enterprise architecture using structured system dynamics. Human Systems Management 25, 13–29 (2006)
29. Sunkle, S., Roychoudhury, S., Kulkarni, V.: Using Intentional and System Dynamics Modeling to Address WHYs in Enterprise Architecture. Proceedings of the International Conference on Software Engineering and Applications (2013)
30. Wojcik, L.A., Hoffman, K.C.: Systems of Systems Engineering in the Enterprise Context: A Unifying Framework for Dynamics. In: 2006 IEEE/SMC International Conference on System of Systems Engineering, pp. 22–29
31. Buckl, S., Matthes, F., Schneider, A.W., Schweda, C.M.: Pattern-based Design Research– An Iterative Research Method Balancing Rigor and Relevance. In: Vom Brocke, J., Hek-

kala, R., Ram, S., Rossi, M. (eds.) Design Science at the Intersection of Physical and Virtual Design, pp. 73–87. Springer, Berlin, Heidelberg (2013)

32. Rittel, H.W.J., Webber, M.M.: Dilemmas in a general theory of planning. Policy sciences 4, 155–169 (1973)
33. Harrell, J. Michael, Sage, A.P.: Enterprise architecture and the ways of wickedness. Information, Knowledge, Systems Management 9, 197–209 (2010)
34. Innes, J.E., Booher, D.E.: Consensus building and complex adaptive systems: A framework for evaluating collaborative planning. Journal of the American Planning Association 65, 412–423 (1999)
35. Innes, J.E.: Information in communicative planning. Journal of the American Planning Association 64, 52–63 (1998)
36. Maynard, M.T., Gilson, L.L.: The Role of Shared Mental Model Development in Understanding Virtual Team Effectiveness. Group & Organization Management 39, 3–32 (2014)
37. Espinosa, J.A., Armour, F., Boh, W.F.: The role of group cognition in enterprise architecting. 44th Hawaii International Conference on System Sciences (2011)
38. Dörner, D.: The logic of failure. Philosophical transactions of the Royal Society of London. Series B, Biological sciences 327, 463–473 (1990)
39. Jonassen, D.H.: Instructional design models for well-structured and III-structured problem-solving learning outcomes. Educational Technology Research and Development 45, 65–94 (1997)
40. Isaksen, S.G., Treffinger, D.J.: Creative problem solving. The Basic Course. New York: Bearly Limited (1985)
41. Surowiecki, J.: The wisdom of crowds. Random House LLC (2005)
42. Stachowiak, H.: Allgemeine Modelltheorie. Springer-Verlag, Wien, New York (1973)
43. Sterman, J.D.: All models are wrong: reflections on becoming a systems scientist. System Dynamics Review 18, 501–531 (2002)
44. Farwick, M., Agreiter, B., Breu, R., Ryll, S., Voges, K., Hanschke, I.: Requirements for automated enterprise architecture model maintenance. In: Zhang, R., Cordeiro, J., Li, X., Zhang, Z., Zhang, J. (eds.) Proceedings of the 13th international conference on enterprise information systems, pp. 325–337. SciTePress (2011)
45. Sweller, J.: Cognitive load theory, learning difficulty, and instructional design. Learning and instruction 4, 295–312 (1994)
46. Niemi, E., Pekkola, S.: Enterprise Architecture Quality Attributes: A Case Study. In: 46th Hawaii International Conference on System Sciences (HICSS), pp. 3878–3887 (2013)
47. Wegmann, A.: The Systemic Enterprise Architecture Methodology (SEAM). Business and IT Alignment for Competitiveness. Lausanne, Switzerland (2002)
48. Senge, P.: The fifth discipline: The art and science of the learning organization. New York: Currency Doubleday (1990)
49. Checkland, P.: From optimizing to learning: A development of systems thinking for the 1990s. Journal of the Operational Research Society, 757–767 (1985)
50. Sein, M., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research (2011)
51. Page, S.E.: Diversity and complexity. Princeton University Press, Princeton, N.J (2011)
52. Sharpe, W.F.: Microcomputer Perspectives: Economies of Scale, Technological Progress and Standardization. Financial Analysts Journal, 25–27 (1983)

53. Baron, J., Schmidt, J.: Technological Standardization, Endogenous Productivity and Transitory Dynamics (2013)
54. Richen, A., Steinhorst, A.: Standardization or harmonization? you need both. European Health Informatics (November 2005), 5 (2005)
55. Barker, S., Fiedler, B.: Developers, Decision Makers, Strategists or Just End-users? Redefining End-User Computing for the 21st Century: A Case Study. Journal of Organizational and End User Computing (JOEUC) 23, 1–14 (2011)
56. Györy, A., Cleven, A., Uebernickel, F., Brenner, W.: Exploring the shadows: IT governance approaches to user-driven innovation. Proceedings of the 20th European Conference on Information Systems, Barcelona, Spain (2012)
57. DIN Deutsches Institut für Normung: Economic benefits of standardization. Summary of results: final report and practical examples. Beuth, Berlin, Köln (2000)
58. Farrell, J., Klemperer, P.: Coordination and lock-in: Competition with switching costs and network effects. Handbook of industrial organization 3, 1967–2072 (2007)
59. Goguen, J.A., Linde, C.: Techniques for requirements elicitation. Proc. RE'93 - First IEEE Symposium on Requirements Engineering, 152–164 (1993)
60. Kim, D.H.: Guidelines for drawing causal loop diagrams. The Systems Thinker 3, 5–6 (1992)