

---

# Understanding the Dynamics of Service-Oriented Architecture Implementation

XITONG LI AND STUART E. MADNICK

XITONG LI is an assistant professor in the Department of Operations Management and Information Technology, HEC Paris, France. He received his Ph.D. in management from the MIT Sloan School of Management, and his Ph.D. in engineering from Tsinghua University. His recent research interests include the economic and social impacts of using online data/information, and innovative technologies using online data and services. His research has appeared, or is forthcoming, in *Journal of Management Information Systems*, *ACM Transactions on Internet Technology*, *IEEE Communications Magazine*, *IEEE Transactions on Automation Science and Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics*, and other leading journals and conference proceedings. He won the Best Paper award at the Forty-Sixth Hawaii International Conference on System Sciences (HICSS).

STUART E. MADNICK is the John Norris Maguire Professor of Information Technologies at the MIT Sloan School of Management and a Professor of Engineering systems at the MIT School of Engineering. He headed MIT's Information Technologies Group for more than 20 years. He has degrees in electrical engineering, management, and computer science from MIT. His current research interests include disparate distributed information systems, database technology, software project management, social media, strategic use of information technology, and cybersecurity. He is author/coauthor of over 380 books, articles, and reports, including *Operating Systems* (McGraw-Hill, 1974) and *Dynamics of Software Development* (Prentice Hall, 1991). He has been active in industry, including projects such as IBM's VM/370 operating system and Lockheed's DIALOG information retrieval system.

**ABSTRACT:** Despite the potential benefits, many organizations have failed in service-oriented architecture (SOA) implementation projects. Prior research often used a variance perspective and neglected to explore the complex interactions and timing dependencies between the critical success factors. This study adopts a process perspective to capture the dynamics while providing a new explanation for the mixed outcomes of SOA implementation. We develop a system dynamics model

---

The authors are grateful to the editor-in-chief, the associate editor, and four anonymous reviewers who provided constructive comments and suggestions on earlier versions of this article.

and use simulation analysis to demonstrate the phenomenon of “tipping point.” That is, under certain conditions, even a small reduction in the duration of normative commitment can dramatically reverse, from success to failure, the outcome of an SOA implementation. The simulation results also suggest that (1) the *duration* of normative commitment can play a more critical role than the *strength*, and (2) the minimal duration of normative commitment for a successful SOA implementation is associated positively with the information delay of organizational learning of SOA knowledge. Finally, we discuss the theoretical causes and organizational traps associated with SOA implementation to help IT managers make better decisions about their implementation projects.

**KEY WORDS AND PHRASES:** normative commitment, organizational traps, service-oriented architecture (SOA), system dynamics, tipping point.

---

It is important for an organization to be able to adapt its IT systems and quickly respond to changing business conditions. This ability is referred to as the organization’s information systems (IS) agility [11, 12], and it is considered a key facilitator for enhancing a firm’s dynamic capabilities and competitive advantages. However, most traditional IT systems, such as enterprise resource planning (ERP) systems, are designed using a monolithic architecture and built as integrated sets of software modules linked to a common database [45]. The investment needed for such a complex system makes it difficult, expensive, and time-consuming for organizations to change their IT systems [11]. To address this problem, practitioners and researchers have advocated service-oriented architecture (SOA) as a new computing paradigm for building IT systems [4, 39]. Service-oriented architecture refers to an architecture style that supports loosely coupled interoperable services to enable business flexibility and agility [7]. It 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 [13].<sup>1</sup>

Prior research on SOA claimed several potential benefits and advantages [10, 35, 53], including enhanced IS agility and improved IT-business alignment [5, 11, 12, 35], and many organizations invested in SOA expecting to reap those potential benefits. A large-scale survey conducted in 2008 by Ross et al. found that the average SOA budget in 2007 was \$4.9 million, while the SOA budgets of some firms were close to \$11 million [47]. A Forrester report [20] revealed that 71 percent of the enterprises surveyed were already using SOA by the end of 2011. However, other studies reported mixed outcomes after implementing SOA. For example, a study by the Burton Group showed that 50 percent of the investigated 20 companies considered their SOA initiatives as a complete failure, whereas only 20 percent of them viewed their SOA initiatives successful [36]. In contrast, a survey from CA Technologies found that 92 percent of SOA initiatives met or exceeded the objectives of business units [32].

To explain these mixed outcomes, recent relevant literature has focused on the critical success factors that affect SOA implementation. The underlying research

often cites factors that might facilitate or impede an organization's ability to receive the intended benefits of SOA. For example, Luthria and Rabhi identified six key factors that influence the organizational adoption of SOA [29]. Boh and Yellin examined two organizational factors that were potentially critical to ensuring the success of SOA implementations: top management support and centralization of IT decision making [6]. Lee et al. presented 20 factors based on their review of 34 SOA studies and 22 interviews [24]. Joachim et al. found that an effective IT governance mechanism was needed to achieve the benefits of SOA [19].

With this study, we add to the literature on this topic and provide a new explanation for the mixed outcomes of SOA implementation by exploring how interactions between critical factors affect the dynamics of SOA implementation. In particular, we investigate how two key factors (the duration of top management's normative commitment and the information delay of organizational learning of SOA knowledge) interact with other critical factors over time to affect the success or failure of SOA implementation.

In this study, we adopt the process perspective [33] and develop a system dynamics (SD) model that is built on Repenning and Sterman's theory of capability traps [42, 44], expanded to incorporate many specifics of SOA implementation. Using the SD model, we conduct simulations that examine the dynamics of SOA implementation in different scenarios. We also demonstrate the phenomenon of the "tipping point."

To achieve a successful SOA implementation, top management must continue their normative commitment for a duration long enough so that the organization can perceive the benefits that in turn motivate internal organizational commitment to SOA. We show that under certain conditions, even a small reduction in the duration of normative commitment can dramatically reverse the outcome of SOA implementation from success to failure. The simulation analysis also reveals that (1) the *duration* of normative commitment can play a more critical role than the *strength* of normative commitment for the success of SOA implementation, and (2) the minimum duration of normative commitment needed for a successful SOA implementation has a positive association with the information delay of organizational learning of SOA knowledge. Finally, we discuss how the dynamics of SOA implementation can result in some organizational traps as well as the structural and organizational causes of the traps.

Our study makes two important contributions to the IS literature. First, our study provides a new explanation, from the process perspective, for the mixed outcomes of SOA implementation. Second, using simulation analysis based on the SD model, the study demonstrates the phenomenon of tipping point and the critical role of the duration of normative commitment. Whereas prior research often discussed top management support in general [18], our study is perhaps the first research that separates the duration and the strength of top management support, finding that duration plays a more critical role than strength for a successful SOA implementation. Also, it is important for IT managers to understand the positive association

between the minimal duration of normative commitment for a successful SOA implementation and the information delay of organizational learning of SOA knowledge.

## Research Methodology

---

In this section, we discuss two different research perspectives: variance and process. Then we describe our semistructured interviews and how we develop the SD model in the study.

### Research Perspectives: Variance versus Process

Mohr identifies two different perspectives of organizational research: variance and process [33]. Variance research aims to explain variations in outcome variables by associating them with predictors and necessary and sufficient conditions; process research seeks to “explain outcomes by examining sequences of events over time” [45, p. 19]. To date, prior research on SOA has largely used the variance perspective and focused on either the potential benefits of SOA [10, 35, 53] or the critical success factors of SOA implementation [6, 19, 24, 29]. However, earlier research neglected to explore the complex interactions and timing dependencies between the critical factors.

Unlike prior research, we adopted the process perspective in this study, making use of SD modeling [51] as the main research methodology. SD modeling is a powerful tool for organizational research [44, 48, 49, 51] and has been widely used in IS research [9, 11, 12, 15, 41]. Compared to other organizational research methodologies, the strength of SD modeling lies in its exploration of feedback loops, timing delays, the nonlinearity associated with the process of organizational change [31, 38], and its ability to generate insights into the dynamics of SOA implementation. An operationalized SD model enabled us to conduct rich simulation analyses and to examine the dynamics of SOA implementation in different scenarios.

### Semistructured Interviews

We utilized semistructured interviews and discussions with subject-matter experts, in combination with an extensive literature review, to build the SD model. The interview sample consisted of 10 North American organizations that have implemented, or are in the process of implementing, SOA. These include: EMC, Raytheon, Oracle, SAP, MIT Lincoln Lab, and several U.S. government agencies. While some of them implemented SOA only within their own organizations, others like Raytheon and MIT Lincoln Lab, not only implemented SOA in their own organizations but helped other organizations with their SOA implementation.

## SD Model Development

We followed standard procedure for SD model development [1, 51]. Specifically, we took the following four steps:

- First, we conducted an initial series of 10 preliminary interviews with 14 IT managers and developers; some of the interviews were conducted as group discussions with multiple participants. Interviews lasted between 60 and 90 minutes. After the research purpose was explained, informants described their professional backgrounds. Then they were asked to share their experiences about SOA implementation and to suggest the key challenges and factors that facilitated or impeded their SOA implementation efforts. In this step, we focused on determining the interactions and timing dependencies between the various factors. To build more confidence in our findings and devise holistic and multifaceted explanations of the change [40], follow-up conversations by telephone or e-mail, an intensive documentation review, and direct onsite observations were also conducted. The result of this step was a preliminary SD model.
- Second, we conducted an extensive literature review after the preliminary SD model was built. The literature review covered prior research on organizational studies [33, 38, 45], system dynamics [44, 48, 49, 51], and information systems [1], including SOA [11, 17, 35] and ERP [45]. The literature review filled in gaps in the preliminary model, leading to a more detailed and refined SD model. Each causal link in the model was justified by supportive evidence from our interviews or the literature review.
- Third, we conducted another set of 15 interviews with 22 interviewees. The interviewees included a range of subject-matter experts, including chief information officers (CIOs), mid-level IT managers, and IT developers. These interviews provided evidence that reflected the perspectives of various organizational levels. Using the model generated from the previous steps as a starting point, the interviews and discussions in this step were more concentrated. Further, various techniques for confirmation were used, such as follow-up conversations, intensive documentation review, and onsite observation. As is typical with qualitative research [27, 40], we triangulated across multiple sources and proceeded iteratively between the interviews and building the SD model.
- Finally, we conducted a number of interviews after a simulation analysis was carried out based on the built SD model. The inputs from those interviews provided face validity of the SD model and enhanced our confidence in the interpretation of the simulation results.

Note that during the interviews and model development, some IT managers and developers spent two to four hours with us, going through every variable and causal link in the SD model and the simulation results. Their input and feedback enabled us to build the SD model largely grounded on their empirical experiences. By doing this, the face validity of the model was strengthened, especially because it was validated directly with the practitioners.

## System Dynamics Model

The most important aspect of an SD model is its structure, which we built through interviews and the literature review. In this section, we briefly introduce the key feedback loops of the SD model as they apply to SOA implementation. Throughout this text, the names of variables are shown in italic font. Details of the model can be found in Appendix A. Key assumptions about the model and its operationalization are documented in Appendix B. We operationalize the SD model and present the equations, parameters, and table functions in the online supplemental appendix.<sup>2</sup>

Figure 1 illustrates the overall SD model. It has two balancing loops (B1 and B2). B1 indicates that IT developers invest only part of their work hours in implementing service-oriented systems. Two sources of pressure affect how IT developers decide to allocate their time: (1) pressure to deliver the functionalities of IT systems on schedule, and (2) pressure to conduct an SOA implementation. The B1 loop relates to the latter pressure by closing the *IS Agility Gap*, with *Management Commitment* playing an important role in generating momentum for the SOA implementation. The B2 loop represents the decision that IT developers make to focus on the functional development on schedule and to bypass SOA implementation. B2 closes *Delivery Rate Gap* and reduces *Pressure to Deliver on Schedule*.

The SD model has three reinforcing loops (R1, R2, R3). R1 represents that *Development Productivity* of IT developers increases when more service-oriented systems are installed, thereby reducing *Pressure to Deliver on Schedule*. Under less schedule pressure, IT developers are more likely to invest more of their work hours in SOA implementation. However, they cannot immediately acquire the needed knowledge of SOA because of its technical complexity [11], and thus cannot increase *Development Productivity* in a short period due to the substantial delay in R1.

R2 explicitly models the mechanism of the time delay between *Proportion of Service-Oriented Systems* and *Effectiveness of SOA*. The time delay comes from the organizational learning process of using knowledge of SOA, and represents the fact that it takes time for an organization to acquire SOA knowledge through building and using service-oriented systems and adhering to SOA design principles.

R3 represents the situation in which an organization perceives more benefits from SOA implementation when *Development Productivity* increases over time. *Management Commitment* to SOA initially comes from *Normative Commitment* from top managers (such as exogenous coercive or mimetic pressures). Meanwhile, the perceived benefits of SOA motivate internal commitment and promote SOA implementation further, producing more perceived benefits. In R3 there is substantial delay between higher SOA penetration and the rise of *Development Productivity*.



## Model Validation and Simulation Analysis

According to the frequently cited textbook,<sup>3</sup> *Business Dynamics*, by John Sterman [51], there are 12 tests for assessing dynamic models [51, pp. 859–861]. The software we used to build the SD model is Vensim PLE for Windows, Version 6.0. Using the capabilities of Vensim PLE, we conducted all of the tests that applied to this study, such as structural assessment, dimensional consistency, parameter assessment, and sensitivity analysis. Our SD model successfully passed all the tests. Some of our interviews with subject-matter experts were conducted specifically to determine appropriate values for the model variables and to validate the behaviors of the model. Feedback from the experts enhanced the face validity of model behaviors. For example, in the base scenario presented in the next section, the model demonstrated the “worse-before-better” phenomenon during a successful SOA implementation, which is consistent with the experiences of the experts. In the simulation analysis, we focused on two key variables: the *Duration* of normative commitment and *Learning Time of Using Knowledge* about SOA.

### Base Scenario: A Successful SOA Implementation

In the base scenario, *Duration* of normative commitment is set as 14 months and *Learning Time of Using Knowledge* is set at 12 months. This means that top management continues its normative commitment for 14 months (the strength of normative commitment stays constant) and then completely removes it at the 14th month.<sup>4</sup> By setting the learning time as 12, we examined a situation in which it took 12 months for the organization to accumulate sufficient knowledge and experience in the use of service-oriented systems to be fully effective. Note that the base scenario we examined was just one of the possible scenarios in which the SOA implementation was successful. In a simulation analysis, we can examine the dynamics of SOA implementation with different parameter settings.

Figure 2 compares the simulation results of the base scenario with that of an alternative scenario, which we will explain in the next section. The curves numbered with “1” in Figure 2 are the simulation results from the base scenario. Figure 2a shows that the organization’s IS agility grows quickly from the beginning. When the normative commitment is removed at the 14th month, IS agility continues to grow and eventually stabilizes at a high level. This is because the service-oriented systems that are built under normative commitment in the first 14 months enable the organization to perceive sufficient benefits from SOA. This is evident in Figure 2b, which shows the perceived benefits of SOA starting to rise quickly before the 14th month, and internal commitment remains motivated to continue developing SOA. After normative commitment is removed, it is internal commitment that continues to drive the momentum of SOA implementation.

Figure 2c shows that developers start to change how they spend their work hours (about 70 percent) on functional requirements when normative commitment is removed at the 14th month: the IT developers put higher priority on developing



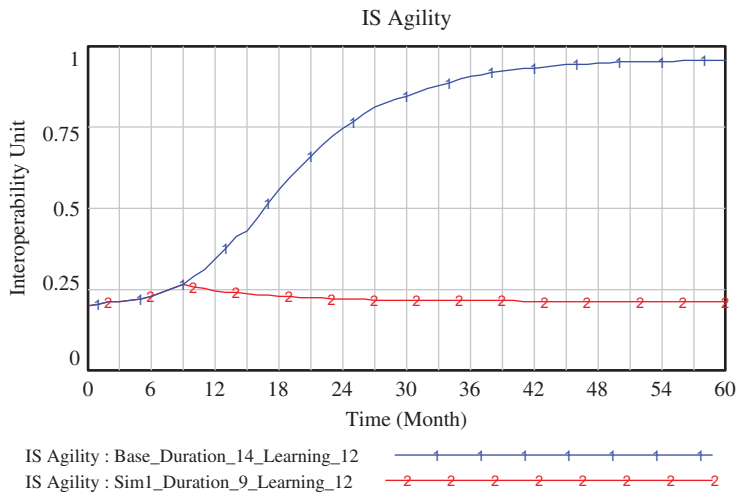


Figure 2a. Simulation vs. Base Model: IS Ability

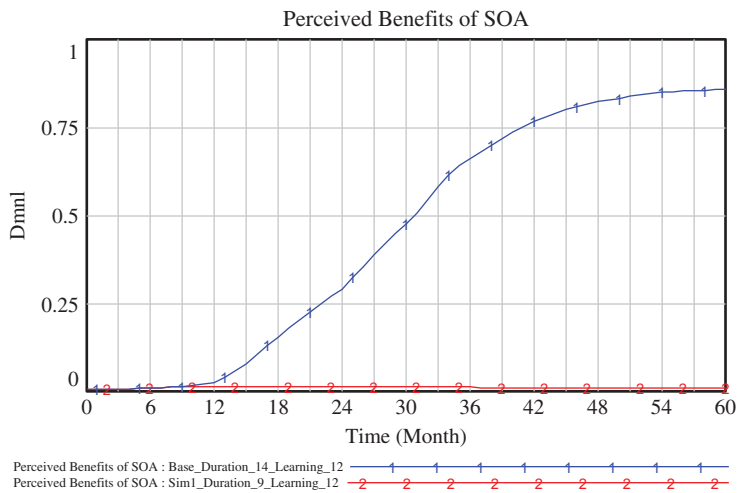


Figure 2b. Simulation vs. Base Model: Perceived Benefits of SOA

and delivering the functional requirements when they face only schedule pressure. But because internal commitment has been motivated already, IT developers still spend about 30 percent of their work hours on implementing SOA. Then the percentage of their work hours spent on SOA rapidly increases back to about 50 percent. Finally, after the organization builds up sufficient IS agility and efficiency using SOA around the 30th month, IT developers can turn to spending more time on functional developments. In the stabilized long term, developers spend about 65 percent of their work hours on functional development and 35 percent on learning and using SOA. Figure 2d shows that when the normative commitment is removed

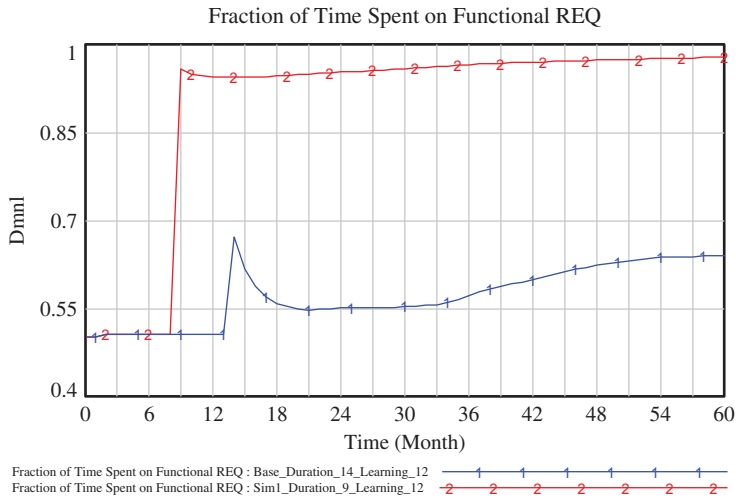


Figure 2c. Simulation vs. Base Model: Fraction of Time Spent on Functional REQ

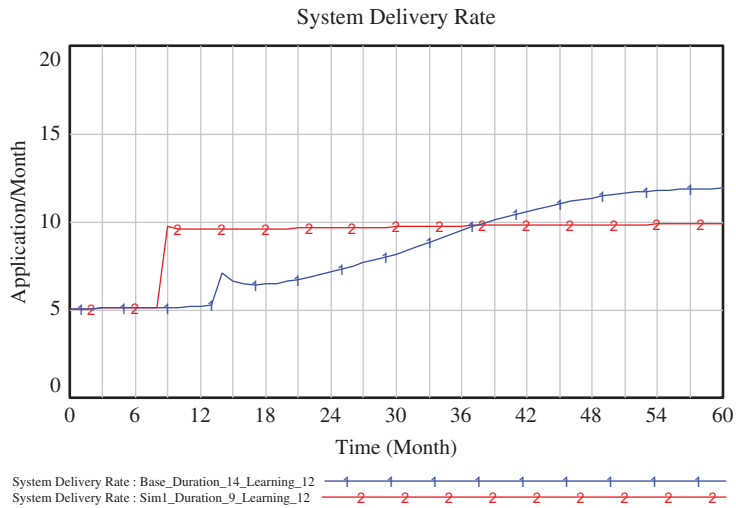


Figure 2d. Simulation vs. Base Model: System Delivery Rate

at the 14th month, the total system delivery rate rises suddenly. But to implement SOA, the organization suffers from the classic “worse-before-better” phenomenon, and the system delivery rate drops during the 14th to 18th month. Only after the 18th month does the system delivery rate stop dropping and begin to grow. Eventually, the system delivery rate stabilizes at a higher level.

The base scenario demonstrates that an organization that completely removes its normative commitment at the 14th month can still succeed with its SOA implementation. The organization does suffer from a “worse-before-better”

phenomenon during the SOA implementation process, which is consistent with the experiences of the experts we interviewed, thus validating the model's structural behavior. We use the base scenario as the benchmark in the simulation analysis.

### Simulation 1: Shorter Duration of Normative Commitment

Simulation 1 examines how a reduction in the duration of normative commitment affects the dynamics of SOA implementation. In this simulation, we reduce the duration of normative commitment from 14 to 9 months, but hold all other factors unchanged. That is, we examine a situation in which top management removes its normative commitment at the 9th month, perhaps because top management is less patient or some other matter arises that takes away their earlier attention.

The lines numbered with “2” in [Figures 2a–d](#) are the results of Simulation 1 as compared to the base scenario (lines numbered with “1”). [Figure 2a](#) shows that the organization's IS agility turns to “decrease” after the normative commitment is removed and never grows higher after that. Eventually, IS agility stabilizes at a low level, meaning the organization fails to enhance its IS agility. [Figure 2b](#) shows that the organization never perceives the benefits of SOA. Without any perceived benefits of SOA, the organization has difficulty generating sufficient internal commitment. As a result, the organization gives up SOA implementation when normative commitment is removed. Without normative commitment or internal commitment, IT developers spend most of their work hours on functional requirements, as shown in [Figure 2c](#). [Figure 2d](#) indicates that the system delivery rate jumps to a higher level when the normative commitment is removed at the 9th month, but remains at that level, which is actually lower than the level eventually attained in the base scenario.

Interestingly, during the period between the 9th and 37th month, the system delivery rate in Simulation 1 is higher than that of the base scenarios, which reflects the “worse-before-better” pattern; that is, the organization suffers a “worse” period before it achieves the “better” level of system delivery rate.

### Simulation 2: Longer Learning Time

Simulation 2 examines how the information delay affects the dynamics of SOA implementation. In this simulation, we increase the learning time from 12 to 20 months, hold the duration of normative commitment at 14, and leave all other factors unchanged. This means that we examine a situation in which it takes 20 months for the organization to accumulate sufficient knowledge of using service-oriented systems to be fully effective. This may occur because (as in this case) the organization size is larger, or the SOA technical complexity is higher, or the organization does not have as effective training procedures as they have in the base scenario.

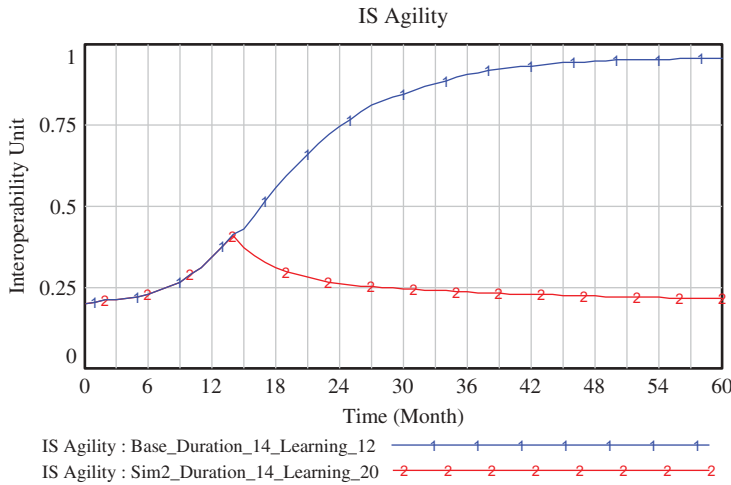


Figure 3. Simulation 2 vs. Base Scenario

The lines numbered with “2” in Figure 3 show the results of Simulation 2 when comparing it to the base scenario (lines numbered with “1”). Like Simulation 1, the SOA implementation in Simulation 2 fails. The results suggest that if an organization has a longer learning time for acquiring SOA knowledge (longer time delay in the reinforcing loops R1, R2, and R3), the organization needs to continue its normative commitment longer in order to succeed with the SOA implementation.

### Simulation 3: Duration vs. Strength of Normative Commitment

In Simulation 3, we fix the learning time at 12 months and compare two different cases. In the first case, the organization’s normative commitment starts at 0.8 and stays constant until the 9th month. After that, the organization removes normative commitment completely. In the second case, the organization’s normative commitment starts at 0.8 but the strength decreases with a constant rate, reaching 0 at the 10th month.

Figure 4a illustrates the normative commitments in the two cases: STEP function, line numbered “1” versus RAMP function, line numbered “2.” During the first 9 months, the strength of normative commitment in the first case is larger than in the second case. Accordingly, the accumulative normative commitment in the first case ( $0.8 \times 9 = 7.2$ ) is almost twice that in the second case ( $0.8 \times 10/2 = 4$ ). Interestingly, Figure 4b shows that the SOA implementation in the first case fails, whereas the SOA implementation in the second case succeeds (the IS agility eventually reaches and stabilizes at a high level close to 1.0) because of the additional normative commitment in the 10th month. The results suggest that the duration of normative commitment plays a more critical role in the success of SOA implementation than the strength of normative commitment.

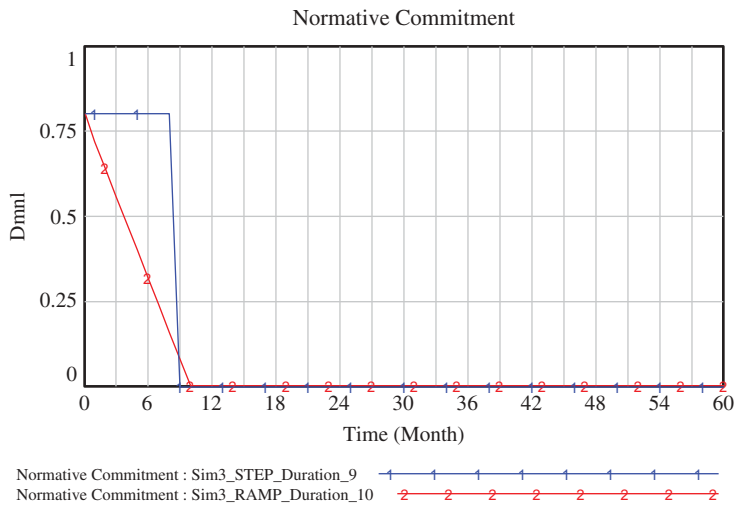


Figure 4a. Duration vs. Strength of Normative Commitment

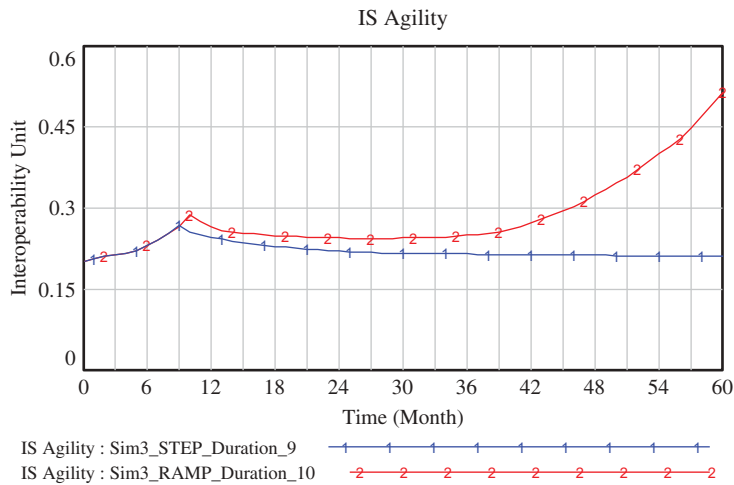


Figure 4b. Duration vs. Strength of Normative Commitment: IS Agility

## Tipping Point and Organizational Traps

### Discovering the Tipping Point

In this simulation, we fix learning time at 12 months and examine how different durations of normative commitment affect the success or failure of SOA implementation. Figure 5 shows the different dynamics that occur when the duration of normative commitment is set at 14, 12, 10, or 9. When normative commitment continues for 10 months or longer, the organization's IS agility grows

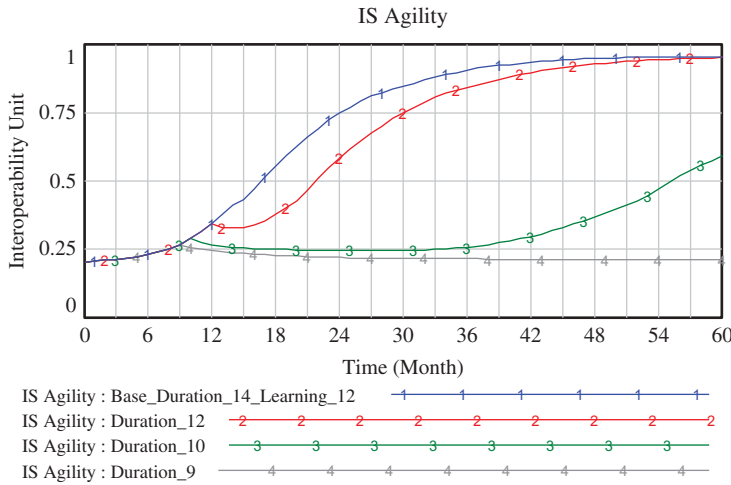


Figure 5. Demonstration of “Tipping Point”

upward and stabilizes at a high level close to 1.0. But the organization fails to enhance its IS agility if normative commitment continues for 9 months or less. The striking finding is that a seemingly small difference in the duration of normative commitment can lead to the ultimate success or failure of SOA implementation.

There appears to be a tipping point for normative commitment that lies between 9 and 10 months. As long as the organization continues its normative commitment for a period beyond the tipping point, IS agility will continue to increase. If normative commitment is discontinued before the tipping point, the entire SOA implementation fails. Figure 5 also shows that, beyond the tipping point, a longer duration of normative commitment allows IS agility to reach and stabilize at a high level faster.

It is important to understand that the precise tipping point depends on various parameters in the model, which vary from organization to organization. The representative values we used were suggested by the experts we interviewed. However, the existence of a tipping point (at whatever point it may occur) is intrinsically determined by the structure of the SOA implementation process, specifically, the two balancing loops (B1, B2) and three reinforcing loops (R1–R3).

As a result of this simulation, we learned that managers have to be patient and continue their normative commitment further beyond the tipping point because a seemingly small reduction in the duration of normative commitment can dramatically reverse the outcome of an SOA implementation from success to failure.

While Figure 5 demonstrates the existence of a tipping point, we note that it is difficult (if not impossible) to determine precisely where the tipping point will be because it depends on the parameters of the social dynamic systems. Figure 5 shows that the tipping point for the normative commitment lies somewhere between 9 and 10 months when the learning time is 12 months. If the learning time is set at

20 months, further simulations show that the tipping point would lie somewhere between 14 and 15 months.

The findings from further simulation analysis suggest that the minimum duration of normative commitment needed for a successful SOA implementation is positively associated with the information delay of organizational learning. Because the duration of normative commitment is more critical to the success of SOA implementation than the strength of normative commitment (see the results from Simulation 3), it is important for managers to understand the positive association between minimal duration of normative commitment and information delay of organizational learning.

## Organizational Traps

Due to the substantial delay in the reinforcing loops (R1–R3), IT developers are likely to ignore or bypass SOA requirements and underinvest in SOA, especially when urgent IT functions are being requested by end users. Based on our interviews, bypassing SOA requirements may even be institutionalized in some organizations. Underinvestment in SOA leads to delivering IT systems that are not service-oriented, which ends in less SOA penetration and less-effective SOA. As a result, the expected benefits of SOA are substantially delayed and negative word of mouth may spread through the organization. In such a situation, the organization is likely to be caught in two different traps: a technology learning trap and an implementation effectiveness trap.

The technology learning trap refers to a situation in which the technology is perceived to be more difficult and complex because it is not well understood due to insufficient learning. A technology learning trap indicates the existence of a cycle of “learning by doing” [2] or, more specifically, “learning by using” [46]. Consequently, developers continue to underinvest in SOA, and to believe that the SOA technology is complex, so their response is to postpone the learning process even further.

The implementation effectiveness trap refers to a situation in which SOA is believed to be inappropriate if an SOA implementation is temporarily less effective and/or the expected benefits of SOA are delayed. Such misperceptions may lead the organization to conclude that SOA is inappropriate in the current organizational context, instead of understanding that the temporary ineffectiveness is due to insufficient SOA penetration in the organization. As a result, the reinforcing loop continues, with developers still underinvesting in SOA, and delivering IT systems that are not service-oriented. These actions further undermine the effectiveness of SOA.

In addition, the two organizational traps are different but intertwined with each other. Because the top management team may change [23] or their strategic attention (and resources) fade away, both traps result in the failure of SOA implementation efforts.

These two organizational traps emerged as a result of our simulations and during the course of our interviews. For example, the chief scientist in a large software vendor complained about the complexity of SOA technologies:

I think the technologies of SOA are very complex and hard to understand. . . . There are so many dependencies among the parts of different services. . . . We used to develop an SOA system. But when we demonstrated it, it failed to work! Because the dependencies are so complex and there is little documentation, we could not find the problems. . . . The contexts of using SOA are also too complex . . . so many configurations . . .

When asked how much time and training the organization had invested in learning SOA, the chief scientist explained that they did not have sufficient experts to train their IT developers, which indicates insufficient learning about SOA technology. By contrast, interviewees who had good experiences with their SOA implementation efforts did not complain about the complexity of SOA technologies and often made favorable comments about SOA's effectiveness .

### Causes of Organizational Traps

We found several plausible causes that may contribute to creating organizational traps.

First, there is a fundamental tension in the trade-off between short-term performance drop and potential long-term benefits. The problem is indicated by the occurrence of the "worse-before-better" pattern [42], which we demonstrated in our simulations. As Hau et al. 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 trade-off between long-term benefits and short-term local needs [17]. Moreover, short-term performance drop and potential long-term benefits are often associated with different groups of organizational actors [18].

Investing resources (e.g., time and budget) in SOA disrupts the organization's normal operation [25]. Developers need to devote a substantial portion of their work hours to SOA implementation, which decreases their responsiveness to requests for IT functionalities in the early stage of the implementation process. In short, to improve IS agility, the organization must be willing to sacrifice expected performance for the short term. System delivery rate is a more salient and immediate performance indicator, and the urgency of IT functionality requests often emphasize this salience. In comparison, IS agility is an organization-level, less-salient, and less-immediate performance indicator with uncertainty due to the substantial anticipated delay before IS agility becomes apparent. Because of this cognitive and perceptual bias, people tend to overemphasize salient factors when processing attributions [52], and organizational actors are likely to overweigh delivery rate gap and ignore improving IS agility.

In some circumstances, sacrificing the long-term benefits of SOA seems inevitable. 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, SOA, as a complex architectural style, manifests the agency of organizational actors (e.g., IT managers and developers) to control their interactions with the SOA technology and its characteristics. Organizational actors have considerable flexibility in design, implementation, use, and interpretation of service-oriented systems, which indicates interpretive flexibility of technological artifacts [37]. Many interviewees noted that there were different ways to deliver the same system functionality when they decided on specific implementation strategies. Interpretive flexibility allows the SOA technology to be appropriated in diverse ways by actors in different organizations or by the same actors in different contexts [30]. It is possible that organizational actors may inappropriately implement or use SOA and misinterpret its effectiveness. An IT manager from a large software vendor commented on the challenge of handling the 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 if 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.

Third, while organizations that enjoy the perceived benefits of SOA early are more dedicated to their SOA implementation, organizations where SOA falls short of their expectations are likely to underinvest in their SOA implementation, resulting in an even worse situation. They will be less likely to attribute the poor results of SOA implementation to their own past actions (i.e., underinvestment in SOA). The complex dynamics of the SOA implementation process bias the actors' interpretation and perception of the appropriateness of SOA, thus triggering a vicious and negative cycle of SOA implementation.

Fourth, when the benefits of SOA do not meet immediate expectations, managers and developers tend to blame "technical complexity" and "inappropriateness of SOA," rather than recognizing that their frustration or skepticism may stem from having bypassed earlier SOA requirements. Organizational actors in an SOA implementation are more likely to perceive the technology itself as ineffective. The tendency of humans to blame technology rather than themselves is widely observed and documented in the literature [3]. Brown and Brudney wrote: "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" [8, p. 423]. After observing the

adoption process of a faculty educational technology in a university, Moser found: “If technology was involved, however, faculty were quick to blame the failure on the technology and abandon newly acquired teaching practice and technology use” [34, p. 67]. When stuck in the technology learning trap and/or the implementation effectiveness trap, blaming the complexity and inappropriateness of the technology gives organizational actors an excuse for their past actions of underinvestment in SOA. The vicious cycle of declining SOA implementation reinforces their excuse and misperception. Such misinterpretation reveals the organizational actors’ self-confirming attribution error [44]. Lorenzi and Riley [28] 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. ([28], p. 200)

Finally, while Figure 5 demonstrates the existence of a tipping point associated with SOA implementation, it is often difficult to predict precisely where the tipping point will occur. Within the parameters of our simulation model, Figure 5 shows the minimal duration of normative commitment for a successful SOA implementation. However, when the implementation is under way, it is difficult, if not impossible, for top management to know ahead of time the precise duration of normative commitment needed so that the entire SOA implementation will succeed. Consequently, impatient managers are likely to shorten the duration of normative commitment before the SOA implementation reaches or exceeds the tipping point. Managers may think the strength of normative commitment could be a substitute for the duration of normative commitment. However, the results from Simulation 3 show that such a substitution does not work. *Duration* of normative commitment plays a unique and critical role in the success of SOA implementation—far more than *strength*.

## Conclusion

Our findings generate important research implications. Earlier IS literature indicates that human agency often plays an important role in IS implementation, for example, in ERP implementation [37, 54]. In the IS literature on ERP, Grant et al. write: “[K]ey stakeholders in the ERP implementation process adopted different discourses” and highlighted the role of their discourses in the social shaping of ERP implementation [16, p. 13]. Scott and Wagner point out that the success or failure of an ERP implementation is actually highly situated and relates to the negotiations between actor networks surrounding the implementation process [50]. Although ERP as a monolithic IS architecture is distinct from SOA, prior research on ERP inspires us to accommodate human agency in this study and to examine the important role that

human agency plays during an SOA implementation, for example, how long top management would continue its normative commitment for SOA.

Next, our study suggests that the technology learning trap and implementation effectiveness trap result not only from the characteristics of the technology (in this case, SOA) and the inherent structure of the implementation process (i.e., the reinforcing feedback loops), but also from the dynamic interactions between human agency and the technology implementation process. The misattribution by organizational actors when they become stuck in the traps reflects another form of self-confirming error; yet it is different from that discussed in Repenning and Sterman's works on process improvement in manufacturing [44].

Finally, SOA implementation requires organizations to invest substantial resources upfront before benefits are recognized by organizational actors—a phenomenon known as “worse-before-better” [42]. During the “worse” period, different organizational actors make judgments, from their own perspectives, about what becomes “worse” to them, how much “worse” it will be, and how long “worse” will continue. Impatient organizational actors are likely to underinvest in SOA implementation, thus causing the entire project to become stuck in the technology learning and implementation effectiveness traps. Once stuck, it is difficult for the organizational actors to correctly recognize and attribute the vicious cycle of the dynamics of the implementation process. Due to that misattribution, subsequent reactions of organizational actors further exacerbate the vicious cycle. Our findings from this study suggest that IT managers take a long-term, global view about their SOA implementation projects. IT managers need to understand the potential organizational traps and prepare appropriate strategies to deal with the challenges. For example, if the organizational size is larger or the organization does not have sufficient SOA expertise, IT managers need to anticipate that the time delay before the benefits of SOA implementation are realized would be longer and therefore have to continue normative commitment for a longer period of time, rather than devoting stronger normative commitment in a short period. Thus, by understanding these causes of organizational traps, IT managers can make better decisions about their implementation projects.

## Supplemental Material

Supplemental data for this article can be accessed at the publisher's website at <http://dx.doi.org/10.1080/07421222.2015.1063284>

## NOTES

---

1. We must emphasize the specificity and difference of SOA compared to traditional ERP systems. While ERP implementation often focuses on consolidating multiple information technology (IT) systems across an organization, SOA implementation focuses on improving an organization's IS agility and flexibility. Because of this difference, the focus of this study is different from prior studies on ERP implementation.

2. For the online appendix, see <http://web.mit.edu/smadnick/www/JMIS/OnlineAppendix.pdf>. (This will be replaced by a permanent link.)
3. About 8,000 citations, according to Google Scholar. See [https://scholar.google.com/scholar?cluster=7563512649899599529&hl=en&as\\_sdt=0,22&scioldt=0,22](https://scholar.google.com/scholar?cluster=7563512649899599529&hl=en&as_sdt=0,22&scioldt=0,22).
4. This phenomenon of top management losing interest in efforts and reducing normative commitment was mentioned in many of our interviews. Sometimes normative commitment slowly faded away, and at other times there was an abrupt distraction (e.g., a change in management). To focus on the role played by the duration of normative commitment, in many of the simulations we assume that top management completely removes normative commitment at a certain time point, and we use a STEP function to model normative commitment. This simplified assumption may not be realistic in some cases, but it also does not compromise the key insights from our findings. As a robustness check, we ran the model assuming that top management reduces normative commitment at a constant rate (using a RAMP function), and we obtained qualitatively similar results.
5. For measurement consideration, a developer's development productivity can be measured as the unit of software components, modules, or functional features.

## REFERENCES

1. Abdel-Hamid, T., and Madnick, S.E. *Software Project Dynamics: An Integrated Approach*. Upper Saddle River, NJ: Prentice Hall, 1991.
2. Arrow, K.J. The economic implications of learning by doing. *Review of Economic Studies*, 29, 3 (1962), 155–173.
3. Avital, M., and Vandenbosch, B. SAP implementation at Metalica: An organizational drama in two acts. *Journal of Information Technology*, 15, 3 (2001), 183–194.
4. Bardhan, I.R.; Demirkan, H.; Kannan, P.K.; Kauffman, R.J.; and Sougstad, R. An interdisciplinary perspective on IT services management and service science. *Journal of Management Information Systems*, 26, 4 (2010), 13–64.
5. 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), 691–708.
6. 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), 3–29.
7. Borges, B.; Holley, K.; and Arsanjani, A. Service-oriented architecture: Components and modeling can make the difference. *Web Services Journal*, 9, 1 (2004), 34–38.
8. Brown, M.M., and Brudney, J.L. Public sector information technology initiatives. *Administration and Society*, 30, 4 (1998), 421–442.
9. Cao, L.; Ramesh, B.; and Abdel-Hamid, T. Modeling dynamics in agile software development. *ACM Transactions on Management Information Systems*, 1, 1 (2010), article 5.
10. 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), 653–668.
11. Choi, J.; Nazareth, D.L.; and Jain, H.K. Implementing service-oriented architecture in organizations. *Journal of Management Information Systems*, 26, 4 (2010), 253–286.
12. Choi, J.; Nazareth, D.L.; and Jain, H.K. The impact of SOA implementation on IT-business alignment: A system dynamics approach. *ACM Transactions on Management Information Systems*, 4, 1 (2013), 1–22.
13. Datta, A.; Dutta, K.; Liang, Q.; and VanderMeer, D. SOA performance enhancement through XML fragment caching. *Information Systems Research*, 23, 2 (2012), 505–535.
14. 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), 147–160.
15. Georgantzis, N.C., and Katsamakas, E.G. Information systems research with system dynamics. *System Dynamics Review*, 24, 3 (2008), 247–264.

16. 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), 2–15.
17. 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*. Waikoloa: IEEE Computer Society, 2008, 314–322.
18. Jiang, J.J.; Chang, J.Y.; Chen, H.-G.; Wang, E.T.; and Klein, G. Achieving IT program goals with integrative conflict management. *Journal of Management Information Systems*, 31, 1 (2014), 79–106.
19. Joachim, N.; Beimborn, D.; and Weitzel, T. The influence of SOA governance mechanisms on IT flexibility and service reuse. *Journal of Strategic Information Systems*, 22, 1 (2013), 86–101.
20. Kanaracus, C. SOA is alive and well, Forrester says. *Computer World*, March 23, 2011.
21. Keith, M.; Demirkan, H.; and Goul, M. Service-oriented methodology for systems development. *Journal of Management Information Systems*, 30, 1 (2013), 227–260.
22. Kotter, J. Leading change: Why transformation efforts fail. *Harvard Business Review*, 73, 2 (1995), 59–67.
23. Lee, J.C., and Myers, M.D. Dominant actors, political agendas, and strategic shifts over time: A critical ethnography of an enterprise systems implementation. *Journal of Strategic Information Systems*, 13, 4 (2004), 355–374.
24. 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), 123–145.
25. Lee, J.S.; Keil, M.; and Kasi, V. The effect of an initial budget and schedule goal on software project escalation. *Journal of Management Information Systems*, 29, 1 (2012), 53–78.
26. 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), 59–87.
27. Locke, K.D., and Locke, K. *Grounded theory in management research*. London: SAGE, 2001.
28. Lorenzi, N.M., and Riley, R.T. Organizational issues = change. *International Journal of Medical Informatics*, 69, 2–3 (2003), 197–203.
29. 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, 1 (2009), 39–56.
30. Luthria, H., and Rabhi, F.A. Service-oriented architectures: Myth or reality? *IEEE Software*, 29, 4 (2012), 46–52.
31. Markus, M.L., and Robey, D. Information technology and organizational change: Causal structure in theory and research. *Management Science*, 34, 5 (1988), 583–598.
32. McKendrick, J. So far, SOA failures are few and far between, survey says. *ZDNet*, January 14, 2009.
33. Mohr, L.B. *Explaining Organizational Behavior: The Limits and Possibilities of Theory and Research*. San Francisco: Jossey-Bass, 1982.
34. Moser, F.Z. Faculty adoption of educational technology. *EDUCAUSE Quarterly*, 30, 1 (2007), 66–69.
35. 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), 145–180.
36. Neubarth, M. SOA: Dead or Alive? *CIOZone*, January 27, 2010.
37. Orlikowski, W.J. The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, 3, 3 (1992), 398–427.
38. Orlikowski, W.J. CASE tools as organizational change: Investigating incremental and radical changes in systems development. *Management Information Systems Quarterly*, 17, 3 (1993), 309–340.

39. Papazoglou, M.P., and Georgakopoulos, D. Service-oriented computing. *Communications of the ACM*, 46, 10 (2003), 25–28.
40. Pettigrew, A.M. Longitudinal field research on change: Theory and practice. *Organization Science*, 1, 3 (1990), 267–292.
41. Rahmandad, H., and Weiss, D.M. Dynamics of concurrent software development. *System Dynamics Review*, 25, 3 (2009), 224–249.
42. Repenning, N.P., and Sterman, J.D. Nobody ever gets credit for fixing problems that never happened. *California Management Review*, 43, 4 (2001), 64–88.
43. Repenning, N.P. A simulation-based approach to understanding the dynamics of innovation implementation. *Organization Science*, 13, 2 (2002), 109–127.
44. 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), 265–295.
45. 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), 17–46.
46. Rosenberg, N. *Inside the Black Box: Technology and Economics*. Cambridge: Cambridge University Press, 1982.
47. Ross, J.W.; Curran, C.; and Chapman, J. Reuse and SOA: Recalibrating expectations. Research Briefing of MIT Sloan Center of Information Systems Research, December 5, 2008.
48. Rudolph, J.W.; Morrison, J.B.; and Carroll, J.S. The dynamics of action-oriented problem solving: Linking interpretation and choice. *Academy of Management Review*, 34, 4 (2009), 733–756.
49. Sastry, M.A. Problems and paradoxes in a model of punctuated organizational change. *Administrative Science Quarterly*, 42, 2 (1997), 237–275.
50. 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), 285–313.
51. Sterman, J.D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill, 2000.
52. Tversky, A., and Kahneman, D. Judgment under uncertainty: Heuristics and biases. *Science*, 185, 4157 (1974), 1124–1131.
53. 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), 473–488.
54. Volkoff, O.; Strong, D.M.; and Elmes, M.B. Technological embeddedness and organizational change. *Organization Science*, 18, 5 (2007), 832–848.
55. Vroom, V.H. *Work and Motivation*. New York: Wiley, 1964.

## Appendix A. Details of the System Dynamics Model

The first key variable in the model is *Proportion of Service-Oriented Systems*, which is the ratio of the number of *Service-Oriented Systems* over the number of total *Installed IT Systems*. This variable captures SOA penetration, which reflects the fact that only a proportion of delivered IT systems are service-oriented systems. Because IT developers have to spend extra time to follow SOA design principles [21, 35] for building service-oriented systems, they may not be able to comply with SOA in the implementation of all IT systems, especially when they face schedule pressure and limited resources.

Delivering IT systems that are not service-oriented is tolerable in many organizations that we interviewed. The delivered IT systems are installed with *System Delivery Rate*. *Service-Oriented System Delivery Rate* is a fraction of *System*

*Delivery Rate*. The fraction coefficient depends on the fraction of working hours spent on implementing SOA. In the following we introduce the key feedback loops.

### (1) Balancing Loop B1

We begin with *Proportion of Service-Oriented Systems* in Figure A1 (the starting variable in a feedback loop is indicated with a triangle for readers' convenience). A higher *Proportion of Service-Oriented Systems* enhances the organization's *IS Agility* [11]. *IS Agility Gap*, defined as the difference between *Desired IS Agility* and actual *IS Agility*, results in *Pressure to Implement SOA*. *Management Commitment* is required to generate *Pressure to Implement SOA*, reflecting the fact that *Management Commitment* is a critical success factor of SOA implementation [6, 24]. While IT developers face *Pressure to Implement SOA*, they also have *Pressure to Deliver on Schedule*. IT developers allocate a larger fraction of their work hours spent on functional requirements when *Pressure to Deliver on Schedule* rises, whereas *Pressure to Implement SOA* allows IT developers to reduce *Fraction of Time Spent on Functional REQ*. To operationalize the relationship, we assume *Fraction of Time Spent on Functional REQ* is an S-shaped function of the ratio of *Pressure to Deliver on Schedule* over *Pressure to Implement SOA*. The rationale of using an S-shaped function lies in the assumption that *Fraction of Time Spent on Functional REQ* increases with the ratio (*Pressure to Deliver on Schedule*/*Pressure to Implement SOA*), but its rate of change diminishes as the ratio approaches 0 or infinity. We implement the S-shaped function using a table function (*Table for Pressure*). Table functions in SD modeling are often used to model nonlinear relationships [51].

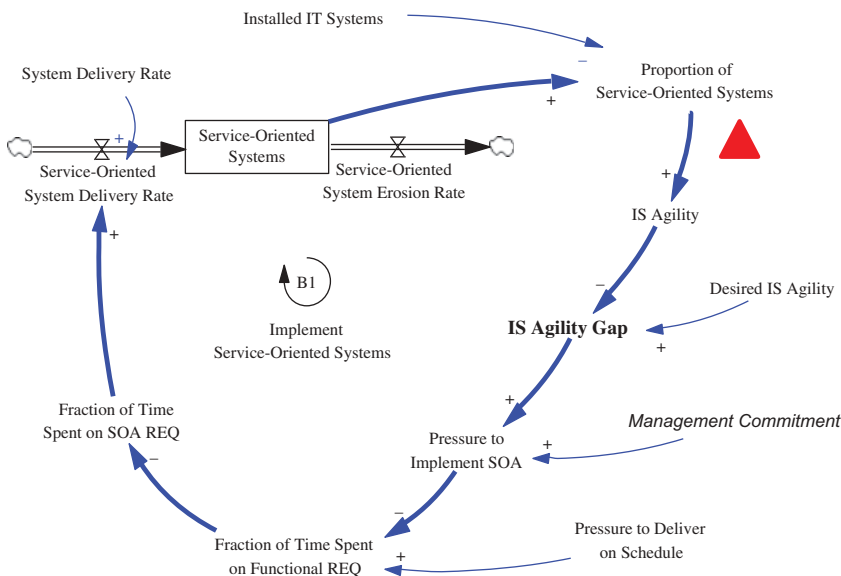


Figure A1. Balancing Loop B1: Implement Service-Oriented Systems under Pressure



## (2) Balancing Loop B2

We begin with *System Delivery Rate* in Figure A2. The IT department has its *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. *Development Productivity* refers to on average how many IT systems a developer can deliver within one unit of time (e.g., one month) when the developer spends all of his work hours on functional development.<sup>5</sup> *System Delivery Rate* refers to how many IT systems the development team delivers within one unit of time. A higher *Fraction of Time Spent on Functional REQ* leads to a higher *System Delivery Rate*. Given *IT System REQ Backlog* and the cycle time requested by business units for delivering IT systems, IT developers calculate *Desired Delivery Rate*. *Delivery Rate Gap* refers to the difference between *Desired Delivery Rate* and *System Delivery Rate*, which creates *Pressure to Deliver on Schedule* and reflects the fact that the primary tasks of developers are to develop IT systems.

## (3) Reinforcing Loop R1

Service-oriented systems developed through the use of SOA design principles are more reusable, interoperable, and easier to integrate with other IT systems [11, 35]. Therefore, it is relatively easier for IT developers to make use of existing reusable service-oriented systems (e.g., components or services) when they develop new IT systems and integrate them with existing service-oriented systems. Accordingly, SOA implementation increases *Development Productivity* of the IT developers on average [11, 17, 35].

In Figure A3 we begin with *Effectiveness of SOA*, which represents the extent to which the developers' *Development Productivity* is increased compared to *Base Development Productivity*. When more service-oriented systems are installed in the organization (*Proportion of Service-Oriented Systems* is higher), SOA becomes more effective and the average *Development Productivity* increases, resulting in the rise of *System Delivery Rate*. *Delivery Rate Gap* is narrowed and *Pressure to Deliver on Schedule* is reduced. As a result, developers are likely to spend more time on developing service-oriented systems, thereby increasing *System-Oriented System Delivery Rate*. Eventually, *Proportion of System-Oriented System* rises further and SOA becomes more effective.

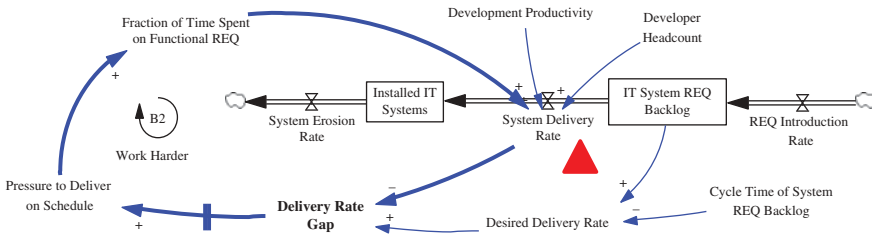


Figure A2. Balancing Loop B2: Work Harder to Deliver on Schedule and Bypass SOA



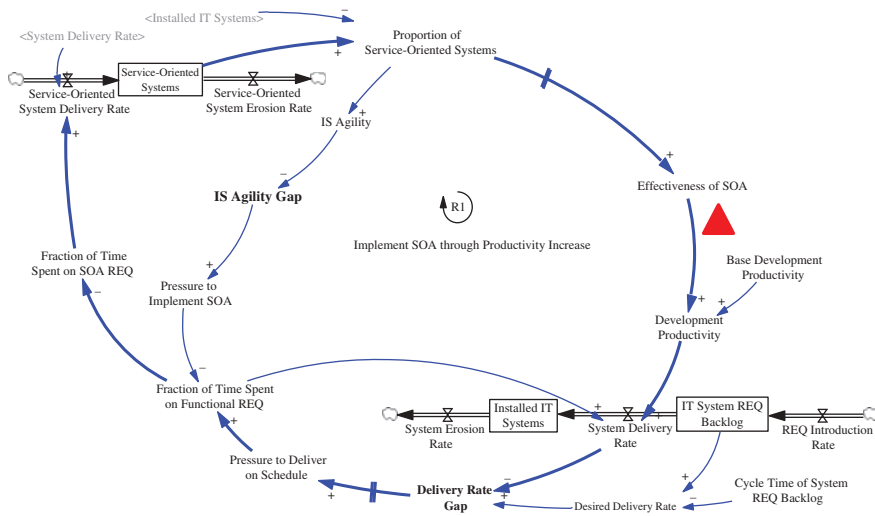


Figure A3. Reinforcing Loop R1: Implement SOA through Productivity Increase

The entire process becomes reinforcing loop R1. A reinforcing loop can operate as either virtuous (i.e., better and better) or vicious (i.e., worse and worse) cycles, depending on its current state [51]. When R1 operates as virtuous cycles, more service-oriented systems are implemented and SOA becomes more effective. Conversely, when R1 operates as vicious cycles, less SOA penetration generates little effectiveness of SOA and contributes little to the developers' development productivity. To successfully implement SOA, it is preferable that R1 operates as a virtuous cycle.

#### (4) Reinforcing Loop R2

We model the key mechanism through which the time delay comes in R1. According to our interviews, *Effectiveness of SOA* depends on two factors: *Proportion of Service-Oriented Systems*, and *Knowledge of Using Service-Oriented Systems* (the organization's knowledge of how to use and build service-oriented systems). Without sufficient knowledge of how to use service-oriented systems, developers may still find it difficult to use existing service-oriented systems or to build new IT systems using SOA design principles, even if *Proportion of Service-Oriented Systems* is high. *Knowledge of Using Service-Oriented Systems* is acquired over time when developers continue to use existing service-oriented systems and adhere to SOA design principles.

We use the first-order information delay [51] to model the process through which *Knowledge of Using Service-Oriented Systems* is acquired. We begin with *Knowledge of Using Service-Oriented Systems* in Figure A4. *Learning Time of Using Knowledge* captures the time delay of the learning process. When service-

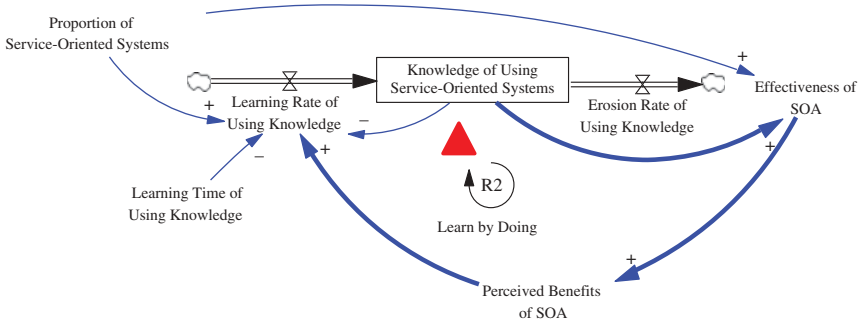


Figure A4. Reinforcing Loop R2: Acquiring Knowledge of Using Service-Oriented Systems by Doing

oriented systems are more technical and complex, the focal organization is expected to accumulate SOA knowledge slowly and *Learning Time of Using Knowledge* is longer. Learning time depends not only on the technical complexity but also on factors such as organization size. We expect *Learning Time of Using Knowledge* to be longer for a larger organization than for a small organization. When more service-oriented systems are installed in the organization (i.e., a higher *Proportion of Service-Oriented Systems*), it is easier to use SOA and faster to acquire SOA knowledge. Also, if service-oriented systems are perceived to be more beneficial, developers are more willing to adhere to SOA design principles, and they acquire SOA knowledge faster. When *Effectiveness of SOA* is realized, developers perceive the benefits of SOA and are more willing to adhere to SOA design principles. Adherence to SOA design principles increases *Learning Rate of Using Knowledge* and *Knowledge of Using Service-Oriented Systems*.

##### (5) Reinforcing Loop R3

We begin with *Effectiveness of SOA* in Figure A5. The positive relationship between the effectiveness of innovative IT (particularly the effectiveness of SOA) and the perceived benefits has been discussed in some detail by Mueller et al. [35]. Choi et al. [11] also point out that SOA implementation effectiveness is an important determinant of the perceived benefits and value derived from SOA. Greater perceived benefits of SOA create favorable word of mouth in the organization [51] and generate additional internal commitment to SOA implementation. The causal link between results of the technology in use and the generation of commitment is also supported by motivation and organizational theories [43, 55]. Commitment generated by the perceived benefits of SOA is considered internal and endogenous, reflecting the additional commitment (*Commitment from Perceived Benefits*) resulting from the use of service-oriented systems.

Besides endogenous commitment, exogenous sources of commitment also contribute to motivating an organization's SOA implementation. Those exogenous sources

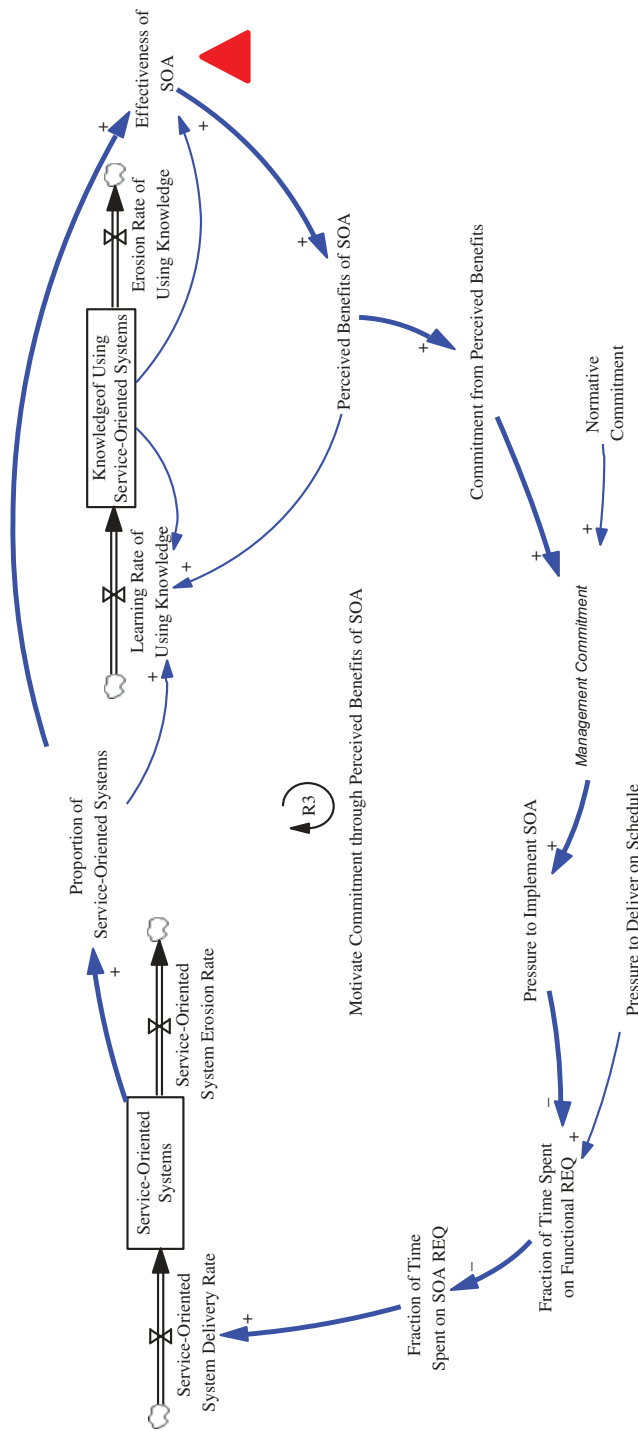


Figure A5. Reinforcing Loop R3: Motivate Commitment through Perceived Benefits of SOA

often come from top management, and we call them *Normative Commitment*. Institutional theory suggests that coercive, mimetic, and normative pressures are important factors affecting innovation adoption [14]. Liang et al. [26] also support the theory that institutional pressures have a positive effect on top management participation in the IT assimilation process. Note that in many cases, top management continues *Normative Commitment* only for a certain period of time because their strategic attention shifts or fades away over time [22]. Based on our interviews, the strategic attention of top management may shift because of changes in the top management team or because of a technology fad. Accordingly, the duration of *Normative Commitment* becomes an important factor in SOA implementation.

Figure A5 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 reduce their hours worked on implementing functional requirements. When service-oriented systems are installed and implemented over time, SOA implementation becomes more effective and enhances *Development Productivity*. With more benefits of SOA perceived by the organization, favorable word of mouth generates internal commitment endogenously and enhances *Management Commitment*.

## Appendix B. Model Assumptions and Validation

The SD model is validated in two ways. On one hand, we validated the simulation results from the model using input and feedback from the subject-matter experts we interviewed. For example, in the scenario of a successful SOA implementation (like the base scenario shown in Figure 2), the model demonstrates the “worse-before-better” phenomenon, which is consistent with the experiences of the experts. Most organizations that succeeded in SOA implementation experienced a period of “performance drop,” and the simulation results of the model replicate their experiences.

On the other hand, we tested behaviors of the model using different variations of the modeling assumptions. The findings from the model are robust relative to the different assumptions and parameter settings. The robustness checks and sensitivity analysis validated the model.

Table B1 lists some of the key assumptions about the SD model and how we justified the assumptions. For example, *Fraction of Time Spent on Functional REQ* is modeled as an S-shaped function of the ratio of *Pressure to Deliver on Schedule* over *Pressure to Implement SOA*. This is done because the qualitative evidence from our interviews shows that *Fraction of Time Spent on Functional REQ* increases with the ratio, but its rate of change diminishes as the ratio approaches 0 or infinity. An S-shaped function can capture this nonlinear relationship. We used different S-shaped functions and the results were qualitatively similar. Thus, the model is robust regarding the selection of different S-shaped functions.

Table B1. Model Assumptions and Validation

Assumptions	Validation and discussion
1. IT developers may not be able to comply with SOA in the implementation of IT systems when they face schedule pressure and limited resources.	This was validated by our interviews with experts and by the literature [11, 21]. In most cases, the first priority of IT developers was to deliver IT functionalities for business users. Many organizations we interviewed confirmed that they allowed IT developers to implement IT systems that were not service-oriented if timely delivery of the IT functionalities was important to business users.
2. <i>Fraction of Time Spent on Functional REQ</i> is an S-shaped function of the ratio of <i>Pressure to Deliver on Schedule over Pressure to Implement SOA</i> .	The qualitative evidence from our interviews showed that <i>Fraction of Time Spent on Functional REQ</i> increased with the ratio, but its rate of change diminished as the ratio approached 0 or infinity. An S-shaped function can capture this nonlinear relationship. We conducted sensitivity analysis and used different S-shaped functions, and the results were qualitatively similar. Thus, the model is robust to the selection of different S-shaped functions.
3. Top management can continue <i>Normative Commitment</i> only for a certain period of time.	This was validated by our interviews and the literature [22]. For example, experts provided multiple reasons why top management's normative commitment could not continue forever: (1) the management team may change, (2) the strategic attention or organizational resources may shift over time due to some crisis, and (3) technology fads shift over time.
4. In the base scenario, top management completely removes normative commitment at a certain time point, and a STEP function is used to model this.	This simplified assumption is realistic in some cases but may not fully capture reality in other cases. Nevertheless, it does not compromise key model insights. As a robustness check, we used an alternative assumption that top management reduces normative commitment gradually (e.g., using RAMP function) and found results that are qualitatively similar. Simulation 3 shows a case in which normative commitment decreases at a constant rate using a RAMP function.

(continues)

Table B1. Continued

Assumptions	Validation and discussion
5. In the base scenario, it takes 12 months for the organization to accumulate sufficient knowledge and experience of using service-oriented systems to be fully effective.	Qualitative evidence from our interviews suggested that many organizations spent 6 months to 2 years (or longer) learning to use SOA effectively. The base scenario is one in which time delay is set to 12 months, and the SOA implementation succeeded. Time delay is different, depending on organizational characteristics (e.g., organizational size) and SOA technology complexity. The simulation analysis examines how the dynamics of SOA implementation change with different values of the time delay.
6. In the base scenario, top management continues its normative commitment for 14 months.	Qualitative evidence from our interviews suggested that top management advocated SOA for 6 months to 2 years. The base scenario is one in which top management advocates SOA and the normative commitment continues for 14 months. The simulation analysis examined how the dynamics of SOA implementation changed with different duration values. <a href="#">Figure 5</a> shows different scenarios of successful SOA implementation when the duration is set at 10, 12, and 14 months. The dynamics resulting from longer durations are qualitatively similar.
7. When using STEP function to model normative commitment, the strength of normative commitment is set to 0.8 before the duration ends.	Setting at 0.8 means the strength of normative commitment counts for 80 percent of the maximum strength of the commitment the organization could give to SOA. Strength is not critical because in most simulations, when we used STEP function to model normative commitment, the magnitude was set as a constant 0.8 before duration ends. Keeping the magnitude constant, our study focused on duration of normative commitment. In the sensitivity analysis we used other values as the constant (e.g., 0.9 or 1.0) and the results were qualitatively similar. In addition, we also used RAMP function to model normative commitment by assuming that strength decreases gradually over time. The simulation results are also similar, suggesting that magnitude is not critical.

Copyright of Journal of Management Information Systems is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.