Association for Information Systems
AIS Electronic Library (AISeL)

All Sprouts Content Sprouts

9-16-2011

Understanding Organizational Traps in Implementing Service-Oriented Architecture

Xitong Li
MIT Sloan School of Management, xitongli@mit.edu

Stuart Madnick
MIT Sloan School of Management, smadnick@mit.edu

芽|Sprouts Working Papers on Information Systems ISSN 1535-6078

Understanding Organizational Traps in Implementing Service-Oriented Architecture

Xitong Li
MIT Sloan School of Management, USA

Stuart Madnick
MIT Sloan School of Management, USA

Abstract
One of the major objectives of adopting service-oriented architecture (SOA) is to enhance the IS agility of organizations and improve IT-business alignment. In practice the contradictory experiences about SOA implementation turn out to be a paradox: why many organizations failed to meet their expectations about SOA implementation efforts, while others succeeded? Contrast to prior research on SOA, this study adopts the process perspective and provides plausible theoretical explanations for the "SOA implementation paradox". Specifically, the study uses multiple case-study methods to develop a system dynamics model which highlights the feedback loops and time delay during the SOA implementation process. The results reveal the dynamic characteristics of learning curve of SOA implementation and two organizational traps (technology learning trap and implementation effectiveness trap) associated with SOA implementation. Technology learning trap refers to the situation that the less learning in using the technology, the more difficult and complex the technology is perceived. Implementation effectiveness trap refers to the situation in which the organization may misperceive the inappropriateness of SOA when SOA implementation is temporally less effective and perceived benefits of SOA are delayed. The theory of the organizational traps can be generalized to a broad context of innovative IS implementation. Further, the theoretical causes of the traps are investigated. Finally, the research implication of this study and connections with existing literature on IS and organization are discussed.

Keywords: Service-oriented Architecture (SOA), IS implementation, organizational traps, organizational change and learning, system dynamics

Permanent URL: http://sprouts.aisnet.org/11-139

Reference: Li, X., Madnick, S. (2011). "Understanding Organizational Traps in Implementing Service-Oriented Architecture," Proceedings > Proceedings of JAIS Theory Development Workshop . *Sprouts: Working Papers on Information Systems*, 11(139). http://sprouts.aisnet.org/11-139

Understanding Organizational Traps in Implementing Service-Oriented Architecture

ABSTRACT

One of the major objectives of adopting service-oriented architecture (SOA) is to enhance the IS agility of organizations and improve IT-business alignment. In practice the contradictory experiences about SOA implementation turn out to be a paradox: why many organizations failed to meet their expectations about SOA implementation efforts, while others succeeded? Contrast to prior research on SOA, this study adopts the process perspective and provides plausible theoretical explanations for the "SOA implementation paradox". Specifically, the study uses multiple case-study methods to develop a system dynamics model which highlights the feedback loops and time delay during the SOA implementation process. The results reveal the dynamic characteristics of learning curve of SOA implementation and two organizational traps (technology learning trap and implementation effectiveness trap) associated with SOA implementation. Technology learning trap refers to the situation in which a certain technology is less understood due to insufficient learning, the more difficult and complex the technology is perceived. Implementation effectiveness trap refers to the situation in which SOA is misperceived to be inappropriate when SOA implementation is temporally less effective and the perceived benefits of SOA are delayed. The theory of the organizational traps can be generalized to a broad context of innovative IS implementation. Further, the theoretical causes of the traps are investigated. Finally, the research implication of this study and connections with existing literature on IS and organizational study are discussed.

Keywords: Service-oriented Architecture (SOA), IS implementation, organizational traps, organizational change and learning, system dynamics

Preliminary version – Please do NOT cite, quote or distribute without the authors' permission.

Comments are very welcome and can be sent to Xitong Li (xitongli@mit.edu).

Understanding Organizational Traps in Implementing Service-Oriented Architecture

INTRODUCTION

In an increasing fast-changing environment, it is important for an organization to be able to adapt its IT systems and quickly respond to changing business conditions. Such an ability is defined as the organization's information systems (IS) agility[1] (Choi et al. 2010) and has been considered as a key facilitator to enhance dynamic capabilities (Sher et al. 2004) and competitive advantages (Luthria et al. 2009b; Winter 2003). However, the traditional IT systems, e.g., Enterprise Resource Planning (ERP) systems, are usually designed using a monolithic architecture and built as integrated sets of software modules linked to a common database, handling corporate functions like finance, human resources, material management, and sales (Robey et al. 2002). It is thus quite difficult, expensive and time-consuming for organizations to make changes to their IT systems designed using the monolithic architecture (Choi et al. 2010). To address such inappropriateness, service-oriented architecture (SOA) has been advocated as a new computing paradigm to build IT systems in organizations (Papazoglou et al. 2003). In this paper, SOA refers to "the architecture style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner" (Borges et al. 2004) and consists of a composite set of business-aligned services that support a flexible and dynamically reconfigurable end-to-end business process realization using interface-based service descriptions.

One of the major objectives of adopting SOA is to enhance the IS agility of organizations and improve IT-business alignment (Bieberstein et al. 2005; Choi et al. 2010; Mueller et al. 2010). Many organizations with the expectation to reap those potential benefits have invested in

---

[1] Information System (IS) agility is defined by Choi et al. (2010) as "the ability to quickly make changes to IT applications in response to changing business conditions".

SOA. A recent Forrester report reveals that 71% of the enterprises surveyed are already using SOA or will be by the end of 2011 (Kanaracus 2011). However, mixed outcomes about SOA adoption and implementation have been often reported. For example, a 2007 InformationWeek Web survey of 278 IT professionals found that 32% of those using SOA said those projects fell short of expectations and "of those, 58% said their SOA projects introduced more complexity into their IT environments, and 30% said they cost more than expected. Out of all respondents using SOAs, just 10% said the results exceeded expectations" (Smith 2008). In stark contrast, CA Technologies recently released a survey which covered 615 companies in the process of SOA-based efforts and found that 92% of their SOA initiatives met or exceeded business unit objectives (McKendrick 2011). The contradictory outcomes turn out to be a paradox of SOA adoption and implementation. The key puzzle regarding the "SOA implementation paradox" is why many organizations failed to meet their expectations about their efforts on SOA implementation, while others succeeded. Given the growing significance and risk of SOA, it is surprising that there is a scarcity of academic research addressing the SOA implementation paradox (Luthria et al. 2009a).

Mohr (1982) identifies two different perspectives of organizational research: variance and process perspectives (Mohr 1982). According to Mohr's distinction, variance research seeks to explain variation in outcome variables by associating them with predictor variables and necessary and sufficient conditions. By contrast, process research seeks to explain organizational phenomena occurring by examining sequences of events over time (Robey et al. 2002). To date, prior academic research on SOA has largely adopted the variance perspective. Specifically, prior research on SOA has intensively claimed potential benefits and business value of SOA (Cherbakov et al. 2005; Mueller et al. 2010; Varadan et al. 2008), but there are only a few empirical works actually measuring the benefits of specific SOA implementation (Baskerville et al. 2005; Moitra et al. 2005). Despite the potential benefits claimed, organizations implementing SOA often encounter challenges in their efforts. To address those challenges, more recent, yet relatively even few, literature turns to explore critical success factor (CSF) that potentially affect SOA implementation. Research in this strand tends to enumerate a number of factors that potentially facilitate or impede organizations to receive the intended benefits of SOA. Prior research, however, failed to explore the causal relationships and, particularly, complex interactions between those critical factors. After all, prior research on SOA largely from the variance perspective posits an invariant relationship between antecedents and outcomes, which is too stringent to explain organizational phenomena (Markus et al. 1988). Besides that, after a extensively literature review, Luthri et al. (2009a) found "there is little or no realistic data available on what, if anything, firms are doing in practice to address the inherent challenges of implementing a service-oriented architecture…" (Luthria et al. 2009a). There is a clear need for research to provide qualitative data from case studies on SOA implementation. In sum, to the best of our knowledge, there has been neither theoretical formulation nor empirical examination of SOA implementation in the literature. This paper seeks to fill the gap at those aspects.

Noticing the widely-reported contradictory experiences with SOA implementation, the research presented in this paper suggests that we may be able to make some sense of the apparently inconsistent outcomes by shifting the focus away from the variance perspective to the process perspective. Unlike the variance research, the process studies "have lower aspirations about 'explained variance', but provide richer explanations of how and why the outcomes occur [and] when they do occur" (Markus et al. 1988). As demonstrated in this paper, the process perspective is appropriate to specify temporal relations among theoretical elements and study the micro-processes of capability-enhancing organizational change (Repenning et al. 2002) and thus allow us to explain different experiences of the SOA implementation paradox.

Using process perspective, this paper interprets the implementation process of SOA as a specific case of IT-based improvement process of an organization's IS agility and conceptualizes it as a form of organizational change. The core research question is: how various critical elements interact with each other during the SOA implementation process and result in the organizational traps[2] associated with the process?

In answering this question, this paper develops a theory through an inductive research strategy of process-focused improvement which was successfully used in Repenning's works on manufacturing process improvement (Repenning et al. 2001; Repenning et al. 2002). Note that our research does not simply apply the existing organizational theory of capability traps (Repenning et al. 2001; Repenning et al. 2002) to understanding the challenges in SOA implementation. In fact, based on our observation and investigation, SOA implementation has at least two inherent characteristics distinct from process improvement in manufacturing (see the discussion in the next section). Accordingly, the model and theory developed in this paper suggest that two different but intertwined organizational traps, technology learning trap and implementation effectiveness trap, play important roles in the difficulty of many SOA implementation efforts. As will be explained in detail, technology learning trap refers to the situation in which a certain technology is less understood due to insufficient learning, the more difficult and complex the technology is perceived. Implementation effectiveness trap refers to the situation in which SOA is misperceived to be inappropriate when SOA implementation is temporally less effective and the perceived benefits of SOA are delayed. Both organizational

---

[2] Organizational traps in this paper specifically refer to technology learning trap and implementation effectiveness trap, both of which will be introduced in the next paragraph.

traps are distinct from the existing theory of capability traps and have not been discussed in the existing literature.

This research makes two important contributions to the literature on information systems and organizational studies. First, the theoretical contribution provides the theory of organizational traps in SOA implementation from the process perspective. The organizational traps constitute a theoretical explanation for the contradictory outcomes of the implementation of innovative technology in general and SOA in particular. The theory offers fruitful insights for understanding the implementation and use of SOA in organizations. Second, the empirical contribution provides meaningful qualitative data about how organizations have implemented SOA in practice. The empirical results reveal that the inherent tradeoffs of SOA implementation between short-term performance drop and potential long-term benefits and between local project needs and organization-level IS agility. Human agency of organizational actors plays an important role in the tradeoffs and may facilitate or impede SOA implementation, depending on different organizational contexts.

THEORETICAL BACKGROUND

We organize the relevant academic research into two parts. The first part is the prior studies on SOA, most (if not all) of which have adopted the variance research perspective. Two particular strands can be distinguished: research on potential benefits of SOA and research on critical success factors of SOA implementation[3]. The second part of relevant literature is Repenning's works on capability traps in manufacturing process improvement (Repenning et al. 2001; Repenning et al. 2002) and innovation implementation (Repenning 2002).

[3] Although there are extensive prior studies on the success and challenges of ERP implementation, research focusing on the challenges of SOA implementation is very limited. This is one of the unique motivations of this paper.

Variance Research on SOA

Potential Benefits of SOA

The potential benefits and business value of SOA have been numerously claimed in the literature (Mueller et al. 2010; Varadan et al. 2008). An early work (Yoon et al. 2007) has analyzed multiple cases and suggested that the realized benefits of SOA can be classified into two groups: improved business agility and cost reduction. The benefits contributing to improved business agility include easier integration of components and systems (Cherbakov et al. 2005), better IT-business alignment (Bieberstein et al. 2005), and a quicker response to market change or customer demand (Choi et al. 2010). The benefits of cost reduction consist of lower application development costs/time, reuse of existing components/services (Fricko 2006), and lower maintenance costs. The recent work (Mueller et al. 2010) develops a comprehensive conceptual framework to understanding the economic potential of SOA. According to their work, SOA relies on three fundamental design principles: modularity, loose coupling, and standards. They built on the resource-based view and argued that SOA can enhance an organization's IS capabilities which in our research is conceptualized as IS agility (Choi et al. 2010). By enhancing IS capabilities, SOA is expected to provide multi-dimensional benefits for organizations, including IT infrastructure, operational, strategic, managerial and organizational benefits. At the individual level, IT developers' productivity is also believed to be enhanced by improved IS reusability and interoperability from SOA design principles (Choi et al. 2010; Hau et al. 2008; Mueller et al. 2010).

Despite the numerous benefits claimed in the literature, organizations adopting SOA cannot receive those intended benefits automatically. This strand of literature generally adopts technological determinism and omits processual elements and the actions of key players associated with SOA implementation and organizational change.

Critical Success Factors of SOA Implementation

More recent (yet relatively few) literature turns to explore critical success factors (CSF) and challenges that potentially affect SOA implementation. The work (Luthria et al. 2008) explores the organizational constraints and challenges experienced by firms considering the enterprise-wide SOA implementation. They analyzed several case studies and propose a set of seven best practices for successful enterprise-level SOA implementation. The top three are: 1) get commitment at the broad level; 2) manage expectations and invest in SOA for the long term; and 3) align the entire organization along the SOA strategy. The work (Luthria et al. 2009a) presents six factors that influence the organizational adoption of SOA, among which the perceived value to the organization is ranked as the most important factor. The work (Boh et al. 2010) examines two organizational factors that are potentially critical in ensuring the success of SOA implementation: top management support and the centralization of IT decision-making. Their empirical results from hypothesis testing indicate that top management support is a significant factor, yet centralization of IT decision-making is not. The recent work (Lee et al. 2010) conducts a more comprehensive research and identifies 20 factors in SOA implementation based on their review of 34 SOA studies and 22 interviews with both vendors and users. Their results show that "building strong support for enterprise-wide core human resources" and "clear goal-setting based on business value" are often ranked among top 3 by all the empirical data (literature review, interview with vendors and users).

The research strand on CSF tends to list a number of factors that potentially affect SOA implementation and facilitate or impede organizations to receive the intended benefits of SOA.

Like the research on potential benefits of SOA, the research on CSF omits processual elements and the actions of key players associated with SOA implementation. Also, prior research on SOA fails to explore the causal relationships among those critical factors. More importantly, none of them investigates how those critical factors interact with each other during the dynamics of the multi-month/year process of SOA implementation. Therefore, prior research on SOA offers little in the way of theoretical explanations for the puzzling SOA implementation paradox.

The abovementioned literature repeatedly points out that top management commitment and support are very critical to the success of SOA implementation. The commitment is required to be long-term and enterprise-wide, rather than short-term or local-focused. Besides that, the perceived benefits and business value from SOA are also very important to the implementation. In fact, management commitment and perceived benefits from SOA are strongly dependent; they interact with each other during the process of SOA implementation. The model and theory in this paper capture this important point.

Capability Traps in Process Improvement

Another relevant literature is Repenning's works on capability traps in process improvement (Repenning et al. 2001; Repenning et al. 2002) and innovation implementation (Repenning 2002). His research focuses on Total Quality Management (TQM) initiatives in manufacturing and develops causal-loop diagrams and system dynamic models to understand the impact of time delays between investing in process improvement and realizing the benefits. He argues that the long delays in the feedback loops of process improvement create the dynamics of the "worse-before-better" pattern and cause capability traps and self-confirming errors. Specifically, considering the "worse-before-better" pattern, workers initially tend to underinvest in process improvement and often find themselves falling short of meeting the performance target due to insufficient process capability. Thus, workers are forced to further shift time from process improvement and increase work hours. Accordingly, the dynamics of process improvement work as vicious cycles and workers are trapped in a downward spiral of eroding process capability, forcing less and less time for improvement. Eventually, the capability traps resulted in the failures of many process improvement efforts.

The phenomenon of beneficial improvement and innovations that go unused have been documented not only in TQM but other administrative initiatives, such as human resource practices (Pfeffer et al. 2000) and best practices for product development (Wheelwright et al. 1995). The inability of many organizations to use the knowledge embodied in the improvement initiatives is a central issue facing organizational theorists (Pfeffer 1997). Although Repenning's work has been successfully used to explain process improvement in manufacturing, it is unknown whether or not the similar phenomenon had happened during IS improvement efforts in general and SOA implementation in particular. To date, none of IS literature has provided theoretical explanation for the contradictory outcomes of SOA implementation from the perspective of improvement process. As one of the major contributions, this research finds two organizational traps (technology learning trap and implementation effectiveness trap), which result from the interaction between human judgmental biases and the physical structure of IS development processes, may occur in the failures of many SOA implementation efforts.

It is worth noting that this research does not simply apply the theory of capability traps to understanding the challenges in SOA implementation. Our observation and investigation reveal that SOA implementation has two inherent characteristics that are distinct from TQM and process improvement programs in manufacturing:

1) SOA implementation often requires enterprise-wide involvement and commitment, while process improvement programs often focus on certain work process (e.g., manufacturing, product development). It is more challenging to call for and maintain enterprise-wide, long-term involvement and commitment in organizations.
2) In regard to process improvement programs, it is relatively easier to identify defects and correct them when the process capability is low (Repenning et al. 2002). Thus, favorable results and word of mouth are easier to achieve and come earlier from the investment in improvement. Unfortunately, SOA implementation is not such a case. The learning curve of the complex technology like SOA creates longer substantial delay and postpones the potential benefits of SOA at the early stage of SOA implementation. It should be anticipated that developers may perceive little SOA effectiveness when they just start learning how to use IT systems developed by SOA design principles. Overcoming the learning-curve barrier is critical to achieve perceived benefits from SOA.

RESEARCH METHODOLOGY

The research methodology used in this study is system dynamics modeling (Sterman 2000), with an aim of generating an explanatory theory of organizational traps associated with SOA implementation. This methodology has been shown to be a powerful modeling tool for organizational theory building (Repenning et al. 2002; Rudolph et al. 2009; Sastry 1997; Sterman 2000). Since Abdel-Hamid and Madnick started using it in investigating software project management in 1980s (Abdel-Hamid et al. 1991), system dynamic modeling has continued to receive increasing attention in IS research (Cao et al. 2010; Choi et al. 2010; Georgantzas et al. 2008; Rahmandad et al. 2009b). This research adopts system dynamics modeling for three primary reasons.

First, system dynamics modeling highlights the processual elements and the actions of key players associated with organizational change which are often omitted in IS studies that rely on the variance perspective and cross-sectional, quantitative data (Markus et al. 1988; Orlikowski 1993). System dynamics modeling provides “a useful context in which to study the micro-processes that impede or facilitate competence-enhancing change” (Repenning et al. 2002) and is compatible with the focus of process perspective of organizational research to “explain outcomes by examining sequences of events over time” (Markus et al. 1988; Robey et al. 2002). As indicated above, process research on SOA is simply missing in the extant literature. Thus, the research approach that specifically includes elements of process and change is particularly appropriate here.

Second, as Weick (1979) notes, “It is the network of causal relationships that impose many of the controls in organizations and that stabilize or disrupt the organization. It is the patterns of these causal links that account for much of what happens in organizations” (Weick 1979). Unfortunately, Law and Urry (2004) pointed out “social science method has problems in understanding non-linearity relationships and flows” (Law et al. 2004). Compared to other organizational research methodologies, system dynamics modeling has its strengths and is particularly useful here, because it allows focusing on the feedback loops and nonlinearity of the change associated with SOA implementation process (Sterman 2000).

Third, system dynamics modeling specifically focuses on the effects of the feedback delay on organizatinal change and learning (Rahmandad et al. 2009a; Sterman 2000). Thus, it allows generating unique insights into the dynamics of implementation and appropriation of SOA, which are less likely to be produced using other organizational methods.

In sum, system dynamics modeling is particularly appropriate in this research, as it provides "an opportunity to examine continuous processes in context… and thereby reveal the multiple sources of loops of causation and connectivity so crucial to identifying and explaining patterns in the process of change" (Pettigrew 1989).

Sample Selection

We employed multiple comparative case study design in this research. Case study is justified as a research strategy that "attempts to examine: (a) a contemporary phenomenon in its real-life context, especially when (b) the boundaries between phenomenon and context are not clearly evident" (Yin 1981). Although case studies often report phenomena observed in a single case (Orlikowski 1992) or two comparative cases (Orlikowski 1993; Repenning et al. 2002), multiple case studies have been effectively used in IS research (Abdel-Hamid et al. 1991; Bandara et al. 2005; Robey et al. 2002). Multiple comparative case studies were conducted here for two reasons. First, Robey et al. (2002) note that "Although some richness of detail may be sacrificed with additional cases, the ability to compare phenomena across different contexts in enhanced" (Robey et al. 2002). By using the case cluster method (McClintock et al. 1979), the ability of multiple case comparisons can be even enhanced. Second, multiple case study design is compatible with our research interest in generating a theory that is applicable to a general organizational context rather than that for a specific one. The research sample consists of 10 North American organizations that have recently implemented or been implementing SOA, including EMC, Raytheon, Oracle, SAP, MIT Lincoln Lab and several US government agencies. While some of them only implemented SOA in their own organizations like US government agencies, most of others, like EMC, Raytheon, Oracle, SAP and MIT Lincoln Lab, have implemented SOA not only in their own organizations but for other organizations as contractors.

Data Collection

In this research, we followed the standard method of data collection that was used in Abdel-Hamid and Madnick's works on software project management (Abdel-Hamid et al. 1991). Sterman (2000) also suggested a similar method of data collection and analysis which mainly uses semistructured interviews and literature review. Specifically, we took three steps as follows.

First, the initial series of 10 unstructured interviews were conducted with 14 different managers and developers. Interviews lasted between 60 and 90 minutes. After the purpose of the research was explained, informants began by describing their professional background. They were then asked to share their experience associated with SOA implementation efforts in the organization. Finally, they were asked to speculate the potential key challenges and elements that facilitated or impeded those SOA implementation efforts. In particular, we focused on collecting data that reveals the dependencies between those elements and the interactions of the elements occurring over time. To build more confidence in our findings and to come up with "holistic and multifaceted explanations of change" (Pettigrew 1990), numerous follow-up conversations on telephone and via email, intensive documentation review, and direct on-site observation were also conducted to supplement the data collection. The various techniques of data collection is particularly beneficial in theory generation because "it provides multiple perspectives on an issue, supplies more information on emerging concepts, allows for cross-checking, and yields stronger substantiation of constructs" (Orlikowski 1993).

Second, an extensive review of the literature was conducted, after a "skeleton" model was built based on the information collected at the first phase. The "skeleton" model served as a useful guideline for the literature review. As Forrester (1968) suggested in his classic book on system dynamics, "A model should come first. And one of the first uses of the model should be to determine what formal data need to be collected" (Forrester 1968). The literature review provides guidance in techniques for qualitative data collection and analysis and filled in many gaps of the "skeleton" model, leading to a more detailed version of the model. Also, each causal link in the model was justified by the supportive evidence from the literature review.

Third, another series of 12 interviews were conducted with 15 informants. With the concepts and model generated from the prior two stages, the interviews of this phase became more structured than that in the first phase. Likewise, various techniques of data collection were used such as numerous follow-up conversations, intensive documentation review and on-site observation. As is typical with interpretive research based on qualitative data (Locke et al. 2001; Pettigrew 1990), we triangulated across the multiple data sources and proceeded iteratively between the data gathered and the model.

Data Analysis

The data analysis was begun with the traditional methods for inductive research study. Constructs and patterns of interest were identified from the initial data analysis and categories of the constructs formed based on the coding. Key variables and causal links among them emerged during the analysis. Then, we used the causal loop diagramming method common in system dynamics (Repenning et al. 2002; Sterman 2000) to develop the "skeleton" model. While developing the model, we emphasized the essential feedback loops that would generate the patterns of SOA implementation emerged from the coded data. These feedback loops allow the emerging model and theory to focus on the processual elements and exploring the dynamics of the implementation process. Following the similar methods, data analysis after the data collection in the second and third phase provided an opportunity to refine, improve and validate the emerging model.

Note that in this research data analysis and collection actually overlapped and were iteratively conducted. Overlapping data analysis with data collection, as Eisenhardt (1989) notes, "not only gives the researcher a head start in analysis, but more importantly allows researchers to take advantage of flexible data collection". Further, the data collected in a later phase allowed us to refine and improve the model[4] and to "criticize it, expose it again and so on in an iterative process that would continue as long as it proves to be useful" (Abdel-Hamid et al. 1991). As indicated above, we combined various techniques of data collection synergistically and conducted the data analysis overlapping data collection through iterative methods of constant comparison and extensive memo writing (Ryan et al. 2000). In such a way, we were able to produce complete accounts of the organizations that had implemented or were implementing SOA. Thus, the resulting causal-loop model is both tightly grounded in our data and provides a logical and internally consistent explanation for how the micro-level interactions involved in SOA implementation combined to create the more macro-level dynamics of the entire process.

MODEL

In this research, we use system dynamics modeling to develop the causal-loop diagrams for the theory building (Sterman 2000). System dynamics models consist of stocks, flows and causal links between variables. Stocks and flows are used to model physical and/or organizational processes, wherein a stock, denoted by a rectangle, represents the level that can accumulate or deplete over time. Flows denoted by straight arrows with valves cause an increase or decrease in stock levels. Stocks and flows complement feedback loops, representing the physical and/or organizational system's structure. Positive (self-reinforcing) and balancing (self-correcting)

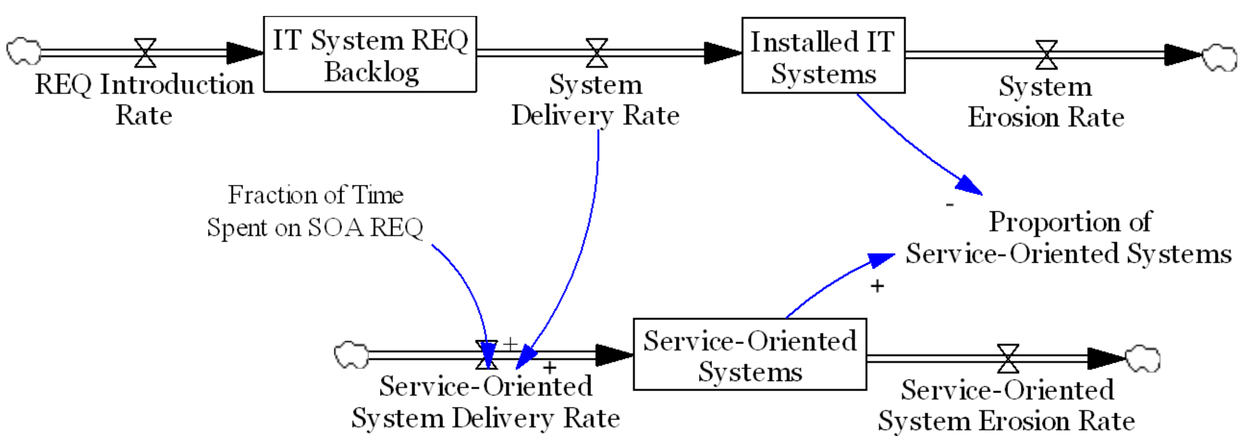
[4] Besides informative interviews, some IT managers/developers even spent 2-4 hours with the authors together and went through every single variable and causal link of the model. Their inputs and feedbacks allowed the authors to produce the final model largely grounded upon empirical experiences of those practitioners. By doing so, the validity of the model was strengthened by being validated directly with the sources.

feedback loops play an important role in determining the dynamic behaviors of organizational systems. The model consisting of causal-loop diagrams captures the key reinforcing and balancing feedback loops.

Note that the causal-loop diagrams in this paper are not intended to provide an accurate mathematical specification of the causal links. Yet by explicitly demonstrating feedbacks with nonlinearity, discontinuities and time delays between causes and effects, causal-loop diagramming enables us to reach a new level of specificity concerning the mutual causality. Thus far this approach has been effectively used in organizational theory development (Perlow et al. 2002; Repenning et al. 2001; Repenning et al. 2002; Rudolph et al. 2009; Sastry 1997). Specifying a formal mathematical model is often the next step in testing the theories embodied in the causal-loop diagrams and is not the focus of this paper. It is also worth noting that in building the model and theory, we focused on the evidences that are generalizable across multiple organizations instead of those merely applied to a specific organization. By doing so, we were able to make the model, although seemingly oversimplified, built in a broad organizational context and produce general theoretical insights and implications.

The first assumption in our model is that only a proportion of delivered IT systems in an organization are service-oriented systems. The rationale of this assumption lies in that IT developers need to spend extra time and energy to follow the SOA design principles (e.g., modularity, loose coupling, and standards) (Mueller et al. 2010) in order to make the delivered systems service-oriented. Otherwise, the delivered systems just turn out to be non-service-oriented and are installed in the organization. The first key variable in the model is Proportion of Service-Oriented Systems, which is the ratio of the number of Service-Oriented Systems to the number of total Installed IT Systems. Basically, Proportion of Service-Oriented Systems is used to capture the penetration of SOA in the organization. In Figure 1, Installed IT Systems is a stock and denoted by a rectangle. Figure 1 shows the stock of IT system requirements backlog accumulates as system requirements are introduced over time. The delivered IT systems are installed with System Delivery Rate. Service-Oriented System Delivery Rate is a fraction of the overall System Delivery Rate. The fraction coefficient depends on the fraction of working hours spent on implementing SOA requirements. The "+" sign at the head of the causal link from System Delivery Rate to Service-Oriented System Delivery Rate means there is a positive causal relationship between the two variables. That is, all other factors are equal, the higher System Delivery Rate, the higher Service-Oriented System Delivery Rate. Any IT systems regardless of service-orientation or not may erode over time due to the change of business environment or need for technology upgrade.

Figure 1: Proportion of Installed IT Systems Are Service-Oriented Systems

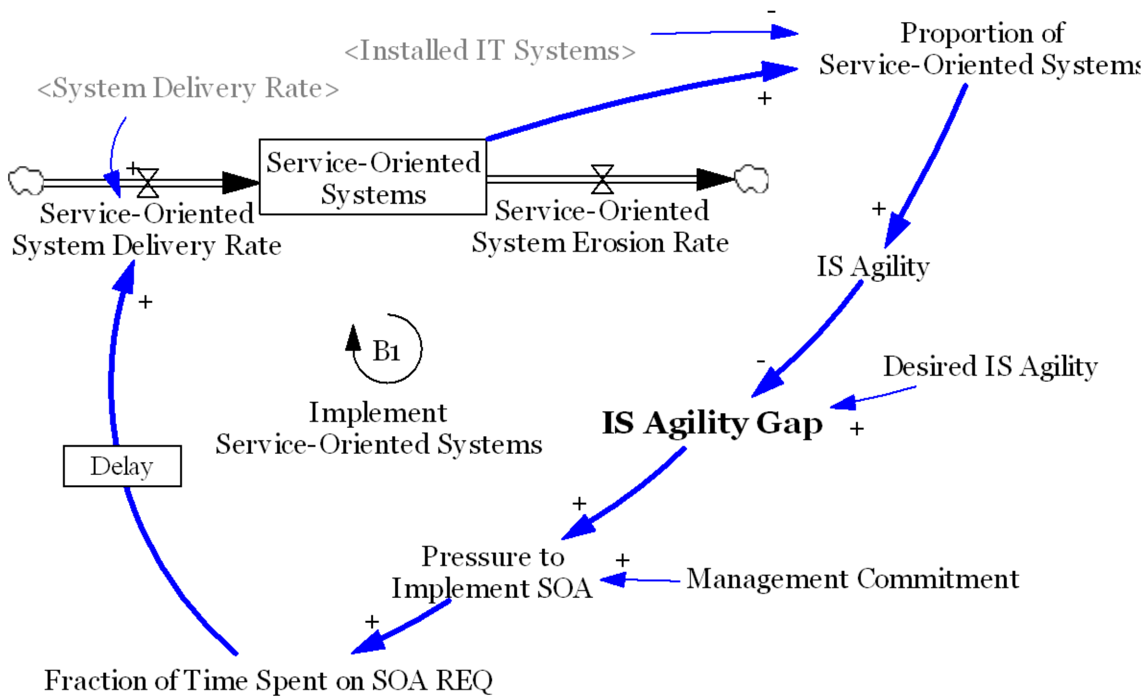


We present the key feedback loops in the rest of this section and then synthesize them in the causal-loop model. Readers may need to keep the entire causal-loop model (see Figure 6) in mind while reading through each of the feedback loops.

Balancing Loop B1

Figure 2 shows the higher Proportion of Service-Oriented Systems, as the conceptualization of the organizational penetration of SOA, enhance the organization's IS agility (Choi et al. 2010). The IS Agility Gap, defined as the difference between the Desired IS Agility and the actual IS Agility, results in the need and Pressure to Implement SOA. From the perspective of most managers, Desired IS Agility is an exogenous demand. Management Commitment is also required along with IS Agility Gap to generate Pressure to Implement SOA, reflecting the fact that management commitment and support is a critical success factor of SOA implementation (Boh et al. 2010; Lee et al. 2010). Under the Pressure to Implement SOA, developers are forced to put a fraction of the work hours spent on implementing SOA requirements. The time spent on SOA requirements represents the developers need to spend extra time to follow the SOA design principles when they develop the IT systems. The more time spent on SOA REQ, the higher Service-Oriented System Delivery Rate. However, the rise of Service-Oriented System Delivery Rate with the increase of Fraction of Time Spent on SOA REQ can only be realized after a substantial time delay, because developers have to learn how to build service-oriented systems. The small rectangle labeled with "Delay" depicts the substantial time delay in the causal link between Fraction of Time Spent on SOA REQ and Service-Oriented System Delivery Rate. In Figure 2, the balancing loop B1 represents the fact that developers implement service-oriented systems under certain pressure created by the combination of the organization's lack of IS agility and management commitment to SOA implementation. By a balancing loop B1, it suggests that the IS Agility Gap is being closed over time when more service-oriented systems are implemented and installed, releasing the Pressure to Implement SOA.

Figure 2: Balancing Loop B1: Implement Service-Oriented Systems under Pressure



Balancing Loop B2 & B3

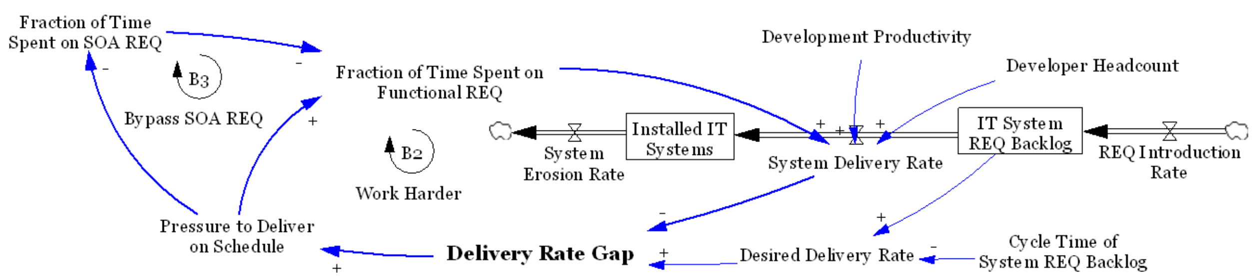
Figure 3 shows the normal work structure of IT developers in their daily work lives. With the IT System REQ Backlog and the cycle time requested by business units for delivering IT systems, IT managers and developers calculate the Desired Delivery Rate. The IT department has its actual System Delivery Rate which is determined by Developer Headcount, a developer's average Development Productivity, and how much time developers need to spend on functional requirements of the IT systems. Note that Development Productivity and System Delivery Rate are two distinct performance indicators. Development Productivity refers to on average how many IT systems[5] that a developer can deliver within one unit of time (say one month) when the developer spends all of his work hours on the development of functionalities. Differently, System Delivery Rate refers to how many IT systems that the development team as a whole (e.g.,

[5] For measurement consideration, a developer's development productivity can be measured as unit of software components, modules or functional features.

the entire IT department) can deliver within one unit of time during which the developers may spend part of their work hours on implementing SOA requirement or attending training sessions about SOA, etc. From the perspective of managers, System Delivery Rate is an aggregate-level and more salient performance indicator. A manager who used to be the CIO of a large US university told us that:

> As a manager, I usually don't care much about a single developer's productivity. I always care about how fast we [as a team] are able to deliver the systems requested [by business units]. In other words, we mostly care about the overall system delivery rate .

Figure 3: Balancing Loop B2 & B3: Work Harder to Deliver on Schedule and Bypass SOA



Delivery Rate Gap refers to the difference between Desired Delivery Rate and System Delivery Rate. Unlike IS Agility Gap creating Pressure to Implement SOA, the Delivery Rate Gap can create the pressure to deliver IT systems on schedule (whether service oriented or not), reflecting the fact that the primary tasks of developers are to develop and deliver IT systems to end users from business units. Pressure to Deliver on Schedule has two simultaneous effects on developers' work decisions: on the one hand, developers are forced to spend a substantial faction of time on implementing the functional requirements requested by end users. In this case, developers have to work harder on functional development (more essential tasks of their works) and try to catch up the delivery schedule; the balancing loop B2 represents this case. After conducing 27 field interviews on software development projects, Abdel-Hamid and Madnick (1991) stated that "when faced with schedule pressures as a project falls behind schedule, software developers typically respond by concentrating more on the essential tasks of the job" (Abdel-Hamid et al. 1991) (p17). On the other hand, the developers, now working harder on functional development, have to reduce part of their work hours that would have been spent on SOA requirements otherwise. In fact, when asked how developers made the tradeoff of work hours under the pressure to deliver systems on schedule, the IT manager of an interviewed organization replied:

> The requirement list we received from other departments usually put functional requirements on top of non-functional [service-oriented] requirements. But those non-functional requirements were not mandatory. When we received the requirement list, we would check it and if we don't have enough time, we just cut off those non-functional requirements.… After all, we have to deliver the capabilities [functionalities of the IT systems] to our end users within the limited schedule and resources.

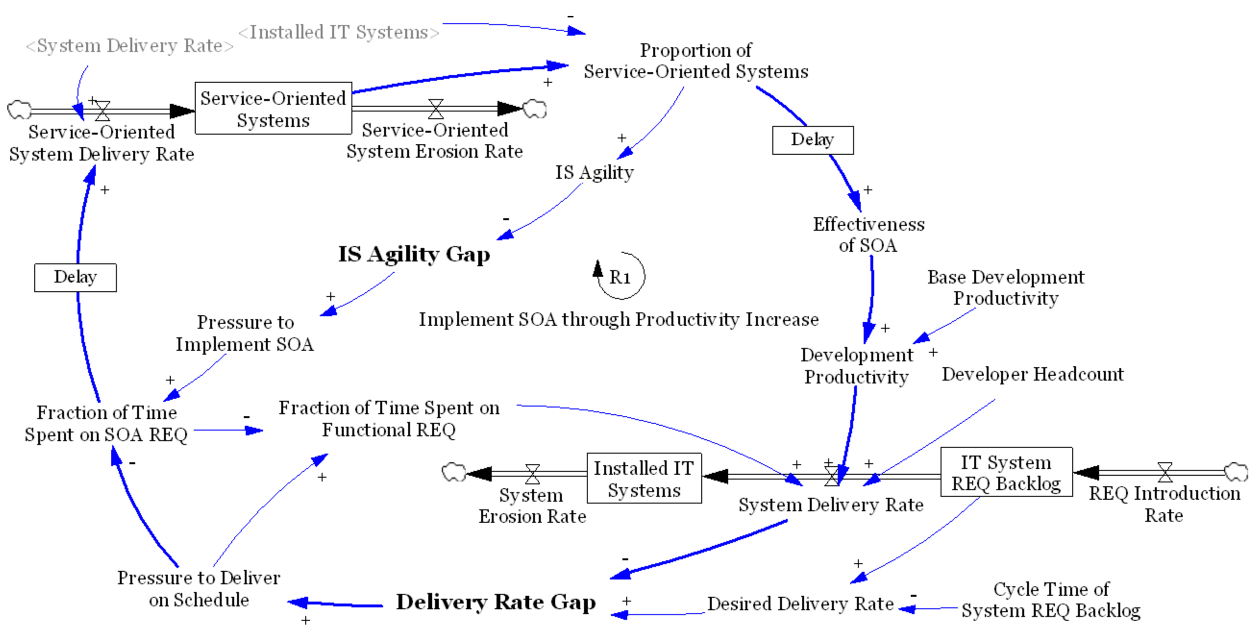
The balancing loop B3 captures such a situation in which developers could bypass the SOA requirements in order to deliver the functionalities of the IT systems on schedule. Both balancing loops B2 and B3 reveal that developers tend to bypass the SOA requirements and work harder to get their development jobs done under the schedule pressure. Balancing loops B2 and B3 close the Delivery Rate Gap and release the Pressure to Deliver on Schedule.

Reinforcing Loop R1

Now we turn to the potential benefits of SOA implementation. Service-oriented systems, developed using SOA design principles (e.g., modularity, loose coupling and standards), are more reusable, interoperable and easier to integrate with other IT systems (Choi et al. 2010; Mueller et al. 2010). Therefore, it is easy for developers to make use of the existing reusable service-oriented systems (e.g., components or services) when they develop new IT systems and integrate them with existing service-oriented systems that are already installed in the organization. In this sense, SOA implementation actually makes the development job easier and allows developers to develop IT systems faster. In other words, SOA implementation increases

Development Productivity of the developers on average (Choi et al. 2010; Hau et al. 2008; Mueller et al. 2010). In Figure 4, we use a key variable Effectiveness of SOA to represent the extent to which the developers' Development Productivity is increased, on average, compared to their Base Development Productivity in the normal situation prior to SOA implementation. Specifically, the more service-oriented systems installed in the organization (i.e., the higher Proportion of Service-Oriented Systems), the more Effectiveness of SOA the developers enjoy and in turn the higher their average Development Productivity. The rise of Development Productivity results in the increase of System Delivery Rate.

Figure 4: Reinforcing Loop R1: Implement SOA through Productivity Increase



With higher System Delivery Rate, Delivery Rate Gap is closed and the Pressure to Deliver on Schedule is released. As a result, developers with less schedule pressure are more likely to spend more time on developing service-oriented systems, increasing the System-Oriented System Delivery Rate and the number of installed System-Oriented System. Eventually, the Proportion of System-Oriented System will rise further and SOA becomes more effective, which further enhance the developers' Development Productivity. The entire process becomes a reinforcing loop R1 which is labeled as "Implement SOA through Productivity Increase", as shown in Figure 4. A reinforcing loop can operate as either virtuous (say better and better) or vicious (say worse and worse) cycles, depending on its current state (Sterman 2000). When the reinforcing loop R1 operates as virtuous cycles, more service-oriented systems are implemented (i.e., more penetration of SOA in the organization), SOA becomes more effective and allows developers to invest more time on further SOA implementation. Conversely, when R1 operates as vicious cycles, less SOA penetration in the organization generates little effectiveness of SOA and contributes little to the developers' development productivity. In this case under the schedule pressure, developers are likely to shift more time which would have been spent on SOA implementation to the functional development. Consequently, less service-oriented systems are developed and the SOA penetration becomes even less.

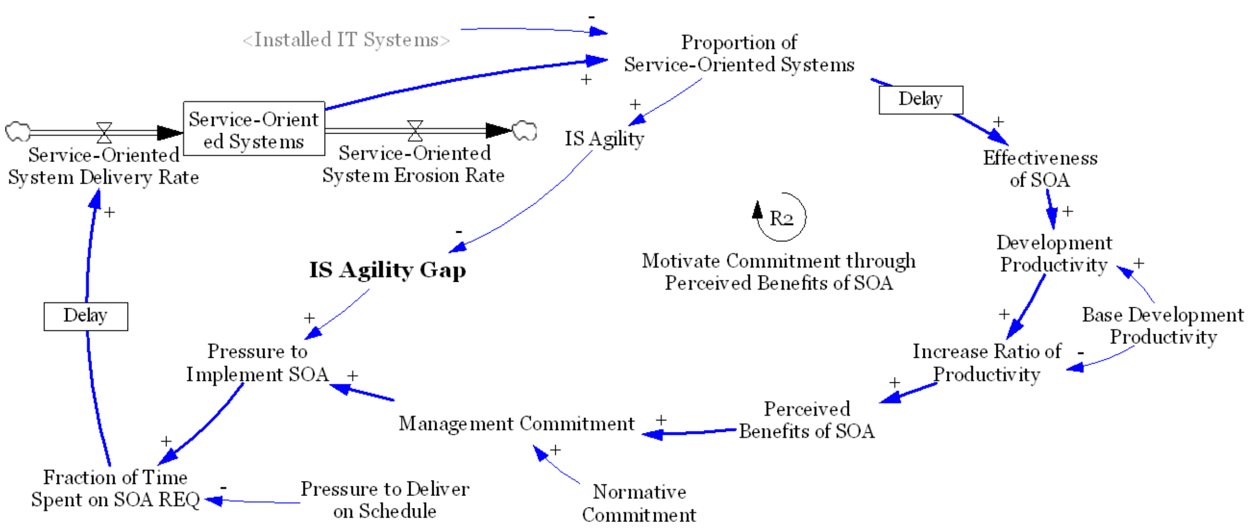
In order to successfully implement SOA, the reinforcing loop R1 operating as a virtuous cycle is preferable. However, this cannot always be the case. Many organizations struggling with their SOA implementation did actually suffer from the vicious cycles of the dynamics (Hau et al. 2008).

It is worth noting that there is a substantial delay between Proportion of Service-Oriented Systems and Effectiveness of SOA, because it takes time for developers to attend training sessions and learn experience so as to acquire sufficient knowledge about how to make use of existing reusable components/services and implement new service-oriented systems. Choi et al. (2010) explicitly documented that "the learning curve and the introduction of the governance mechanism will introduce a delay in implementation, as opposed to using current technology". We will discuss the impact of the delays in the next section.

Reinforcing Loop R2

As discussed, Effectiveness of SOA increases the developers' Development Productivity, compared to their Base Development Productivity. The degree to which extent the development productivity is increased can be represented as Increase Ratio of Productivity, as shown in Figure 5. The bigger the ratio, the more benefits and value are perceived by the organizational actors (e.g., managers and developers). The positive relationship between the effectiveness of innovative IT (particularly the effectiveness of SOA) and the perceived benefits has been largely discussed (Mueller et al. 2010). In particular, Choi et al. (2010) pointed out that SOA implementation effectiveness is an important determinant of the perceived benefits and value derived from SOA.

Figure 5: Reinforcing Loop R2: Motivate Commitment through Perceived Benefits of SOA



Greater perceived benefits of SOA create favorable word of mouth in the organization (Sterman 2000) and generate the additional commitment to SOA implementation internally. The causal link between results of the technology in use and the commitment generation has been supported by many motivation and organizational theories (Repenning 2002; Vroom 1964). In this research, the commitment generated by perceived benefits of SOA are considered to be internal or endogenous, emphasizing the additional commitment actually results from the results attribute to use of service-oriented systems. This is supported by social cognition theory saying "Performance successes strengthen self-beliefs of capability" (Wood et al. 1989).

Besides the endogenous sources of commitment, there are exogenous sources of commitment which are labeled as Normative Commitment. Institutional theory suggests that coercive, mimetic and normative pressures are important factors affecting the innovation adoption (DiMaggio et al. 1983). The work (Liang et al. 2007) on the assimilation of enterprise systems also provides support that institutional pressures positively affect top management participation in the ERP assimilation process. In other words, management commitment and participation mediate the effects of institutional pressures on IT assimilation (Liang et al. 2007).

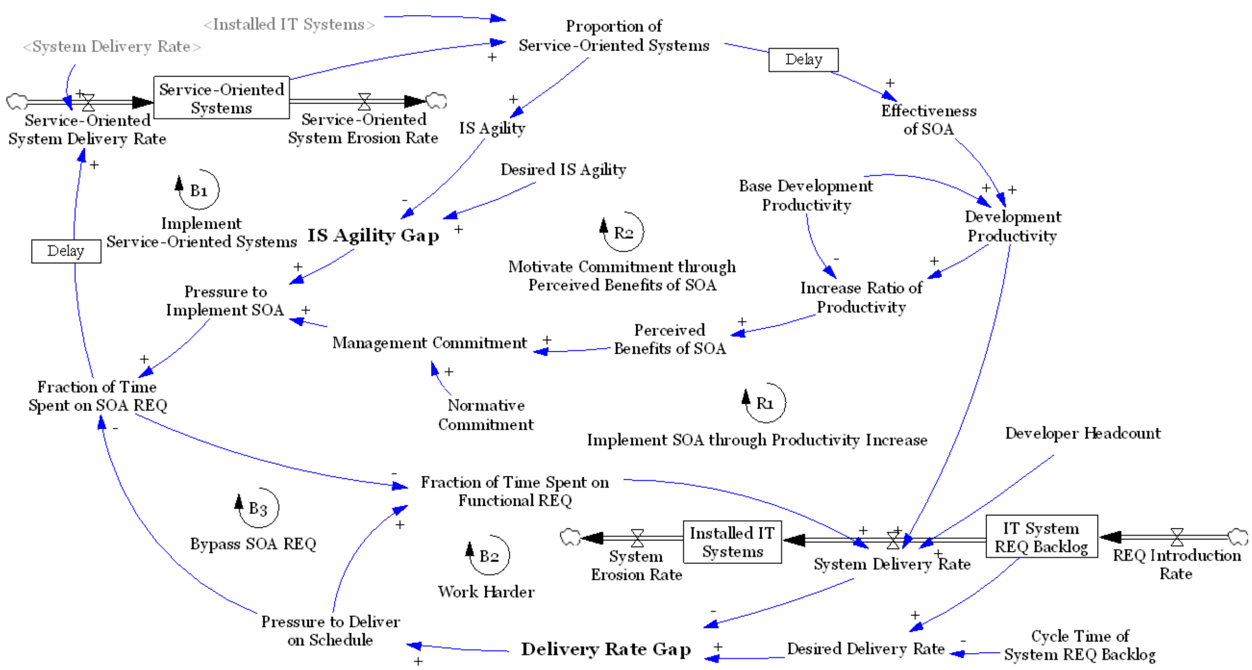
Figure 5 shows the mediating role played by Management Commitment between Normative Commitment and SOA implementation. Management Commitment generates Pressure to Implement SOA and forces developers to spend part of their work hours on implementing SOA, even though they are still under the Pressure to Deliver on Schedule. When service-oriented systems are implemented and installed over time, SOA implementation becomes more effective and enhances developers' Development Productivity. With more benefits of SOA perceived by the organization, the effects of favorable word of mouth would generate internal commitment endogenously and in turn enhance the total Management Commitment. The reinforcing loop R2, labeled as "Motivate Commitment through Perceived Benefits of SOA", represents the dynamics of the commitment motivation. Similar to R1, the reinforcing loop R2 operating as virtuous cycles is clearly preferable and highlights the importance of management commitment, which is consistent with the existing empirical evidence (Boh et al. 2010; Lee et al. 2010).

The Causal-Loop Model

The causal-loop model in Figure 6 includes three balancing loops and two reinforcing loops.

The balancing loop B1 indicates developers invest part of their work hours to implement service-oriented systems. There are two kinds of pressures that affect the tradeoff decision of IT developers and managers: one is the pressure to deliver the functionalities of IT systems on schedule, and the other is the pressure to SOA implementation. The balancing loop B1 operates to close the IS agility gap over time, yet management commitment to SOA plays the important role in generating the momentum for SOA implementation.

Figure 6: The Causal-Loop Model of SOA Implementation



The balancing loop B2 represents the decision that developers would naturally make to work harder and to get the functional development completed on schedule. The decision actually shifts part of their work hours to the functional development and bypasses SOA implementation, as revealed by the balancing loop B3. Both B2 and B3 indicate the tendency of developers that puts high priority on delivering functionalities to end users on schedule, which is confirmed by many managers that we interviewed. B2 and B3 operate to close the delivery rate gap and release the schedule pressure. It is worth noting that there is relatively shorter delay within B2 and B3 than the delay within B1. In case of a large delivery rate gap (e.g., urgent IT functionalities are requested by end users from business units), it is very likely for the IT managers and developers to make the decision that bypasses the SOA requirements and accelerates the development of functionalities. This is because System Delivery Rate is a more salient performance indicator associated to closing the Delivery Rate Gap more quickly, while IS Agility is an organizational, less salient performance indicator. That human beings tend to overemphasize salient factors when processing attributions is a well-known cognitive and perceptual bias (Tversky et al. 1974). Since bypassing SOA requirements only undermines the IS agility in the long term, it is less likely for people to attribute the unsatisfactory IS agility to such shortcuts due to the substantial delay. In contrast, closing the Delivery Rate Gap more quickly may probably bring IT managers and developers favorable gains or avoid negative words from other organizational actors (e.g., end users from business units).

The reinforcing loop R1 represents developers' development productivity increase over time with more and more service-oriented systems installed in the organization, releasing the Pressure to Deliver on Schedule. Under less intense schedule pressure, it is more likely for IT managers and developers to invest their work hours in SOA implementation. However, developers cannot immediately acquire the knowledge of SOA considering the technical complexity (Choi et al. 2010) and thus cannot increase their development productivity in a short period. The substantial delay in the reinforcing loop R1 has two important effects on the decisions of IT managers and developers. On the one hand, the substantial delay suggests managers have to keep investing in SOA for a long time and sacrifice the system delivery rate before the development productivity takes off. This means the “worse” period of the “worse-before-better” pattern (Repenning et al. 2001) may last long, but apparently, not every organization or manager would tolerate a substantially long “worse” period. On the other hand, the substantial delay makes the causal link between the effectiveness of SOA and system delivery rate uncertain and less salient. It is thus difficult for people to attribute the rise of system delivery rate to the investment in SOA implementation several months or even years ago, because people tend to attribute to more available and salient causes due to cognitive biases (Tversky et al. 1974).

The reinforcing loop R2 represents the situation where an organization perceives more benefits from SOA implementation when the development productivity is being improved over time. Management commitment to SOA may initially come from normative commitment from top managers. Meanwhile, the perceived benefits of SOA would motivate management commitment from the internal environment of the organization (perhaps from various organizational actors including managers and developers) and promote the SOA implementation further, resulting in more perceived benefits. Similar to R1, R2 is also subject to the substantial delay between higher SOA penetration and the rise of development productivity. Yet the substantial delay in R2 has a very important effect but different from the two effects of the delay in R1. That is, the substantial delay in R2 largely postpones the potential benefits of SOA to be perceived by organizational actors. Accordingly, organizations that decide to implement SOA have to be patient enough and tolerate a perhaps long period during which perhaps little benefits of SOA are perceived, especially at the early stage of SOA implementation. Thus, the primary part of management commitment to SOA has to come from normative commitment. In such situations, top management (e.g., CIOs) has to use their leadership to resist possibly unfavorable initial word of mouth about SOA. The normative commitment has to be maintained for long enough before perceived benefits of SOA arrive and enough endogenous commitment is motivated. The implication is that top management's leadership in maintaining normative commitment is important to leverage the benefits of IT in general (Armstrong et al. 1999) and SOA in particular.

THEORY

Learning Curve of SOA Implementation

SOA, as a new architectural style (Borges et al. 2004), has certain technical complexity and its implementation process is characterized by a high learning barrier (Choi et al. 2010). In order to implement service-oriented systems and make use of them in the future IS development, developers have to invest substantial time and energy in learning the new architectural style, e.g., learning SOA design principles and methodologies (Mueller et al. 2010) [6]. This suggests there is substantial delay from the investment in SOA to the rise of development productivity and the perceived benefits of SOA; the delays are labeled in Figure 6. In other words, it is likely to see little rise of development productivity and little perceived benefits at the early stage of SOA implementation process.

Prior research has demonstrated the presence of an organizational learning curve in the implementation of software packages (Saraswat et al. 1990), the adoption of CASE tools (Kemerer 1992), and software development methodology (Boh et al. 2007). Traditional wisdom originally from industrial learning curve suggests that "the rate at which the average cost of production decreases as the cumulative amount produced increases" (Kemerer 1992) and that the

[6] Learning SOA for an organization is far more than just SOA design principles and methodologies. Based on the evidences from our interview data, we found there are at least three levels of SOA learning: (1) basic concepts, design principles; (2) specific tools and technologies (e.g., J2EE); and (3) specific practices, standards and methodologies used in the organizations.

learning curve is shaped as exponential decreasing (Argote et al. 1990; Kemerer 1992). Steeper learning curve implies higher learning rate and more rapid cost decreases, which is a favorable case.

However, considering the substantial delays in the reinforcing loops R1 and R2 (as labeled in Figure 6), we postulate the learning rate of SOA implementation may not be decreasing constantly during the implementation process. Specifically, we hypothesize *the learning curve of SOA implementation is likely to be flat or decrease very slowly at the early stage of the implementation process. Only after a certain point, the learning curve would become steep*. In other words, it is likely that the learning curve of SOA implementation turns out to be reversely S-shaped, instead of an exponential-decreasing shape suggested by the traditional wisdom (Argote et al. 1990; Kemerer 1992). The hypothesis on the learning curve of SOA implementation has been observed empirically. As Hau et al. (2008) note, "the first release of an SOA application to take additional time because adherence to SOA design principles often leads to longer design time without yielding immediate benefits" (Choi et al. 2010; Hau et al. 2008).

In fact, the hypothesis on the learning curve of SOA implementation can be generalized to a larger context of innovative technology implementation. According to a survey of 60 sites, Chew et al. (1991) conclude that performance on initial projects with new technology implementation is usually worse than performance on projects with the old technology and this effect eventually wears off with improved positive performance (Chew et al. 1991). After investigating CASE tool adoption from a perspective of learning curve, Kemerer also observed that initial projects adopting CASE tools are relatively more expensive than later projects (Kemerer 1992), which is similar to the observation of SOA implementation done by Hau et al. (2008). This research suggests that the time delays in the reinforcing loops R1 and R2 is a plausible explanation for the reversely S-shaped learning curve of innovative technology implementation in general and SOA implementation in particular.

Organizational Traps

As indicated above, due to the substantial delay, managers and developers are likely to bypass SOA requirements and/or underinvest in SOA, especially when urgent IT functionalities are requested by end users from business units and intense schedule pressure is created. Bypassing SOA requirements may even be institutionalized in some organizations. A project manager that we interviewed told that:

> There are actually "waiver processes". When the requests [of IT functionalities] from end users are urgent enough or some emergence happens, they can apply for the waiver and don't need to go through the whole process [e.g., bypassing SOA requirements]. This is a tactic vs. strategic balance in our organization. And different departments actually have different waiver processes…

Underinvestment in SOA postpones the first release of an SOA application and perhaps allows delivering non-service-oriented systems, leading to less penetration of SOA in the organization and in turn less effectiveness of SOA implementation. As a result, the perceived benefits of SOA are further delayed and negative word of mouth may spread across the organization. For example, the technical complexity of SOA is overemphasized, instability of open standards for SOA and inappropriateness of SOA are misperceived (Choi et al. 2010; Hau et al. 2008). In such situations, the organization is likely to be stuck in two different traps: technology learning trap and implementation effectiveness trap.

Technology learning trap refers to the situation in which a certain technology is less understood due to insufficient learning, the more difficult and complex the technology is perceived. Technology learning trap indicates a vicious cycle of the "learning-by-doing" (Arrow 1962) or more specifically "learning-by-using" process (Rosenberg 1982). Consequently, developers may continue to underinvest in SOA and thus the initial flat part of the learning curve of SOA implementation is actually prolonged, which postpones the perceived benefits of SOA even further. Thus, the organization is trapped in the initial flat yet prolonging part of the learning curve of SOA implementation and can hardly see it taking off ever.

Implementation effectiveness trap refers to the situation in which SOA is misperceived to be inappropriate when SOA implementation is temporally less effective and the perceived benefits of SOA are delayed. It is likely for the organization to falsely conclude that SOA is inappropriate to its organizational context, rather than to admit that it is just due to insufficient SOA penetration in the organization. As a result, developers continue to underinvest in SOA and deliver non-service-oriented systems, further undermining the effectiveness of SOA implementation. In such a case, the organization is stuck in the trap and can hardly realize the effectiveness of SOA.

Technology learning trap and implementation effectiveness trap are different but intertwined with each other. Because top management team may change (Lee et al. 2004) or their strategic attention (and resources) often fades away over time (Kotter 1995), the two traps result in the failures of SOA implementation efforts of many organizations when the normative commitment to SOA fades away.

Note that the theory of the two organizational traps suggested in this research rejects the traditional perspective of a static, deterministic learning curve of innovation technology implementation (Boh et al. 2007; Chew et al. 1991; Kemerer 1992; Saraswat et al. 1990). The learning rate (i.e., the slope of the learning curve) is nondeterministic and actually affected by various organizational contexts and properties, such as the interpretation and actions of organizational actors as well as structures (Orlikowski 1992; Orlikowski 1993). The view of a dynamic learning curve may explain the inconsistent statistics of measuring the learning curve for CASE tools disclosed by Kemerer (1992), as prior research with technological determinism tends to ignore organizational contexts and properties which may affect organizational change and learning. Further, the theory of organizational traps suggests the social world (in this case the organizational outcomes of SOA implementation) is "produced and *reinforced* by humans through their action and interaction" (Orlikowski et al. 1991).

Causes of the Traps

We extend the theory of organizational traps in SOA implementation by discussing the plausible causes contributing to the traps.

First, the fundamental tension that results in the traps during SOA implementation is the tradeoff between short-term performance drop and the potential long-term benefits, indicated by the "worse-before-better" pattern (Repenning et al. 2001). As Hau et al. (2008) note, one of the primary challenges of SOA implementation is that "many firms failed to realize the benefits of SOA because they suffered from the inherent tradeoff between long-term benefits versus short-term local needs of project management" (Hau et al. 2008). Investing resources (e.g., time and financial budgets) in SOA implementation and corresponding organizational change clearly disrupts the normal operation of the organization to a large extent. In particular, developers need to devote substantial amount of their work hours to SOA implementation, which decreases their responsiveness to the request for IT functionalities from business units at the early stage of the implementation process. As the model suggests (see Figure 6), there are two different performance gaps that managers and developers need to close up: delivery rate gap and IS agility gap. In order to improve IS agility, the organization probably has to sacrifice the short-term performance. However, system delivery rate is a more salient, immediate and certain performance indicator and the urgency of IT functionality request from business units often emphasizes the salience, whereas IS agility is an organizational-level, less salient, less immediate and uncertain performance indicator. Because of the cognitive and perceptual bias that human tend to overemphasize salient and certain factors when processing attributions (Tversky et al. 1974), organizational actors are likely to overweight delivery rate gap and ignore improving IS agility. Repenning and Sterman (2002) support the tradeoff tension and point that "subjects [in many experiments] have been shown to grossly overweight the short-run positive benefits of their decisions while ignoring the long-run, negative consequences" (Repenning et al. 2002). In some circumstances, sacrificing the long-term benefits of SOA seems inevitable for some organizations. The CIO of a large energy company explained this dilemma:

> Firms will likely scale back on SOA investment due to economic conditions, sacrificing long term benefit for short term gains. As the short term view is focused on survival, this is the right change of focus. This will result in higher overall SOA costs as investments to date will either become stranded, or written off. At some future point, when such projects resume, technology and staff will have changed, not permitting continuity from where things were left. Time to realize benefits will be extended due to both total cost and total time to implement.

Second, there may be interest conflicts of different groups of organizational actors (e.g., top management, business line units, and IT unit). Local business units often focus much on how fast their requests of new IT functionalities can be delivered on schedule, so that they can catch the business opportunities that come out but may disappear in a short period. Based on our interviews, local business units usually do not understand or appreciate much about the organizational IS agility. In many cases, local business units have much power to urge their requests, because they control the resources of generating revenues for the organization. Thus, organizations probably have to give up part of the needs of local business units to some extent at least at the early stage of the implementation process. Hau et al. (2008) discuss the tension between "long term global benefits of SOA [e.g., IS agility] and short term local needs of project management [requested by business units]" (Hau et al. 2008) is a key challenge in SOA implementation. Choi et al. (2010) explicitly point out that "most firms cannot afford giving up short-term benefits when project management goals [set by business units] and the necessity to adhere to SOA design principles conflict with each other" (Choi et al. 2010).

Third, SOA, as a complex architectural style, manifests the agency of organizational actors (e.g., IT managers and developers) to control their interaction with the technology of SOA and its characteristics. Thus, organizational actors have much flexibility in design, implement, use, and interpretation of service-oriented systems, indicating the notion of interpretive flexibility of technological artifacts (Orlikowski 1992). An IT project manager from a large organization clearly noted:

> Our developers usually have alternative ways to deliver the same capability [functionalities of the IT systems] to end users. Since we have different choices, we chose the way that we think is appropriate to develop the systems… We are able to bypass the service-oriented requirements when we do not have enough time and resource to do it or SOA is not a good idea…

Interpretive flexibility allows the technology of SOA to be appropriated in diverse ways by actors in different organizations or by the same actors in different organizational contexts. Thus, there is possibility that organizational actors may inappropriately implement or use SOA and misinterpret the effectiveness of SOA implementation. An IT manger from a large software vendor commented on the challenge of interpretive flexibility of SOA implementation:

> It is difficult to monitor along the way whether the developers actually use the SOA standards and methodology to build the systems. So I think QA [quality assurance] is important. But even there is a QA process, it usually comes in at the end of the project. Enforcement of compliance to SOA standards and methodology is challenging.

While organizations enjoying the perceived benefits of SOA early are more dedicated to their SOA implementation efforts, those falling short of their expectation of SOA are likely to underinvest in their SOA implementation, resulting in an even worse situation. It is less likely for them to attribute the worse results of SOA implementation to their past actions (e.g., underinvestment in SOA). The complex dynamics of SOA implementation process bias organizational actors' interpretation and appropriateness of SOA, triggering the vicious cycle of declining SOA implementation.

Finally, when the perceived benefits of SOA do not meet the immediate expectations, managers and developers tend to blame the technical complexity and inappropriateness of SOA instead of attributing that they have bypassed the SOA requirements in their past actions. Organizational actors in the case of SOA implementation are more likely to attribute the ineffectiveness to the technology itself. The tendency of humans to blame the technology rather than themselves is widely observed and documented in the literature (Avital et al. 2001). As Shneiderman (1990) notes, "Complex and confusing systems enable users and designers to blame the machine" (Shneiderman 1990). Also, Brown et al. (1998) wrote that "An understandable reaction [for frustrated users] is to blame the technology, but the attempts to achieve the advantages of information systems can be thwarted by both technological and organizational constraints" (Brown et al. 1998). After observing the adoption process of the faculty educational technology in a university, Moser (2007) found that "If technology was involved, however, faculty were quick to blame the failure on the technology and abandon newly acquired teaching practice and technology use" (Moser 2007). When stuck in the technology learning trap and implication effectiveness trap, blaming the complexity and inappropriateness of the technology, providing a possibility for organizational actors to excuse their past actions of underinvestment in SOA implementation, is actually misperceived and false. The vicious cycle of declining SOA implementation reinforces the excuse and misperception. Such misinterpretation about the ineffectiveness or inappropriateness of SOA implementation reveals organizational actors' self-confirming attribution error (Repenning et al. 2002). Accordingly, Lorenzi et al. (2003) clearly suggest "Existing organizational and/or people problems often surface during the implementation of new technical systems. Instead of waiting for latent problems to emerge, organizations should deal with managerial problems before implementing new technology. If it is not possible to effectively handle the problems, the organization must at least avoid placing blame for the problem on the technological system" (Lorenzi et al. 2003).

IMPLICATION

This paper has presented the findings of a study on SOA implementation and, particularly, the theory of organizational traps associated with the implementation of innovative IS technologies. The findings and theory articulated here have important implications for research and practice.

Research Implication

This research assumes to a large extent that SOA, once implemented well in an organization, can enhance the organization's IS agility and developers' development productivity in the long term (Choi et al. 2010; Hau et al. 2008; Mueller et al. 2010; Yoon et al. 2007). Yet the intended benefits of SOA cannot be realized automatically and many organizations have encountered organizational and human challenges in their SOA implementation efforts (Fricko 2006; Luthria et al. 2008). Therefore, this research adopts the process perspective which is largely overlooked by prior studies that use variance models (Mohr 1982) and seek to explain the variance in outcomes from SOA. Using the process perspective is a clear distinction between this paper and the existing literature on SOA (Boh et al. 2010; Lee et al. 2010; Mueller et al. 2010; Yoon et al. 2007). By doing so, this research suggests a theoretical explanation for the contradictory results of SOA implementation paradox from an organizational perspective. That is, it is likely to be explained by the differences in processes, interpretations and actions around the implementation and use of service-oriented systems in different organizations. Due to the dynamics and nonlinearity of the process, the connections between starting conditions, actions, and outcomes in SOA implementation are not deterministic.

In fact, the extant IS literature indicates that human agency often plays an important, and sometimes critical, role in organization-wide IS implementation, like Enterprise Resource Planning (ERP) implementation (Orlikowski 1992; Volkoff et al. 2007). In the IS literature on ERP, Grant et al. (2006) write “key stakeholders in the ERP implementation process adopted different discourses” and highlighted the role of their discourses in the social shaping of ERP implementation (Grant et al. 2006). Scott el al. (2003) further point out that the “success” or “failure” of the ERP implementation is actually highly situated and relate to “the negotiations between actor networks surrounding the implementation process” (Scott et al. 2003). Although ERP as a monolithic IS architecture is very different from SOA and thus has different organizational implications, prior research on ERP leads us to accommodate human agency in this research and to examine the important role that human agency plays during the SOA implementation process.

The implementation of SOA manifests the agency and voluntarism of organizational actors (e.g., IT managers and developers). There is a high degree to which those organizational actors are engaged in SOA implementation process during both the implementation and use of service-oriented systems. As indicated in the model (see Figure 6), the actions that implement SOA and that use SOA are mutually influenced by each other, both of which in most circumstances are acted by the same organizational actors (e.g., IT managers and developers) in similar organizational context (time and space), because developers of service-oriented systems are also users of those systems. This characteristic of time-space continuity between the implementation and use of SOA illuminates the notion of "duality of technology" named by Orlikowski (1992), that is, "the technology is created and changed by human action, yet it is also used by humans to accomplish some action" (Orlikowski 1992). Thanks to the clear characteristic of time-space continuity, the implementation process of SOA is a meaningful field to study the duality of technology which can provide insights different from that derived from the conceptual dualism of technology dominating the IS literature (Orlikowski 1992).

SOA implementation requires organizations to invest substantial resources upfront before potential benefits are perceived by organizational actors, known as the "worse-before-better" phenomenon (Repenning et al. 2001). That is, there are substantial delays between the implementation investment and the perceived benefits. During the "worse" period of the implementation, different organizational actors often make different senses and judgments about what becomes "worse" to them, how "worse" it will be, and how long the "worse" will last. Impatient organizational actors are likely to underinvest in SOA implementation and thus the dynamics can get stuck in the two intertwined traps: technology learning trap and implementation effectiveness trap. Once stuck in the traps, it would be difficult for organizational actors to correctly attribute to the vicious cycle of the dynamics of the implementation process. Due to the misattribution, the subsequent reaction of the organizational actors may further exacerbate the situation of the vicious cycle. This research suggests that the technology learning trap and implementation effectiveness trap result from not only the characteristics of technology (in this case, SOA) and the inherent structure of the implementation process (i.e., the balancing and reinforcing feedback loops), but the dynamic interactions between human agency and the technology implementation process, indicating the duality of technology (Orlikowski 1992). The misattribution of organizational actors when they are stuck in the traps reflects another form of self-confirming error; yet it is different from that discussed in Repenning's works on process improvement in manufacturing (Repenning et al. 2002).

More empirical research on the theory of organizational traps and the theoretical explanation for SOA implementation paradox is needed. In particular, empirical validation and elaboration of the findings in this research will enrich the theory developed here and provide more refined understanding of the dynamics of the SOA implementation process in various organizational settings.

Practice implication

Managers who make the decision of SOA implementation for their organizations need to be aware and prepared of the potential traps in the implementation process. Long-term commitment is definitely helpful to SOA implementation. Patient and consistent decision-makings about the tradeoff between short-term performance drop and potential long-term benefits and about the tradeoff between local project needs and organization-level SOA implementation are important. In addition, institutionalization of the long-term commitment and consistent decision-making about the tradeoffs using appropriate governance mechanisms may increase the chance of successful SOA implementation (Joachim et al. 2011; Varadan et al. 2008).

References

Abdel-Hamid, T., and Madnick, S.E. *Software project dynamics: an integrated approach*, New Jersey, USA, 1991.

Argote, L., and Epple, D. "Learning curves in manufacturing," *Science* (247:4945) 1990, p 920.

Armstrong, C.P., and Sambamurthy, V. "Information Technology Assimilation in Firms: The Influence of Senior Leadership and IT Infrastructures," *Information systems research* (10:4) 1999, pp 304-327.

Arrow, K.J. "The Economic Implications of Learning by Doing," *The Review of Economic Studies* (29:3) 1962, pp 155-173.

Avital, M., and Vandenbosch, B. "SAP implementation at Metalica: an organizational drama in two acts," *Journal of Information Technology* (15:3) 2001, pp 183-194.

Bandara, W., Gable, G.G., and Rosemann, M. "Factors and measures of business process modelling: model building through a multiple case study," *European Journal of Information Systems* (14:4) 2005, pp 347-360.

Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M., and Virili, F. "Extensible Architectures: The Strategic Value of Service Oriented Architecture in Banking," *ECIS 2005 Proceedings*) 2005, p 61.

Bieberstein, N., Bose, S., Walker, L., and Lynch, A. "Impact of Service-Oriented Architecture on Enterprise Systems, Organizational Structures, and Individuals," *IBM Systems Journal* (44:4) 2005, pp 691-708.
Boh, W.F., Slaughter, S.A., and Espinosa, J.A. "Learning from experience in software development: A multilevel analysis," *Management science* (53:8) 2007, pp 1315-1331.
Boh, W.F., and Yellin, D.M. "Enablers and benefits of implementing Service-Oriented Architecture: an empirical investigation," *International Journal of Information Technology and Management* (9:1) 2010, pp 3-29.
Borges, B., Holley, K., and Arsanjani, A. "Service-Oriented Architecture: Components and Modeling Can Make the Difference," *Web Services Journal* (9:1) 2004, pp 34-38.
Brown, M.M., and Brudney, J.L. "Public sector information technology initiatives," *Administration & Society* (30:4) 1998, p 421.
Cao, L., Ramesh, B., and Abdel-Hamid, T. "Modeling dynamics in agile software development," *ACM Transactions on Management Information Systems (TMIS)* (1:1) 2010, p 5.
Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., and Rackham, G. "Impact of Service Orientation at the Business Level," *IBM Systems Journal* (44:4) 2005, pp 653-668.
Chew, W.B., Leonard-Barton, D., and Bohn, R.E. "Beating Murphy's law," *Sloan Management Review* (32:3) 1991, pp 5-16.
Choi, J., Nazareth, D.L., and Jain, H.K. "Implementing Service-Oriented Architecture in Organizations," *Journal of Management Information Systems* (26:4) 2010, pp 253-286.
DiMaggio, P.J., and Powell, W.W. "The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields," *American sociological review* (48:2) 1983, pp 147-160.
Forrester, J.W. *Industrial Dynamics*, Cambridge, MA, USA, 1968.
Fricko, A. "SOAs Require Culture Change and Service Reuse," *Business Communications Review* (36:5) 2006, p 58.
Georgantzas, N.C., and Katsamakas, E.G. "Information systems research with system dynamics," *System Dynamics Review* (24:3) 2008, pp 247-264.
Grant, D., Hall, R., Wailes, N., and Wright, C. "The False Promise of Technological Determinism: The Case of Enterprise Resource Planning Systems," *New Technology, Work and Employment* (21:1) 2006, pp 2-15.
Hau, T., Ebert, N., Hochstein, A., and Brenner, W. "Where to Start with SOA: Criteria for Selecting SOA Projects," Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS 2008), IEEE Computer Society, Waikoloa, Big Island, Hawaii, 2008, pp. 314-322.
Joachim, N., Beimborn, D., and Weitzel, T. "What Are Important Governance and Management Mechanisms to Achieve IT Flexibility in Service-Oriented Architectures (SOA)?: An Empirical Exploration," Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS 2011), IEEE Computer Society, Kauai, Hawaii USA, 2011, pp. 1-10.
Kanaracus, C. "SOA is Alive and Well, Forrester Says," Computer World, 2011.
Kemerer, C.F. "How the learning curve affects CASE tool adoption," *Software, IEEE* (9:3) 1992, pp 23-28.
Kotter, J. "Leading Change: Why Transformation Efforts Fail," *Harvard Business Review* (73) 1995, pp 59-67.
Law, J., and Urry, J. "Enacting the Social," *Economy and Society* (33:3) 2004, pp 390-410.
Lee, J.C., and Myers, M.D. "Dominant Actors, Political Agendas, and Strategic Shifts Over Time: A Critical Ethnography of An Enterprise Systems Implementation," *The Journal of Strategic Information Systems* (13:4) 2004, pp 355-374.
Lee, J.H., Shim, H.J., and Kim, K.K. "Critical Success Factors in SOA Implementation: An Exploratory Study," *Information Systems Management* (27:2) 2010, pp 123-145.
Liang, H., Saraf, N., Hu, Q., and Xue, Y. "Assimilation of Enterprise Systems: The Effect of Institutional Pressures and the Mediating Role of Top Management," *Management Information Systems Quarterly* (31:1) 2007, p 6.
Locke, K.D., and Locke, K. *Grounded theory in management research* SAGE Publications Ltd, 2001.
Lorenzi, N.M., and Riley, R.T. "Organizational issues= change," *International Journal of Medical Informatics* (69:2-3) 2003, pp 197-203.
Luthria, H., and Rabhi, F. "Organizational Constraints to Realizing Business Value from Service Oriented Architectures: An Empirical Study of Financial Service Institutions," *Service-Oriented Computing - ICSOC 2008*) 2008, pp 256-270.
Luthria, H., and Rabhi, F. "Service Oriented Computing in Practice: An Agenda for Research into the Factors Influencing the Organizational Adoption of Service Oriented Architectures," *Journal of Theoretical and Applied Electronic Commerce Research* (4) 2009a, pp 39-56.
Luthria, H., and Rabhi, F. "Using Service Oriented Computing for Competitive Advantage," Proceedings of Americas Conference on Information Systems (AMCIS 2009), 2009b, p. 140.

Sprouts

Markus, M.L., and Robey, D. "Information technology and organizational change: causal structure in theory and research," *Management science* (34:5) 1988, pp 583-598.
McClintock, C.C., Brannon, D., and Maynard-Moody, S. "Applying the logic of sample surveys to qualitative case studies: The case cluster method," *Administrative Science Quarterly* (24:4) 1979, pp 612-629.
McKendrick, J. "So Far, SOA failures are Few and Far Between, Survey Says," ZDNet, 2011.
Mohr, L.B. *Explaining organizational behavior: The limits and possibilities of theory and research* Jossey-Bass, San Francisco, CA, USA, 1982.
Moitra, D., and Ganesh, J. "Web Services and Flexible Business Processes: Towards the Adaptive Enterprise," *Information & Management* (42:7) 2005, pp 921-933.
Moser, F.Z. "Faculty Adoption of Educational Technology," *EDUCAUSE quarterly* (30:1) 2007, p 66.
Mueller, B., Viering, G., Legner, C., and Riempp, G. "Understanding the Economic Potential of Service-Oriented Architecture," *Journal of Management Information Systems* (26:4) 2010, pp 145-180.
Orlikowski, W.J. "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Organization Science* (3:3) 1992, pp 398-427.
Orlikowski, W.J. "CASE tools as organizational change: Investigating incremental and radical changes in systems development," *Mis Quarterly*) 1993, pp 309-340.
Orlikowski, W.J., and Baroudi, J.J. "Studying information technology in organizations: Research approaches and assumptions," *Information systems research* (2:1) 1991, pp 1-28.
Papazoglou, M.P., and Georgakopoulos, D. "Service-oriented computing," *Communications of the ACM* (46:10) 2003, pp 25-28.
Perlow, L.A., Okhuysen, G.A., and Repenning, N.P. "The speed trap: Exploring the relationship between decision making and temporal context," *The Academy of Management Journal* (45:5) 2002, pp 931-955.
Pettigrew, A.M. "Issues of time and site selection in longitudinal research on change," *The Information Systems Research Challenge: Qualitative Research Methods* (1) 1989, pp 3-19.
Pettigrew, A.M. "Longitudinal field research on change: theory and practice," *Organization Science* (1:3) 1990, pp 267-292.
Pfeffer, J. "New Directions for Organization Theory:: Problems and Prospects,") 1997.
Pfeffer, J., and Sutton, R.I. *The Knowing-Doing Gap: How Smart Companies Turn Knowledge into Action* Harvard Business School Press, Cambridge, MA, USA, 2000.
Rahmandad, H., Repenning, N., and Sterman, J. "Effects of feedback delay on learning," *System Dynamics Review* (25:4) 2009a, pp 309-338.
Rahmandad, H., and Weiss, D.M. "Dynamics of concurrent software development," *System Dynamics Review* (25:3) 2009b, pp 224-249.
Repenning, N.P. "A Simulation-Based Approach to Understanding the Dynamics of Innovation Implementation," *Organization Science* (13:2) 2002, pp 109-127.
Repenning, N.P., and Sterman, J.D. "Nobody Ever Gets Credit for Fixing Problems That Never Happened," *California management review* (43:4) 2001, pp 64-88.
Repenning, N.P., and Sterman, J.D. "Capability Traps and Self-Confirming Attribution Errors in the Dynamics of Process Improvement," *Administrative Science Quarterly* (47:2) 2002, pp 265-295.
Robey, D., Ross, J.W., and Boudreau, M.C. "Learning to implement enterprise systems: An exploratory study of the dialectics of change," *Journal of Management Information Systems* (19:1) 2002, pp 17-46.
Rosenberg, N. *Inside the Black Box: Technology and Economics* Cambridge University Press, Cambridge, 1982.
Rudolph, J.W., Morrison, J.B., and Carroll, J.S. "The dynamics of action-oriented problem solving: linking interpretation and choice," *The Academy of Management Review (AMR)* (34:4) 2009, pp 733-756.
Ryan, G.W., and Bernard, H.R. "Data Management and Analysis Methods," in: *Handbook of Qualitative Research,* Y.S.L. Norman K. Denzin (ed.), Citeseer, Sage, CA, USA, 2000, pp. 769-801.
Saraswat, S.P., and Gorgone, J.T. "Organizational learning curve in software installation: an empirical investigation," *Information & Management* (19:1) 1990, pp 53-59.
Sastry, M.A. "Problems and paradoxes in a model of punctuated organizational change," *Administrative Science Quarterly* (42:2) 1997, pp 237-275.
Scott, S.V., and Wagner, E.L. "Networks, Negotiations, and New Times: The Implementation of Enterprise Resource Planning into an Academic Administration," *Information and organization* (13:4) 2003, pp 285-313.
Sher, P.J., and Lee, V.C. "Information Technology as a Facilitator for Enhancing Dynamic Capabilities Through Knowledge Management," *Information & Management* (41:8) 2004, pp 933-945.

Shneiderman, B. "Human values and the future of technology: a declaration of empowerment," Proceedings of the conference on Computers and the quality of life, ACM, 1990.
Smith, R. "A Simpler Approach To SOA," Information Week 2008.
Sterman, J .D. *Business Dynamics: Systems Thinking and Modeling for a Complex World with CD-ROM* Irwin/McGraw-Hill, 2000.
Tversky, A., and Kahneman, D. "Judgment under Uncertainty: Heuristics and Biases," *Science* (185:4157) 1974, p 1124.
Varadan, R., Channabasavaiah, K., Simpson, S., Holley, K., and Allam, A. "Increasing Business Flexibility and SOA Adoption Through Effective SOA Governance," *IBM Systems Journal* (47:3) 2008, pp 473-488.
Volkoff, O., Strong, D.M., and Elmes, M. B. "Technological Embeddedness and Organizational Change," *Organization Science* (18:5) 2007, pp 832-848.
Vroom, V.H. *Work and Motivation* Wiley, New York, 1964.
Weick, K.E. *The social psychology of organization* Addison-Wesley, 1979.
Wheelwright, S.C., and Clark, K.B. *Leading Product Development* Free Press, 1995.
Winter, S.G. "Understanding dynamic capabilities," *Strategic Management Journal* (24:10) 2003, pp 991-995.
Wood, R., and Bandura, A. "Social Cognitive Theory of Organizational Management," *The Academy of Management Review* (14:3) 1989, pp 361-384.
Yin, R.K. "The case study crisis: Some answers," *Administrative Science Quarterly* (26:1) 1981, pp 58-65.
Yoon, T., and Carter, P. "Investigating the Antecedents and Benefits of SOA Implementation: A Multi-Case Study Approach," *AMCIS 2007 Proceedings*) 2007, pp 195-205.

芽|Sprouts

Working Papers on Information Systems | ISSN 1535-6078