Understanding the Organizational Traps in Implementing Net-Centric Systems

Xitong Li
Stuart Madnick

Working Paper CISL# 2012-07

June 2012

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E62-422
Massachusetts Institute of Technology
Cambridge, MA 02142

Understanding the Organizational Traps in Implementing Net-Centric Systems

Xitong Li (xitongli@mit.edu)

Stuart E. Madnick (smadnick@mit.edu)

June 8, 2012

1 Introduction

In an increasingly 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 (Choi et al. 2010) and has been considered as a key facilitator to enhance dynamic capabilities (Sher and Lee 2004) and competitive advantages (Luthria and Rabhi 2009b). However, with the traditional monolithic IS architecture, it is quite difficult, expensive, and time-consuming for organizations to make changes to their IT systems (Choi et al. 2010). To address these issues, Service-Oriented Architecture (SOA) has been advocated to provide 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 (Borges et al. 2004). The net-centric strategy in the defense and intelligence communities are closely analogous to the concept of SOA and make use of SOA technology as a key element (*viz.* Croom 2006, AFEI 2008, Śliwaa and Amanowicz 2011).

One of the major objectives of adopting net-centric systems and 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 net-centric systems. 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 the adoption and implementation of SOA 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 reported outcomes presents a paradox in the adoption and implementation of SOA and net-centric systems. The key question regarding the "implementation paradox" is why many organizations failed to meet their expectations about their implementation efforts, while others succeeded.

In the literature on net-centric systems, the potential benefits and business value of net-centric systems have been largely claimed (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 implementation (Baskerville et al. 2005; Moitra and Ganesh 2005). Despite the potential benefits claimed, organizations adopting net-centric systems often encounter challenges in implementation efforts. To address the challenges, more recent (yet relatively few) literature turns to explore critical success factors (CSF) and challenges that potentially affect the implementation of net-centric systems. Research in this stream tends to enumerate a number of factors that potentially facilitate or impede organizations to receive the intended benefits of net-centric systems. However, prior research failed to explore the causal relationships between 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 implementation of net-centric systems. Luthri et al. (2009a) extensively reviewed the literature and found that "*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 and Rabhi 2009a). The system dynamics model developed in this research aims to explore aspects that have not been extensively investigated in the literature.

Considering the certain technical complexity of net-centric systems and necessary organizational change, organizations adopting net-centric systems have to invest substantial resources (e.g., time and financial resources) in the implementation before they gain the potential benefits of net-centric systems. As Hau et al. (2008) argued, 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). From the literature on organizational theories, Repenning's works on process improvement (Repenning and Sterman 2001; Repenning and Sterman 2002) and innovation implementation (Repenning 2002) explore the challenge of capability traps in manufacturing and provide insights for us to address the challenging tradeoff faced by organizations in their implementation efforts in net-centric systems. Our research demonstrates similar but different phenomenon may happen in the failures of many implementation efforts of net-centric systems. Note that our research does not simply apply the existing theory of capability traps (Repenning and Sterman 2001; Repenning and Sterman 2002) to understanding the challenges in implementation. In fact, based on our investigation and observation, the implementation of net-centric systems has at least two inherent characteristics that are distinct from manufacturing process improvement implementation (see the discussion in the next section). Accordingly, the model and theory developed in this research suggest two different but intertwined organizational traps, i.e., technology learning trap and implementation effectiveness trap, play important roles in the failures of many implementation efforts of net-centric systems. Technology learning trap refers to the situation that the less learning in using the technology that has occurred, 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 net-centric systems when the implementation is temporally less effective and the perceived benefits of net-centric systems are delayed. Both organizational traps are distinct from the existing theory of capability traps. In addition, the second part of this research focuses on investigating how normative commitment helps to overcome the organizational traps and the learning barrier during the implementation process.

In this research, we use system dynamics modeling as the main research methodology (Sterman 2000). System dynamics has been proved to be a powerful modeling tools for organizational theory building (Repenning and Sterman 2002; Rudolph et al. 2009; Sterman 2000) and has recently received increasing attention for IS research (Choi et al. 2010; Georgantzas and Katsamakas 2008). In regard to research methodologies, Law and Urry (2004) pointed out that "*social science method has problems in understanding non-linearity relationships and flows [e.g., positive feedback loops]*" (Law and Urry 2004). Compared to other organizational research methodologies, system dynamics modeling has its unique strengths in that it allows this research to focus on the feedback loops and nonlinearity of organizational structures and time delay in the feedback loops. Besides system dynamics modeling, we use multiple research methods (including literature review, qualitative interviews, meetings/seminars and case study) for the model and theory building.

In sum, this research not only contributes to the IS literature but also sheds light on organizational theory and system dynamics literature.

The rest of this paper is organized as follows: Section 2 discusses the theoretical foundation. Section 3 presents the system dynamic model. Section 4 discusses the theory about the two organizational traps. After that, this paper turns to the second part of this research. Section 5 presents the experimental simulation results of the system dynamic model. Section 6 discusses the theory about normative commitment. Section 7 discusses the implications of this research and concludes the paper.

2 Theoretical Foundation

This research is mainly built upon three streams of the existing literature: the first two come from the literature on SOA and the third is drawn from the literature on process improvement. Each research stream provides theoretical foundation for the model and theory building.

2.1 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 and Carter 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 potentials 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 claimed 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 stream of literature generally adopts the perspective of technological determinism and fails to answer the puzzling "SOA implementation paradox": why many organizations failed to meet their expectations about SOA initiatives, while others succeeded.

2.2 Critical Success Factors of SOA Implementation

To answer the SOA implementation paradox, more recent (yet relatively few) literature turns to explore critical success factors (CSF) and challenges that potentially affect SOA implementation. The work (Luthria and Rabhi 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 and Rabhi 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 and Yellin 2010) examines two organizational factors that are potentially critical in ensuring the success of SOA implementation: (1) top management support and (2) 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 literatures 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).

In sum, the research stream 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. However, prior research failed 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.

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 our research capture this important point.

2.3 Capability Traps in Process Improvement

The third piece of theoretical foundation of this research is largely built upon Repenning's works on capability traps in process improvement (Repenning and Sterman 2001; Repenning and Sterman 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 recognizing 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 system 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 and Sutton 2000) and best practices for product development (Wheelwright and Clark 1995). The inability of many organizations to use the knowledge embodied in the improvement initiatives is a central issue facing organizational theorists (Pfeffer 1997). However, there is little IS literature documenting whether or not the similar phenomenon has happened during IS improvement efforts, especially in SOA implementation. After all, the capability traps result from the interaction between human judgmental biases and the physical structure of work processes (Repenning and Sterman 2002). This research explores whether phenomena like capability traps may also be involved 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. In fact, based on our investigation and observation, SOA implementation has at least two inherent characteristics that are distinct from TQM and process improvement programs:

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 and Sterman 2002). Thus, favorable results and positive word of mouth are easier to achieve and come earlier from the investment in improvement. Unfortunately, SOA implementation is not the case. The learning curve of the complex technology like SOA creates even 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.

3 Model Building

3.1 Causal-Loop Diagrams

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) 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 research are not intended to provide an accurate mathematical specification of the relationships in the causal links; but they demonstrate the nonlinearity, discontinuities and delays between causes and effects and are valid in organizational theory building

(Repenning and Sterman 2001; Repenning and Sterman 2002). Specifying a formal mathematical model is often the next step in testing the theories embodied in the causal-loop diagrams and is the second part of this paper. It is also worth noting that in this research we intentionally avoided replicating the context of a specific firm; we removed the details that were merely applied to a specific firm. The model and theory building aim to generate insights and implications in a broad organizational context.

The first assumption in our model is that only a proportion of delivered IT systems in an organization are net-centric (NC) systems. The rationale of this assumption lies in that IT developers need to spend extra time and energy to follow the net-centric principles (e.g., modularity, loose coupling, and standards) (Mueller et al. 2010) in order to make the delivered systems net-centric. Otherwise, the delivered systems just turn out to be non-net-centric and are installed in the organization. The first key variable in the model is `Proportion of NC IT Systems`, which is the ratio of the number of `NC IT Systems` to the number of total `Installed IT Systems`. Basically, `Proportion of NC IT Systems` is used to capture the penetration of net-centric systems in the organization during the implementation process. 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 when system requirements are introduced over time. The delivered IT systems are installed with `System Delivery Rate`. `NC System Delivery Rate` is a fraction of the overall `System Delivery Rate`; the fraction actually depends on how much time developers spend on implementing net-centric requirements, `Fraction of Time Spent on NC REQ`. The "+" sign at the head of the causal link from `System Delivery Rate` to `NC 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 `NC System Delivery Rate`. Any IT systems regardless of net-centric or not may erode over time due to the change of business environment or need for technology upgrade.

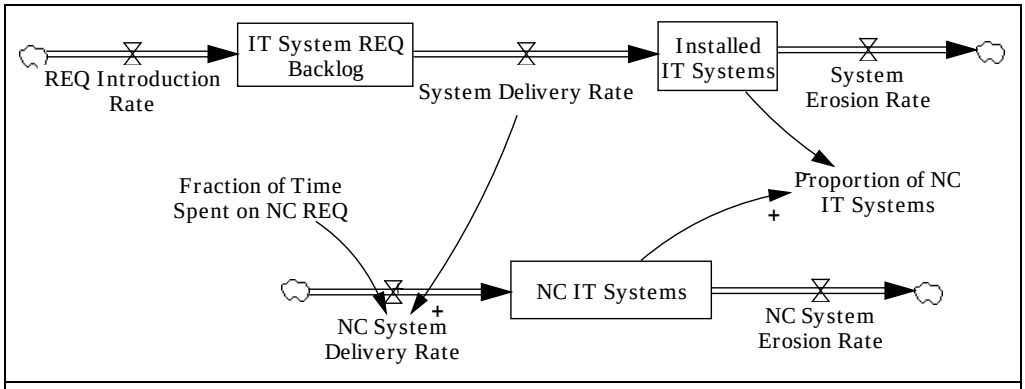


Figure 1. Proportion of Installed IT Systems are Net-Centric (NC) Systems

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 7) in mind while reading through each of the feedback loops.

3.1.1 Balancing Loop B1: Implement Net-Centric IT Systems under Pressure

Figure 2 shows the higher `Proportion of NC IT Systems`, as the conceptualization of the organizational penetration of net-centric systems, 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 NC Systems`. 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 NC Systems`, reflecting the fact that management commitment and support is a critical success factor of SOA implementation (Boh and Yellin 2010; Lee et al. 2010). Under the `Pressure to Implement NC Systems`, developers are forced to put a fraction of the work hours spent on implementing net-centric

requirements. The time spent on net-centric requirements represents the developers need to spend extra time to follow the net-centric design principles when they develop the IT systems. The more time spent on net-centric REQ, the higher NC System Delivery Rate.

Developers not only face the Pressure to Implement NC Systems, but also the pressure to deliver IT systems on schedule, 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 large faction of time on implementing the functional requirements requested by end users. On the other hand, the developers actually shift away part of their work hours that would have been spent on net-centric 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 [net-centric] 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.

Therefore, under the two kinds of pressures, developers first decide how much time they would spend on the functional development and then spend the rest of their time on net-centric requirements. In Figure 2, the balancing loop B1 represents the fact that developers implement net-centric IT systems under the two kinds of pressures. By a balancing loop B1, it suggests that the IS Agility Gap is being closed over time when more net-centric systems are implemented and installed, releasing the Pressure to Implement NC Systems.

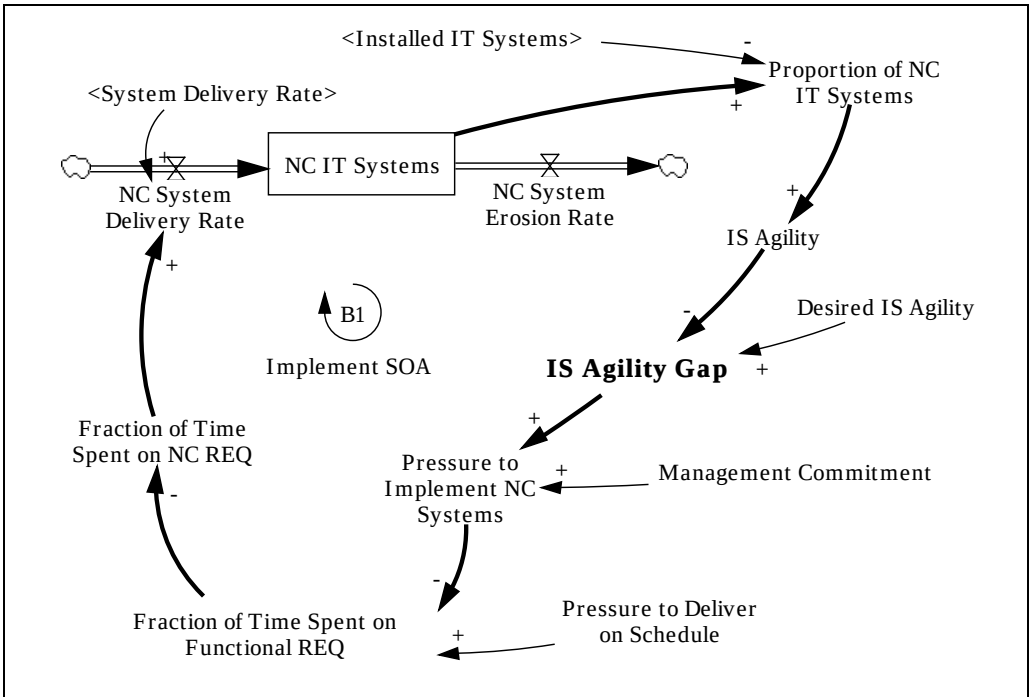


Figure 2. Balancing Loop B1: Implement Net-Centric IT Systems under Pressure

3.1.2 Balancing Loop B2: Work Harder to Deliver on Schedule and Bypass NC REQ

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 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., the entire IT department) can deliver within one unit of time during which the developers may spend part of their work hours on implementing net-centric requirement or attending training sessions about net-centric systems, 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 care the system delivery rate much.

Delivery Rate Gap refers to the difference between Desired Delivery Rate and System Delivery Rate. Unlike IS Agility Gap creating Pressure to Implement NC Systems, the Delivery Rate Gap can create the Pressure to Deliver on Schedule. In case of high pressure to deliver on schedule, developers have to work harder on functional development and try to catch up the delivery schedule. Thus, developers are often forced to spend a large faction of time on implementing the functional requirements and bypass net-centric requirements. The balancing loop B2 captures such a situation in which developers tend to bypass net-centric requirements and work harder to get their development jobs done under the schedule pressure. Balancing loops B2 works to close the Delivery Rate Gap and release the Pressure to Deliver on Schedule.

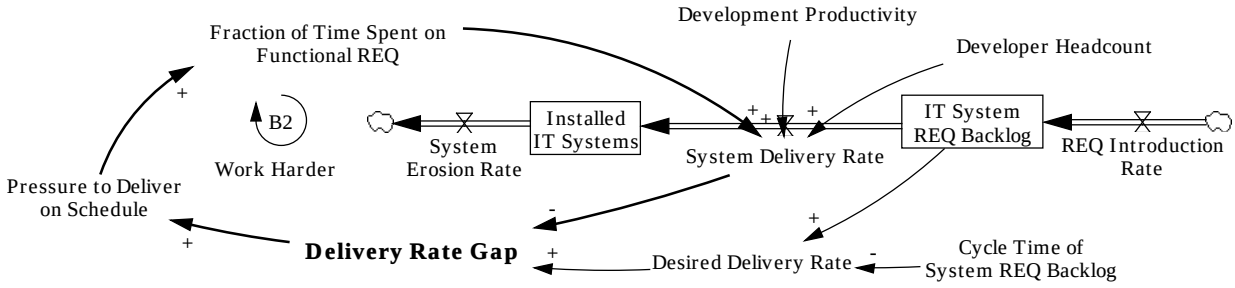


Figure 3. Balancing Loop B2: Work Harder to Deliver on Schedule and Bypass NC REQ

3.1.3 Reinforcing Loop R1: Implement NC Systems through Productivity Increase

Now we turn to the potential benefits of implementation of net-centric systems. Net-centric systems, developed using net-centric design principles (e.g., modularity, loose coupling and standards), are more reusable, interoperable and easier to integrate with IT systems (Choi et al. 2010; Mueller et al. 2010). Therefore, it is easy for developers to make use of the existing reusable net-centric 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, net-centric implementation actually makes the development job easier and allows developers to develop IT systems faster. In other words, net-centric systems increase 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 NC Systems to represent the extent to which the developers' Development Productivity is increased, on average, compared to their Base Development Productivity in the basic situation prior to net-centric implementation. Specifically, the more net-centric systems installed in the organization (i.e., the higher Proportion of NC IT Systems), the more Effectiveness of NC

Systems 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.

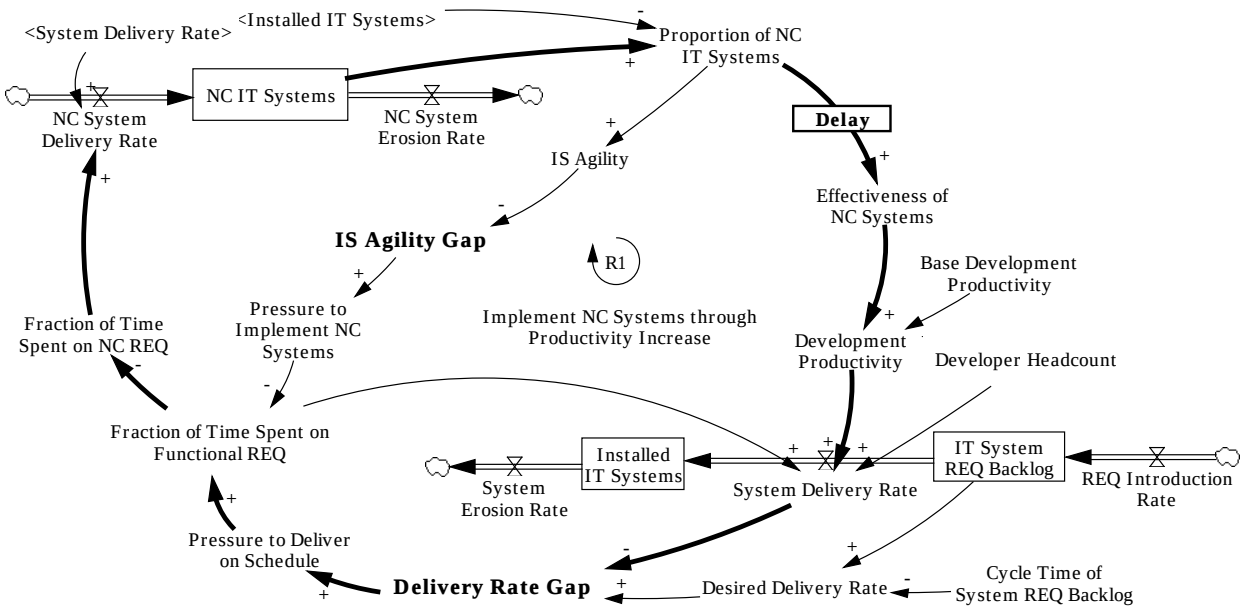


Figure 4. Reinforcing Loop R1: Implement NC Systems through Productivity Increase

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 net-centric IT systems, increasing the NC System Delivery Rate and the number of installed NC IT Systems. Eventually, the Proportion of NC IT Systems will rise further and net-centric systems become more effective, which further enhance the developers' Development Productivity. The entire process becomes a reinforcing loop R1 which is labeled as "Implement NC Systems through Productivity Increase", as shown in Figure 4.

The reinforcing loop R1 can operate as either virtuous or vicious cycles, depending on its current state (Sterman 2000). When R1 operates as virtuous cycles, more net-centric IT systems are implemented (i.e., more penetration of net-centricity in the organization), net-centric systems become more effective and allow developers to invest more time on further implementation. Conversely, when R1 operates as vicious cycles, less net-centric penetration in the organization generates little effectiveness of net-centric systems and contributes little to the developers' development productivity. Under the schedule pressure, developers are likely to shift more time which would have been spent on net-centric implementation to the functional development. Consequently, less service-oriented systems are developed and the net-centric penetration becomes even less.

In order to successfully implement net-centric systems, the reinforcing loop R1 operating as virtuous cycles is preferable. However, this cannot always be the case. Many organizations struggling with their net-centric 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 NC IT Systems and Effectiveness of NC Systems, because it takes time for developers to attend training sessions and introduce new work structure 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.

3.1.4 Reinforcing Loop R2: Motivate Commitment through Perceived Benefits of NC Systems

As discussed, `Effectiveness of NC Systems` increases the developers' `Development Productivity`, compared to their `Base Development Productivity`. Therefore, the benefits and value of net-centricity will be perceived by the organizational participants (e.g., managers and developers). The positive relationship between the effectiveness of innovative IT (particularly the effectiveness of net-centric systems) and the perceived benefits has been largely discussed (Mueller et al. 2010). In particular, Choi et al. (2010) pointed out that net-centric implementation effectiveness is an important determinant of the perceived benefits and value derived from net-centric systems.

Greater perceived benefits of net-centric systems create favorable word of mouth in the organization (Sterman 2000) and generate the additional commitment to net-centric 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 net-centric systems are considered to be internal or endogenous, emphasizing the additional commitment actually results from the results attribute to the use of net-centric IT systems. This is also supported by social cognition theory saying "performance successes strengthen self-beliefs of capability" (Wood and Bandura 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 and Powell 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).

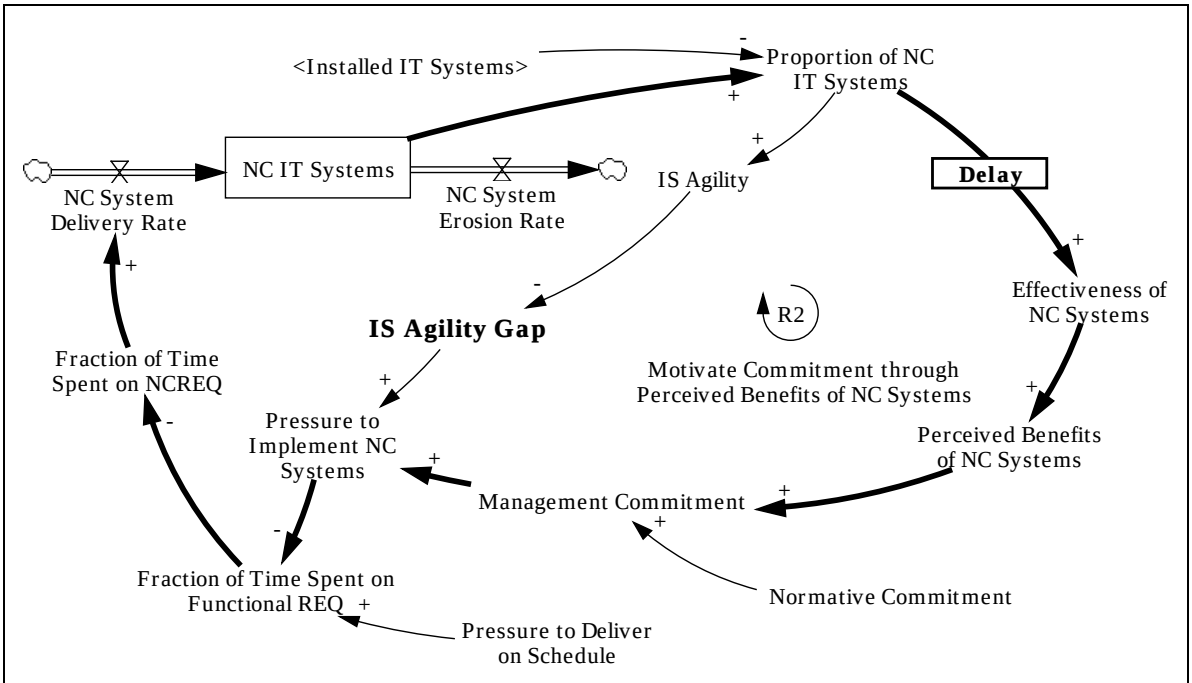


Figure 5. Reinforcing Loop R2: Motivate Commitment through Perceived Benefits of NC Systems

Figure 5 shows the mediating role played by `Management Commitment` between `Normative Commitment` and net-centric implementation. `Management Commitment` generates `Pressure to Implement NC Systems` and forces developers to spend part of their work hours on implementing net-

centric systems, even though they are still under the Pressure to Deliver on Schedule. When net-centric IT systems are implemented and installed over time, net-centric implementation becomes more effective and enhances developers' Development Productivity. With more benefits of net-centric systems perceived by the organization, the effects of 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 NC Systems", 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 and Yellin 2010; Lee et al. 2010). Likewise, there is substantial delay in the reinforcing loop R2.

3.1.5 Reinforcing Loop R3: Acquiring Knowledge of Net-Centricity by Doing

The time delay between Proportion of NC IT Systems and Effectiveness of NC Systems plays an important role in the dynamics of the reinforcing loops R1 and R2. Now we need to explore from where the time delay comes. According to our conversations with managers and developers from the organizations implementing net-centric systems, the Effectiveness of NC Systems depends on two factors. The one factor is Proportion of NC IT Systems, which means net-centricity is more effective with more net-centric IT systems installed in the organization. In other words, when more installed IT systems are service-oriented (e.g., loose-coupling, interoperable), it becomes easier to integrate with those installed IT systems when developers build new IT systems. The other factor is Knowledge of Net-Centricity, the organization's knowledge about how to use and build net-centric IT systems. Without sufficient Knowledge of Net-Centricity, developers may still feel difficult to make use of existing net-centric IT systems or build new IT systems using net-centric design principles, even if Proportion of NC IT Systems is high. Knowledge of Net-Centricity is acquired over time when developers persist to make use of existing net-centric IT systems and adhere to net-centric design principles. In fact, net-centric systems, as a new architectural style (Borges et al. 2004), have certain technical complexity and its implementation process is characterized by a high learning barrier (Choi et al. 2010). In order to realize the effectiveness of net-centric systems, developers have to invest substantial time and energy in learning the new architectural style, e.g., learning net-centric design principles and methodologies (Mueller et al. 2010).

We use the first-order information delay to model the process through which Knowledge of Net-Centricity is acquired, as shown in Figure 6. A key construct Learning Time is used to capture the time delay of the learning process. Learning Time also represents the technical complexity of net-centric systems to the organization. That is, if net-centric systems are more technical complex, the organizational learning time is expected to be longer. Note that Learning Time depends not only on the technical complexity itself, but also on organizational contexts, such as organization size. We expect the Learning Time is longer for a larger organization than that for a small organization.

In Figure 6, there are two key factors that affect the Learning Rate of Knowledge of NC: one is Proportion of NC IT Systems and the other is Perceived Benefits of NC Systems, both of which would increase the learning rate. When more service-oriented IT systems are installed in the organization (i.e., higher Proportion of NC IT Systems), it is easier to use and learn net-centric IT systems. Also, if net-centric systems are perceived to be more beneficial, developers are more willing to adhere to the net-centric design principles in their development work and thus they would acquire knowledge of net-centricity faster.

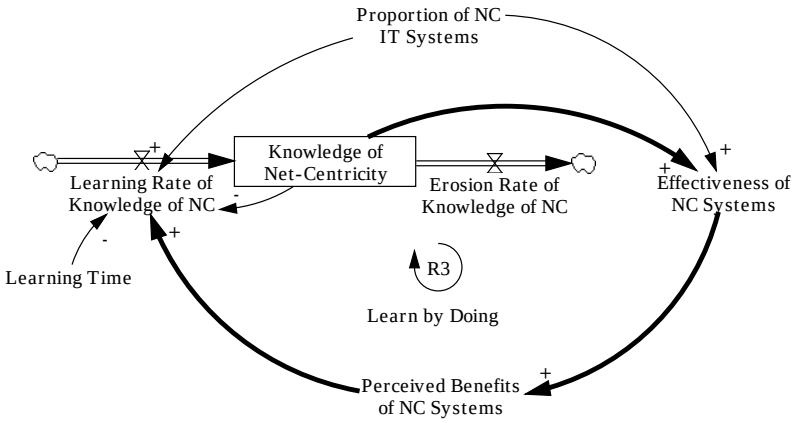


Figure 6. Reinforcing Loop R3: Acquiring Knowledge of Net-Centricity by Doing

As shown in Figure 6, when the `Effectiveness of NC Systems` is realized and more benefits of net-centric systems are perceived by the developers, they are more willing to adhere to net-centric design principles and learn how to use and build net-centric IT systems. The adherence to net-centricity will increase the `Learning Rate of Knowledge of NC` and the organizational `Knowledge of Net-Centricity`, which in turn makes net-centric systems more effective in the organization. The reinforcing loop R3 represents the process through which developers acquire knowledge of net-centricity by doing.

3.2 The Model

We synthesize the key feedback loops in the system dynamic model, as shown in Figure 7.

The balancing loop B1 indicates developers invest part of their work hours to implement net-centric IT 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 implementation of net-centric systems. The balancing loop B1 operates to close the organization's IS agility gap over time, yet management commitment to net-centricity plays the important role in generating the momentum for implementing net-centric systems.

The balancing loop B2 represents the decision that developers would naturally make to work harder and to get the functional development done on schedule. The decision actually shifts part of their work hours to the functional development and under-invests in net-centricity. B2 indicates that 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 operates to close the delivery rate gap and release the schedule pressure. It is worth noting that there is relatively shorter delay within B2 than the delay within B1. In case of a big 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 net-centric 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 tend to overemphasize salient factors when processing attributions is a well-known cognitive and perceptual bias (Tversky and Kahneman 1974). Since bypassing net-centric requirements only hurts the IS agility in the long term, it is difficult for people to attribute to such shortcuts after the substantial delay. But closing the `Delivery Rate Gap` more quickly may probably bring IT managers and developers favorable gains or avoid negative words from other organizational participants (e.g., end users from business units).

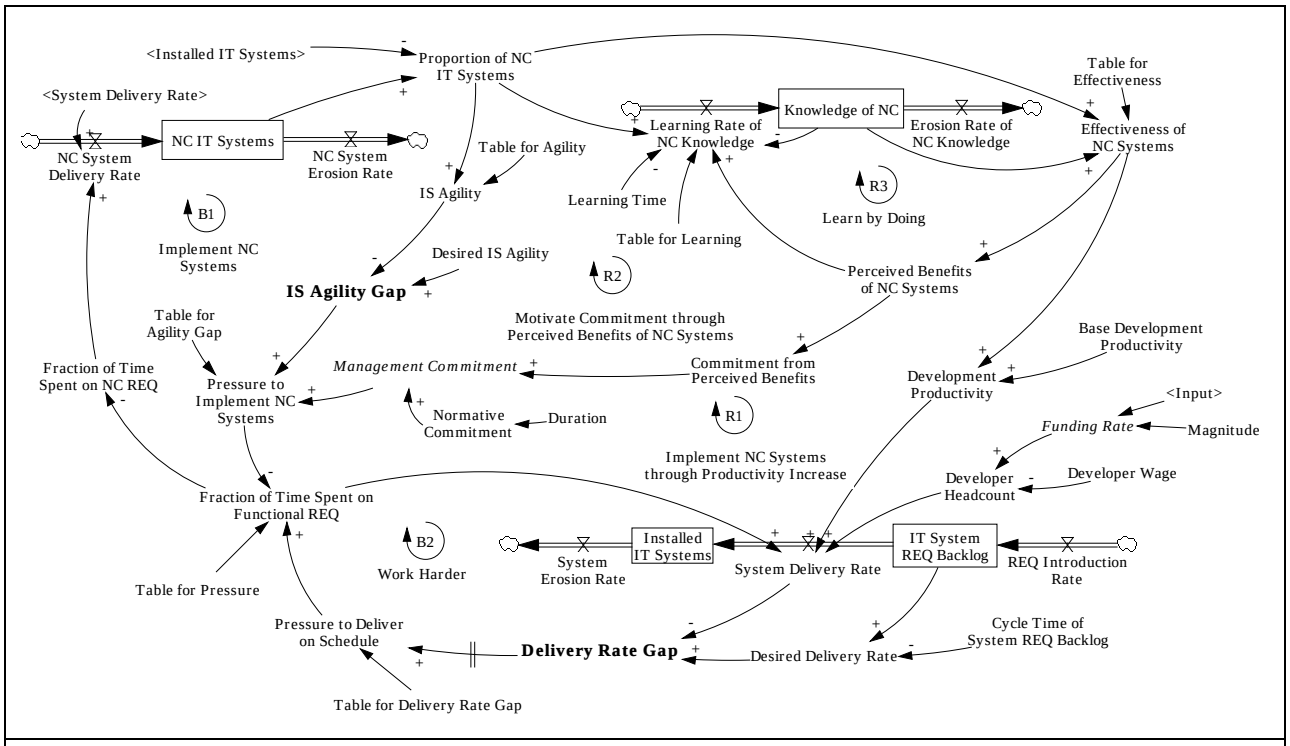


Figure 7. The Model of Implementing Net-Centric Systems

The reinforcing loop R1 represents developers' development productivity increase over time with more and more net-centric 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 implementation of net-centric systems. However, developers cannot immediately acquire the knowledge of net-centricity 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 net-centric systems 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 and Sterman 2001) may last long, but apparently, not every organization or manager would tolerate a much long "worse" period. On the other hand, the substantial delay makes the causal link between the effectiveness of net-centric systems 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 implementing net-centric systems several months or even years ago, because people tend to attribute to more available and salient causes due to cognitive biases (Tversky and Kahneman 1974).

The reinforcing loop R2 represents an organization perceives more benefits from net-centric implementation when the development productivity is being improved over time. The perceived benefits of net-centric systems motivate the management commitment from the internal environment of the organization and promote the net-centric implementation further, resulting in more perceived benefits. Similar to R1, R2 is also subject to the substantial delay between higher net-centric penetration and the perceived benefits of net-centric systems. Yet the substantial delay in R2 has a very important effect that is different from the two effects of the delay in R1. That is, the substantial delay in R2 largely postpones the potential benefits of net-centric systems to be perceived by the organization. Accordingly, organizations that decide to implement net-centric systems have to be patient enough and tolerate a perhaps long period during which little benefits of net-centric systems are perceived, especially at the early stage of implementation of net-centric systems. Thus, the primary part of management commitment to net-centricity 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 net-centricity. The normative commitment has to be maintained for long enough before perceived benefits of net-centric systems 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 and Sambamurthy 1999) and net-centric systems in particular.

The reinforcing loop R3 models the mechanism of the time delay between `Proportion of NC IT Systems` and `Effectiveness of NC Systems`. The time delay is subject to the organizational learning process of knowledge of net-centricity. It takes time for an organization to acquire knowledge of net-centricity through using and building net-centric IT systems and adherence to net-centric design principles.

4 Theory on Organizational Traps

In this section, the theory about organizational traps that impede the net-centric implementation is built before the simulation experiments.

4.1 Organizational Traps

There is substantial delay from the investment in net-centricity to the rise of development productivity and the perceived benefits of net-centric systems. Considering the reinforcing loops R1 and R2, we postulate the learning curve of net-centricity is likely to be very low and flat at the early (but probably long) stage of the implementation and to take off at some point and grow up quickly. As Hau et al. (2008) observed, it is quite common that "*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). Due to the substantial delay, managers and developers are likely to bypass net-centric requirements and/or underinvest in net-centric systems, especially when urgent IT functionalities are requested by end users from business units and intense schedule pressure is created. 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 net-centric requirements]. This is a tactic vs. strategic balance in our organization. And different departments actually have different waiver processes...

Underinvestment in net-centricity postpones the first release of a net-centric application and perhaps allows delivering non-service-oriented systems, leading to less penetration of net-centricity in the organization and in turn less effectiveness of net-centric implementation. As a result, the perceived benefits of net-centric systems are further delayed and negative word of mouth may spread across the organization. For example, the technical complexity of net-centric systems is overemphasized, instability of open standards for net-centricity and inappropriateness of net-centricity 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 that the less learning in using the technology, 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 net-centric systems and thus the low, flat part of the learning curve of net-centricity is actually prolonged, which postpones the perceived benefits of net-centric systems even further. Thus, the organization is trapped in the low, flat and prolonging part of the learning curve of net-centricity and can hardly see it taking off.

Implementation effectiveness trap refers to the situation in which the organization may misperceive the inappropriateness of net-centric systems when net-centric implementation is temporally less effective and perceived benefits of net-centric systems are delayed. It is likely for the organization to falsely conclude that net-centricity is inappropriate to its organizational context, rather than to admit that it is just due to insufficient net-centric penetration in the organization. As a result, developers continue to underinvest in net-centric systems and deliver non-net-centric systems, further undermining the

effectiveness of net-centric implementation. In such a case, the organization is stuck in the trap and can hardly realize the effectiveness of net-centric systems.

Technology learning trap and implementation effectiveness trap are different but intertwined with each other. Because top management attention (and resources) often fades away over time (Kotter 1995), the two traps result in the failures of net-centric implementation efforts of many organizations when the normative commitment of top management to net-centricity fades away.

5 Experimental Simulation

In this and next section, we turn to investigate how to overcome the technology learning trap and the implementation effectiveness trap. Specifically, we focus on exploring how the duration of the normative commitment affects the dynamics of the implementation process. We use experimental simulation to address this question.

In order to operationalize the system dynamics model (see Figure 7), we used 5 table functions in the model. We performed several kinds of model validation to enhance the validity of the model, including model check and unit check. Our model has successfully passed the model validation.

5.1 Base Simulation

In the experimental simulation, we currently focus on two key variables: one is the duration of the normative commitment, and the other is the learning time. In the base model, the duration of the commitment is set to be 12 months and the learning time is 12 months also. This means that the normative commitment lasts for 12 months and after that, top management simply removes the normative commitment. We also assume that a developer becomes more knowledgeable and effective about net-centric methods over time, with 12 months being the average time to be fully competent.

Figure 8 shows the simulation results of the base model. As shown in Figure 8(a), the organization's IS agility grows gradually before the first 12 months. When the normative commitment is removed at the 12th month, the IS agility turns to drop for a while. But at around the 15th month, the IS agility stops to drop and turns to grow again. Eventually, the organization's IS agility grows to equilibrate at a very high level. Thus, this result reveals that it is a good case in which the organization has successfully built up its IS agility through implementing net-centric systems.

As shown in Figure 8(b), most of the developers' work hours are spent on functional requirements when the normative commitment is removed suddenly at the 12th month. But since there may be sufficient internal commitment already motivated through perceived benefits of net-centric systems, the developers would like to spend a certain fraction of their time on implementing net-centric requirements. With more benefits of net-centric systems are perceived, the developers would like to spend more time on net-centric requirements and thus less time on functional requirements. Figure 8(b) shows the fraction of time spent on functional requirements drops gradually after the sudden rise at the 12th month. When the organization builds up sufficient IS agility, developers, with less pressure to implement net-centric systems, can turn to spend more time on their primary jobs, i.e., functional requirements.

Figure 8(c) indicates that when the normative commitment is removed at the 12th month and developers bypass net-centric requirements, the entire system delivery rate actually rises suddenly. But in order to implement net-centric systems and build up the IS agility, the organization does suffer from the "worse-before-before" phenomenon, that is, the system delivery rate drops during the 12th to 20th month. Only after the 20th month, the system delivery rate turns to grow gradually and exceeds the highest point at the 12th month and equilibrates at a higher level. Figure 8(d) shows that the IT system requirement backlog also equilibrates at a certain level.

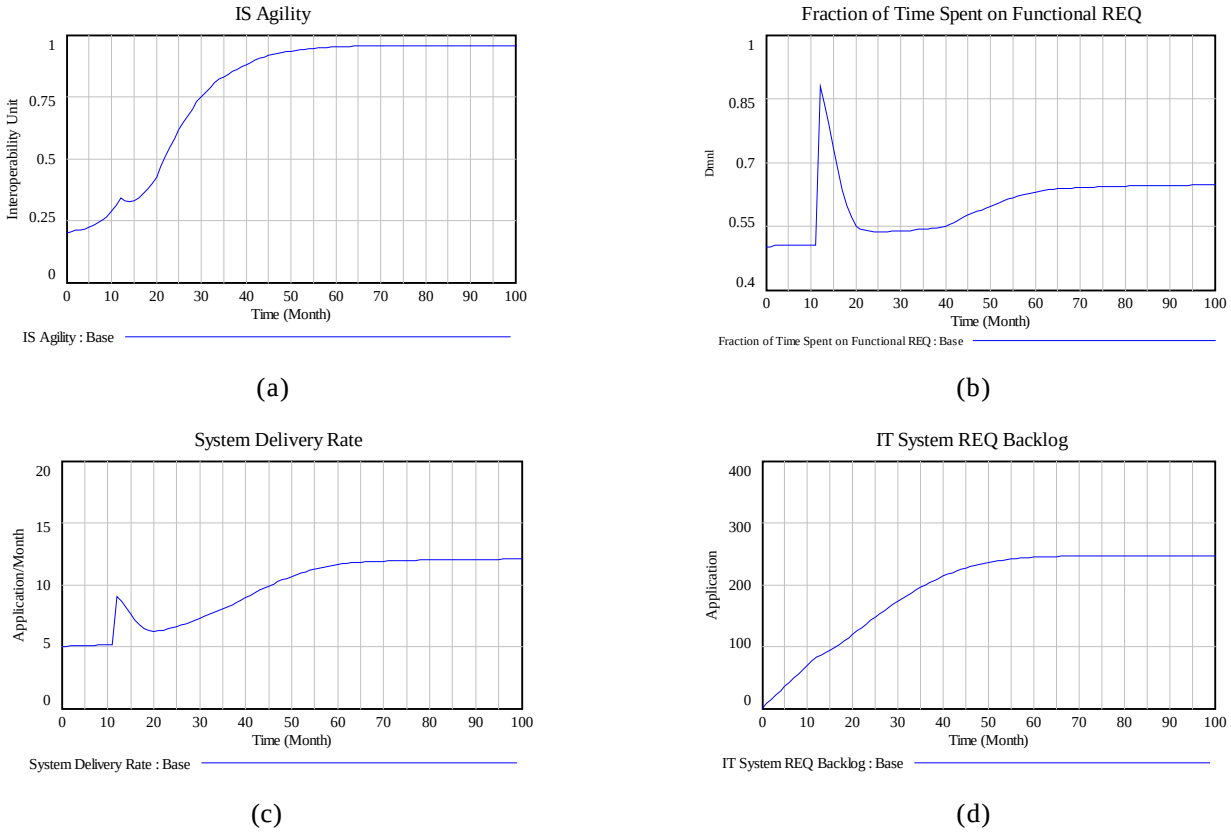
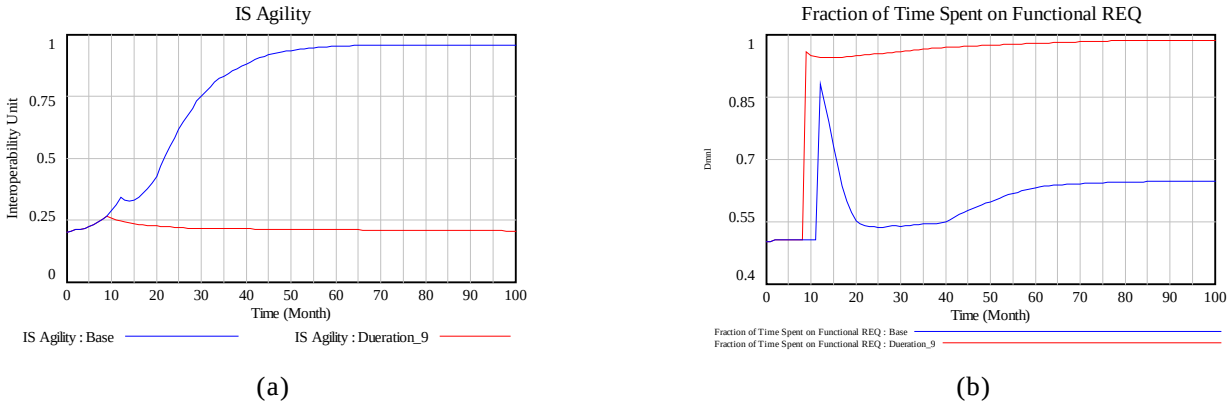


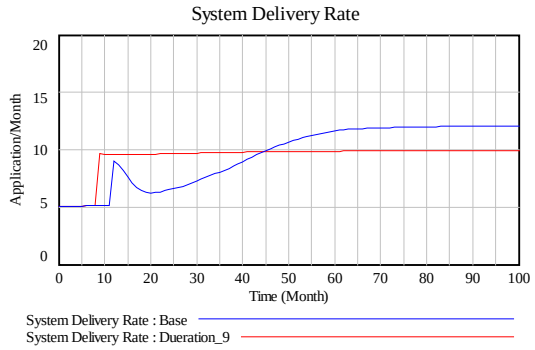
Figure 8. Simulation Results of the Base Model

In sum, the simulation results of the base model demonstrate a case in which the organization removing its normative commitment at the 12th month can still successfully implement net-centric systems and build up the IS agility. Removing normative commitment and bypassing net-centric requirement do increase the system delivery rate. But in order to successfully implement net-centric systems, the organization does suffer from a "worse-before-better" phenomenon.

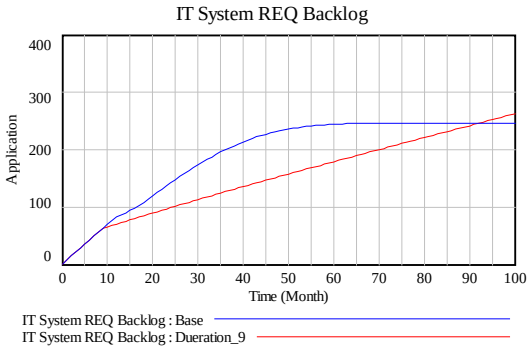
5.2 Simulation of Experiment One

In the first experimental simulation, we change the duration of normative commitment from 12 months to 9 months, holding other factors unchanged. That is, we assume the normative commitment is removed at the 9th month, three months before the base simulation. This experimental simulation allows us to focus on examining how the shorter duration of normative commitment affects the dynamics of the implementation process.





(c)



(d)

Figure 9. Simulation Results of Experiment One

Figure 9 shows the simulation results of Experiment One, comparing to that of the base simulation. As shown in Figure 9(a), the organization's IS agility turns to decrease after the normative commitment is removed at the 9th month and can never grow up. Eventually, the IS agility stays at a low level just as the level from the very beginning. This result reveals that little effectiveness of net-centric systems has been realized when the normative commitment is removed at the 9th month. Without realizing the effectiveness of net-centric systems, little benefit of net-centricity is perceived in the organization, resulting in little internal commitment on net-centricity. In this case, the organization simply stops implementing net-centric systems. As shown in Figure 9(b), the developers spend most of their work hours on functional requirements and thus little time on net-centric requirements.

Figure 9(c) indicates that the system delivery rate jumps to a higher level when the normative commitment is removed at the 9th month and developers spend little time on net-centric requirements. But the system delivery rate remains at that level which is actually lower than the case of the base simulation. This means the organization actually loses the opportunity of implementing net-centric systems to increase its system delivery rate. Also, the organization loses the opportunity of implementing net-centric systems to enhance its IS agility, as shown in Figure 9(a).

Figure 9(d) shows that the IT system requirement backlog always grows up, which means the IT department cannot complete the system requirements requested by the business units. In contrast, the IT department can accomplish the system requirements in the case of base simulation.

5.3 Simulation of Experiment Two

In the second experimental simulation, we change the learning time from 12 months to 16 months, holding the duration of normative commitment as 12 months and other factors unchanged. That is, we assume the learning time for a certain organization to acquire knowledge of net-centricity is 16 months, rather than 12 months, and the normative commitment is removed at the 12th month as the base simulation. This experimental simulation allows us to focus on examining how the longer learning time, in terms of a given duration of normative commitment, affects the dynamics of the implementation process.

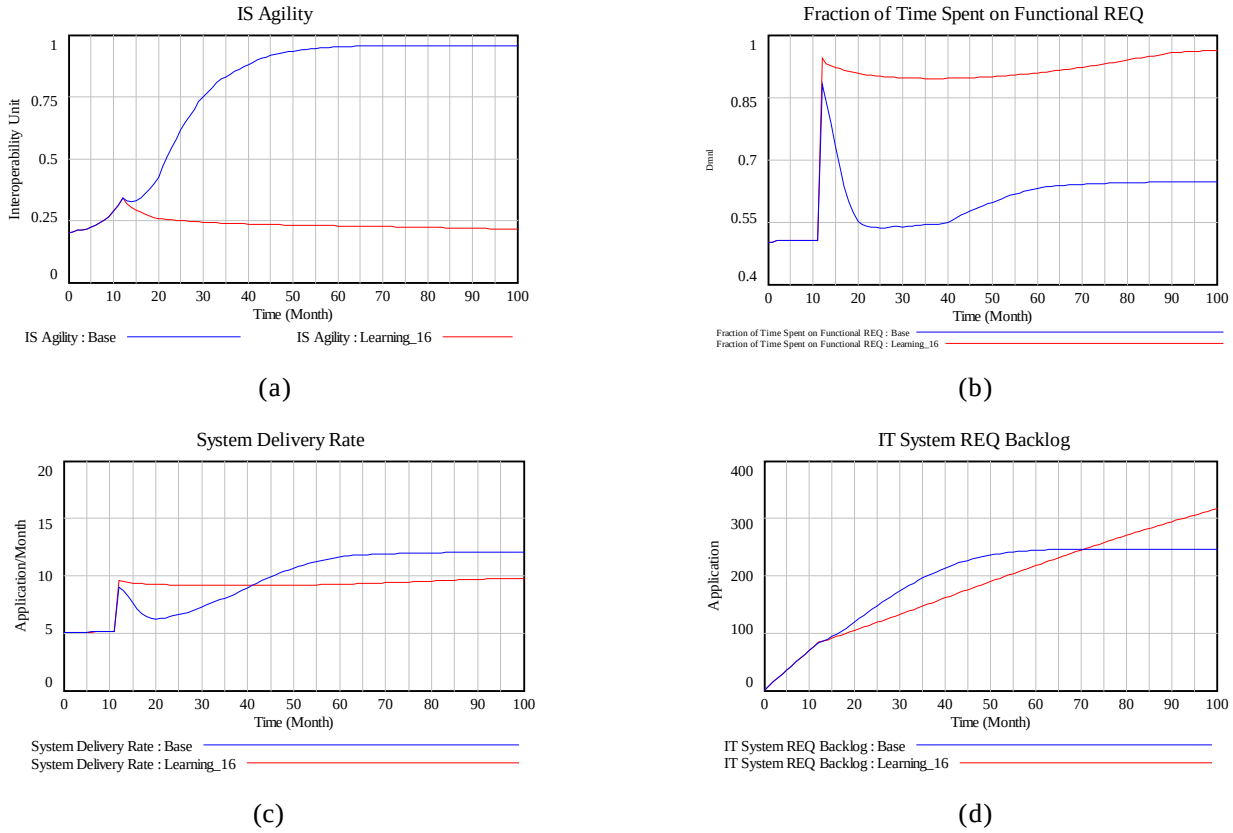


Figure 10. Simulation Results of Experiment Two

Figure 10 shows the simulation results of Experiment Two, comparing to that of the base simulation. As shown in Figure 10(a), the organization's IS agility turns to decrease after the normative commitment is removed at the 12th month and can never grow up. Eventually, the IS agility gradually falls to the low level just as that from the very beginning. This result reveals that little effectiveness of net-centric systems has been realized when the normative commitment is removed at the 12th month. Without realizing the effectiveness of net-centric systems, little benefit of net-centric systems is perceived in the organization, resulting in little internal commitment on net-centricity. In this case, the organization simply stops implementing net-centric systems. As shown in Figure 10(b), the developers spend most of their work hours on functional requirements and thus little time on net-centric requirements.

Figure 10(c) indicates that the system delivery rate jumps to a higher level when the normative commitment is removed at the 12th month and developers spend little time on net-centric requirements. But the system delivery rate remains at that level which is actually lower than the case of the base simulation. This means the organization actually loses the opportunity of implementing net-centric systems to increase its system delivery rate. Also, the organization loses the opportunity of implementing net-centric systems to enhance its IS agility, as shown in Figure 10(a). Figure 10(d) shows that the IT system requirement backlog always grows up, which means the IT department cannot complete the system requirements requested by the business units. In contrast, the IT department can accomplish the system requirements in the case of base simulation.

6 Theory on Normative Commitment

According to the experimental simulation, we can see that the dynamics of Experiment One and Experiment Two are very similar, in both of which the organization fails in the net-centric implementation and loses the opportunity to enhance its IS agility.

Experiment One tells that given the learning time 12 months, lasting the normative commitment for 12 months is enough to succeed in the net-centric implementation, while lasting the normative commitment

for 9 months is not. If the normative commitment is removed at the 9th month (only 3 months early), the organization does not realize much effectiveness of net-centric systems and thus perceives little benefits of net-centric systems. With little perceived benefits of net-centric systems, the organization participants (developers and managers) are not convinced that net-centricity is appropriate to their organization. Negative word of mouth about net-centric systems will be generated and spread in the organization, resulting in less pressure to implement net-centric systems. On the other hand, developers' development productivity is not improved and developers still face large pressure to deliver the functional requirements on schedule. Consequently, developers will shift the time spent on net-centric requirements to the development of functional requirements and develop non-net-centric IT systems in the organization. This results in less proportion of net-centric IT systems and even less effectiveness of net-centric systems. In addition, developers do not adhere to net-centric design principles due to less perceived benefits of net-centric systems, lowering the learning rate of knowledge of net-centricity. In this case, all the three reinforcing loops R1~R3 operate as the vicious cycles and the net-centric implementation fails.

Experiment Two tells that it is not enough for an organization with 16-month learning time to last the normative commitment for 12 months, while it is enough for an organization with 12-month learning time. If the learning time of an organization is 16 months rather than 12 months (for example, a larger organization or long-standing may have longer learning time, compared to a small or new organization), the organization needs to put longer normative commitment. In other words, the minimum duration of the normative commitment is probably positively associated with the learning time of the organization to acquire knowledge of net-centricity. The organization has to last the normative commitment long enough in order to acquire sufficient knowledge of net-centricity and realize sufficient effectiveness of net-centric systems. Otherwise, if the normative commitment fades away at the midway, the three reinforcing loops R1~R3 operate as the vicious cycles and the organization is likely to fail in the net-centric implementation.

Therefore, the theory herein is about the relationship between normative commitment and organization learning time. We summarize the theory in the following proposition:

Proposition: *Certis parabis, organizations with longer learning time need to last the normative commitment longer in order to acquire sufficient knowledge of net-centricity and realize the effectiveness of net-centric systems, succeeding in the net-centric implementation.*

7 Discussion

In this research, we assume to a large extent that net-centric systems, once implemented well in an organization, can enhance the organization's IS agility and developers' development productivity at least in the long term (Choi et al. 2010; Hau et al. 2008; Mueller et al. 2010; Yoon and Carter 2007). Despite best practices of net-centric implementation offered by vendors, consultants and leading organizations (Krafzig et al. 2005), many organizations encountered organizational and human challenges in their net-centric implementation efforts (Fricko 2006; Luthria and Rabhi 2008). In fact, prior research suggests 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). For example, 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 and Wagner 2003). Although ERP as a monolithic IS architecture is very different from net-centric systems and thus has different organizational implications, prior research on ERP implementation leads us to accommodate human agency in this research and to examine the important role that human agency plays during the implementation process of net-centric systems.

Net-centric implementation, like any organization-wide IS implementation, requires organizations to invest substantial resources upfront before potential benefits are perceived by organizational participants, known as the "worse-before-better" phenomenon. That is, there are substantial delays between the implementation investment and the perceived benefits. During the "worse" period of the implementation, different organizational participants 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 participants are likely to underinvest in net-centric implementation and thus the dynamics can get stuck in two intertwined traps: technology learning trap and implementation effectiveness trap. Once stuck in the traps, it would be difficult for the organizational participants to correctly attribute to the vicious cycle of the dynamics of the implementation process. According to their misattribution, the subsequent reaction of the organizational participants 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, net-centric systems) 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 (Orlikowski 1992).

Managers who make the decision of net-centric 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 net-centric implementation. Patient and consistent decision-makings about the tradeoff between short-term performance drop and potential long-term benefits and the tradeoff between local project needs and organization-level net-centric 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 net-centric implementation (Joachim et al. 2011; Varadan et al. 2008). The traps discussed here can be of even greater significance in defense and intelligence systems where unique operational and security considerations require extensive, often multi-year, customization of commercial SOA software before benefits can begin to be seen.

References

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

Association for Enterprise Integration (AFEI). 2008. Industry Recommendations for DoD Acquisition of Information Services and SOA Systems. SOA Acquisition Working Group, AFEI Executive Forum on Business Change, Arlington, Va., July 7."

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

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

Bieberstein, N., Bose, S., Walker, L., and Lynch, A. 2005. "Impact of Service-Oriented Architecture on Enterprise Systems, Organizational Structures, and Individuals," *IBM Systems Journal* (44:4), pp 691-708.

Boh, W.F., and Yellin, D.M. 2010. "Enablers and Benefits of Implementing Service-Oriented Architecture: An Empirical Investigation," *International Journal of Information Technology and Management* (9:1), pp 3-29.

Borges, B., Holley, K., and Arsanjani, A. 2004. "Service-Oriented Architecture: Components and Modeling Can Make the Difference," *Web Services Journal* (9:1), pp 34-38.

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., and Rackham, G. 2005. "Impact of Service Orientation at the Business Level," *IBM Systems Journal* (44:4), pp 653-668.

Choi, J., Nazareth, D.L., and Jain, H.K. 2010. "Implementing Service-Oriented Architecture in Organizations," *Journal of Management Information Systems* (26:4), pp 253-286.

Croom, C. 2006. "Service-Oriented Architectures in Net-Centric Operations," *CROSSTALK The Journal of Defense Software Engineering*, July 2006, pp 13-15.

DiMaggio, P.J., and Powell, W.W. 1983. "The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields," *American sociological review* (48:2), pp 147-160.

Fricko, A. 2006. "Soas Require Culture Change and Service Reuse," *Business Communications Review* (36:5), p 58.

Georgantzas, N.C., and Katsamakas, E.G. 2008. "Information Systems Research with System Dynamics," *System Dynamics Review* (24:3), pp 247-264.

Grant, D., Hall, R., Wailes, N., and Wright, C. 2006. "The False Promise of Technological Determinism: The Case of Enterprise Resource Planning Systems," *New Technology, Work and Employment* (21:1), pp 2-15.

Hau, T., Ebert, N., Hochstein, A., and Brenner, W. 2008. "Where to Start with Soa: Criteria for Selecting Soa Projects," *Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS 2008)*, Waikoloa, Big Island, Hawaii: IEEE Computer Society, pp. 314-322.

Joachim, N., Beimborn, D., and Weitzel, T. 2011. "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)*, Kauai, Hawaii USA: IEEE Computer Society, pp. 1-10.

Kanaracus, C. 2011. "Soa Is Alive and Well, Forrester Says." Computer World.

Kotter, J. 1995. "Leading Change: Why Transformation Efforts Fail," *Harvard Business Review* (73), pp 59-67.

Krafzig, D., Banke, K., and Slama, D. 2005. *Enterprise Soa: Service-Oriented Architecture Best Practices*. Prentice Hall PTR.

Law, J., and Urry, J. 2004. "Enacting the Social," *Economy and Society* (33:3), pp 390-410.

Lee, J.H., Shim, H.J., and Kim, K.K. 2010. "Critical Success Factors in Soa Implementation: An Exploratory Study," *Information Systems Management* (27:2), pp 123-145.

Liang, H., Saraf, N., Hu, Q., and Xue, Y. 2007. "Assimilation of Enterprise Systems: The Effect of Institutional Pressures and the Mediating Role of Top Management," *Management Information Systems Quarterly* (31:1), p 6.

Luthria, H., and Rabhi, F. 2008. "Organizational Constraints to Realizing Business Value from Service Oriented Architectures: An Empirical Study of Financial Service Institutions," *Service-Oriented Computing - ICSOC 2008*), pp 256-270.

Luthria, H., and Rabhi, F. 2009a. "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), pp 39-56.

Luthria, H., and Rabhi, F. 2009b. "Using Service Oriented Computing for Competitive Advantage," *Proceedings of Americas Conference on Information Systems (AMCIS 2009)*, p. 140.

McKendrick, J. 2011. "So Far, Soa Failures Are Few and Far between, Survey Says." ZDNet.

Moitra, D., and Ganesh, J. 2005. "Web Services and Flexible Business Processes: Towards the Adaptive Enterprise," *Information & Management* (42:7), pp 921-933.

Mueller, B., Viering, G., Legner, C., and Riempp, G. 2010. "Understanding the Economic Potential of Service-Oriented Architecture," *Journal of Management Information Systems* (26:4), pp 145-180.

Orlikowski, W.J. 1992. "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Organization Science* (3:3), pp 398-427.

Pfeffer, J. 1997. "New Directions for Organization Theory:: Problems and Prospects,").

Pfeffer, J., and Sutton, R.I. 2000. *The Knowing-Doing Gap: How Smart Companies Turn Knowledge into Action*. Cambridge, MA, USA: Harvard Business School Press.

Repenning, N.P. 2002. "A Simulation-Based Approach to Understanding the Dynamics of Innovation Implementation," *Organization Science* (13:2), pp 109-127.

Repenning, N.P., and Sterman, J.D. 2001. "Nobody Ever Gets Credit for Fixing Problems That Never Happened," *California management review* (43:4), pp 64-88.

Repenning, N.P., and Sterman, J.D. 2002. "Capability Traps and Self-Confirming Attribution Errors in the Dynamics of Process Improvement," *Administrative Science Quarterly* (47:2), pp 265-295.

Rosenberg, N. 1982. *Inside the Black Box: Technology and Economics*. Cambridge: Cambridge University Press.

Rudolph, J.W., Morrison, J.B., and Carroll, J.S. 2009. "The Dynamics of Action-Oriented Problem Solving: Linking Interpretation and Choice," *The Academy of Management Review (AMR)* (34:4), pp 733-756.

Ryan, G.W., and Bernard, H.R. 2000. "Data Management and Analysis Methods," in: *Handbook of Qualitative Research*, Y.S.L. Norman K. Denzin (ed.). Sage, CA, USA: Citeseer, pp. 769-801.

Scott, S.V., and Wagner, E.L. 2003. "Networks, Negotiations, and New Times: The Implementation of Enterprise Resource Planning into an Academic Administration," *Information and organization* (13:4), pp 285-313.

Sher, P.J., and Lee, V.C. 2004. "Information Technology as a Facilitator for Enhancing Dynamic Capabilities through Knowledge Management," *Information & Management* (41:8), pp 933-945.

Śliwaa, J. and Amanowicz, M. 2011. "Success Factors for SOA Implementation in Network Centric Environment," *Journal of Telecommunications and Information Technology* 1/2011, pp 43-53. http://www.nit.eu/czasopisma/JTIT/2011/1/43.pdf

Smith, R. 2008. "A Simpler Approach to Soa." Information Week

Sterman, J.D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World with Cd-Rom*. Irwin/McGraw-Hill.

Tversky, A., and Kahneman, D. 1974. "Judgment under Uncertainty: Heuristics and Biases," *Science* (185:4157), p 1124.

Varadan, R., Channabasavaiah, K., Simpson, S., Holley, K., and Allam, A. 2008. "Increasing Business Flexibility and Soa Adoption through Effective Soa Governance," *IBM Systems Journal* (47:3), pp 473-488.

Volkoff, O., Strong, D.M., and Elmes, M.B. 2007. "Technological Embeddedness and Organizational Change," *Organization Science* (18:5), pp 832-848.

Vroom, V.H. 1964. *Work and Motivation*. New York: Wiley.

Wheelwright, S.C., and Clark, K.B. 1995. *Leading Product Development*. Free Press.

Wood, R., and Bandura, A. 1989. "Social Cognitive Theory of Organizational Management," *The Academy of Management Review* (14:3), pp 361-384.

Yoon, T., and Carter, P. 2007. "Investigating the Antecedents and Benefits of Soa Implementation: A Multi-Case Study Approach," *AMCIS 2007 Proceedings*), pp 195-205.