Preface

Are you worried about getting a business return on your service-oriented architecture (SOA) investment? Are you a business manager who has been disappointed by an information technology (IT) project, or an IT manager or architect who has been disappointed by the business's reaction to your project? If your answer is yes to any of these questions, I wrote this book for you.

If you've been part of one of these disappointing projects, most likely the reason for your disappointment was a disconnect between the business and IT communities—a disconnect that resulted in the project failing to deliver the business value that both sides expected. It is likely that this disconnect was not even recognized until late in the project. So late that a lot of concrete had been poured over the misunderstandings and misconceptions. So late that correcting these misunderstandings and misconceptions began to look like another project.

Such disconnects are simply intolerable in service-oriented architectures. Obtaining a solid return on a SOA investment requires more than simply avoiding such disconnects. It demands proactive and constructive communications between the business and IT communities. The business must clearly define the business objectives of the SOA initiative—the things that will provide the actual business return on the SOA investment.

The business and IT communities must then join forces and work together to achieve these objectives. Together, they must define the business process and system changes required to produce the expected business results. This collaboration is not just to make SOA initiatives succeed. It is essential for any project that is supposed to produce business value. For the most part, failed projects are projects that have either lost sight of the business objectives or failed to focus the business process and system changes on achieving those objectives.

The challenge here is that business processes and information systems have become so intertwined that it is literally impossible to make changes to one without altering the other. In particular, changes to information systems alter business processes—often in unexpected and undesirable ways. Despite this, project plans rarely stop to actually consider the design of business processes, unless the project happens to be tackling a major business process reengineering effort. As a result, business processes just sort of evolve, piecemeal, project by project, as you make changes to your information systems.

When the scope of a project lies entirely within a single business unit, you can get away with this. Working within that business unit, the business users and developers sit down and discuss how it's going to work, and the developers go off and update the systems. This casual approach works reasonably well when there is only one development group and only one user group. However, this approach is woefully inadequate when there are multiple user groups, multiple development groups, and multiple systems involved. In fact, it is a recipe for disaster.

Service-oriented architectures are supposed to bring an end to this chaos. SOA is supposed to provide clean, well-defined interfaces between business entities—between service providers and service consumers. But who, exactly, are those service providers and service consumers? Are they systems? Well, yes and no. There are, indeed, systems providing and consuming services. However, those systems are providing functionality on behalf of business units for use by other business units. Thus, when you define business services, you are actually defining the boundaries between business units and the interfaces between them. In other words, you are defining the structure and organization—the very architecture—of your business units.

The architecture of business units is not a technical issue—it is a business issue. *SOA determines the architecture of both business units and systems!* Consequently, both business and IT need to work together to successfully implement SOA.

A Closer Look

Taking another look at those failed projects, a couple of questions arise. Which projects failed? Most likely the ones that involved multiple business units. Why did they fail? They failed because nobody on the business side of the house thought out what the business process needed to be and how the various business units and systems ought to participate in that process. Or, if they did, they failed to succinctly communicate this understanding to the IT developers.

Either way, this left the IT developers—often multiple groups of developers—guessing as they defined the dialogs between information systems belonging to different business units. Guessing about what the overall business process was supposed to be and guessing about how it should handle exceptions. They implemented these guesses, and only then were the inadequacies of the guesswork recognized. Then they began the arduous process of evolving these business processes into something that actually worked on a business level—as the project slipped into cost overruns and delays.

Let me ask you a question. Would you consider automating the interactions between your enterprise and one of its suppliers without first coming to an agreement about what those interactions would be? How the quote-order-shipment-payment process would actually work? Of course not. Then why on earth should you treat the internal dialog between your business units—your order management group, your warehouse management group, and your financial group—any differently?

Let's be clear about this. I'm not talking about massive business process reengineering. I'm simply talking about taking the time to think out the business process, thinking through what the business process ought to look like and how the business units and information systems will participate in that process. You need this picture to include both sunny-day scenarios and exception handling, and then convince yourself that this vision will produce the business value you expect from the project. Then, and only then, should you make an investment in implementing the business process and system changes.

Why is this important? Because it is business processes that actually provide value to the enterprise. Yes, information systems are an increasingly important component of those business processes, but there is no inherent business value in the systems themselves. Their value lies entirely in their ability to make business processes work. Business processes are what is important. The reason we do IT projects is to make business processes provide more value.

Oops! I slipped (on purpose). In that last paragraph, I made the very mistake I am trying to help you avoid! I said that business processes are important—and then immediately started talking about IT projects.

This thinking, that there is a separation or schism between business and IT, has become an institutionalized habit in many enterprises. It usually extends all the way up to the very top of the organizational hierarchy. Yet this same business–IT separation or schism is the root cause of many project failures.

This schism is an outright showstopper for SOA. This is why architects, business managers, and IT managers, from the frontlines to the chief operating officer (COO) and chief information officer (CIO), need to work together to solve this problem. You are the only ones who can provide the solution.

So what am I asking you to do? If you are a manager, you can help by doing these four things.

1. ***Understand the nature of the problem***. Business process architecture determines how business units interact with one another to make business processes work. Systems architecture determines how systems interact with one another to make the technical part of the business process work. Business processes and systems have become so intertwined that you can't design one without designing the other. Thus, business process architecture and systems architecture are but two different views of the same architecture—the total architecture.
2. ***Put someone in charge of your total architecture***. If you are to make sense of and manage your enterprise's total architecture, somebody needs to own it—all of it. Business process architecture and systems architecture belong together, under one roof. Today systems architecture is in the IT organization, and as for business process architecture—well, nobody owns that! Don't believe me? Try to find someone in your organization who can describe your complete order-to-cash business process! Then ask yourself how you are supposed to manage something you can't even describe. You need to put someone in charge of your total architecture and get it under control.
3. ***Demand total architecture visibility.*** Demand that the architecture of your business processes and systems be captured in a form that can be readily understood and shared. Only then will it be possible to have meaningful discussions about modifying and improving business processes before you incur the cost of actually implementing the changes.

4. ***Provide the authority needed to manage the total architecture***. To be effective, the group managing the total architecture needs to report to the business operations manager responsible for executing business processes. This is the only person who (a) sets the enterprise priorities and (b) has the authority to command cooperation and change in both the business and IT sides of the house. You don't need to do a massive reorganization. But you do need to take leadership personnel from business and IT, task them with rationalizing and managing how your business actually works, and support them with the authority to make it work.

If you are a business process or systems architect, take action in these ways.

1. ***Understand the depths of the interdependencies between business processes and systems***. From determining the functional boundaries between business units and systems to determining the performance requirements and appropriate level of investment in fault tolerance and high availability, all system requirements are derived from an understanding of the business process. These are your responsibilities.
2. ***Understand the ultimate dependency of systems upon people for flexibility***. Understand the importance of feedback in detecting and responding to breakdowns in business processes and systems.
3. ***Understand the social and organizational issues surrounding your work***. Understand the extent to which your work depends on the cooperation of all organizations involved. Be on the lookout for signs of misaligned priorities, and take action to raise the visibility of these misalignments.
4. ***Understand how to efficiently organize a total architecture development***. Understand the importance of providing early feedback regarding cost and schedule feasibility.
5. ***Execute every project from the total architecture perspective to provide true business value***. Design business processes and systems together to deliver expected business benefits.

Succeeding with SOA is a call to action for both managers and architects.

For managers, the call is to set the organizational stage to actively manage your total architecture. Architecting your business is a business activity, not an IT activity. The business side of the house needs to take ownership and lead this effort. Put someone in charge of designing and documenting your overall business processes! They are, after all, the lifeblood of your enterprise. Don't leave them to chance.

For architects, the call is to realize that you are architecting business units and business processes as well as systems, and to structure your work accordingly. This volume will help you understand the context for your work. The companion volume, *SOA in Practice: Implementing Total Architecture*, will give you the tools and techniques for actually developing a total architecture.

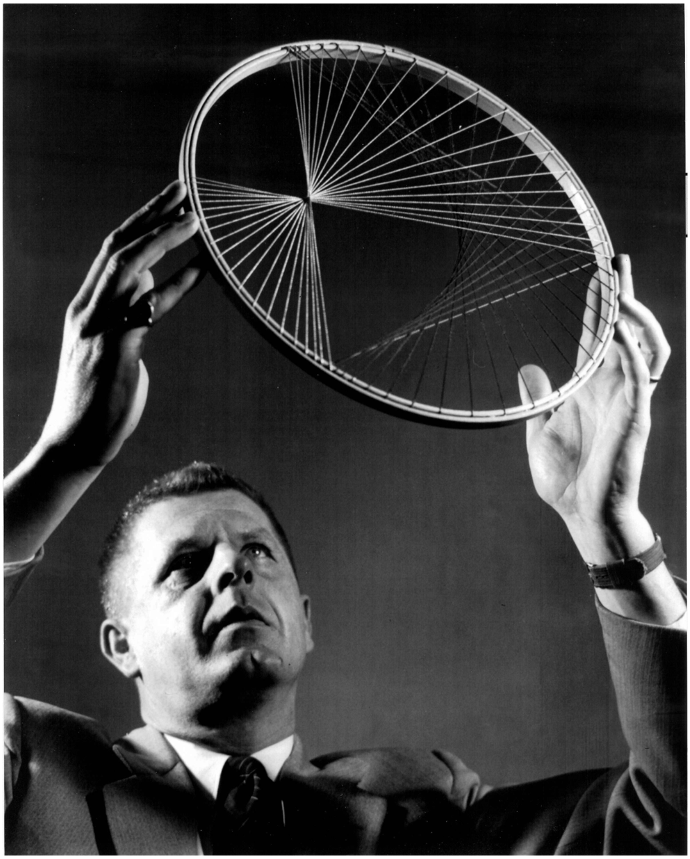
About the Cover

An ironic coincidence arose during the production of this book in the form of an unintended linkage between the cover art and the book's dedication. The cover art uses strings to show complex linkages between people. To me, this image is an appropriate visual analogy for the complex relationships between individuals and organizations that we find in our everyday business processes. Understanding these relationships and using that understanding to guide the design of both business processes and systems is the focal point of this book. The visual analogy could not be better.

Now, the irony. William Cady Stone, to whom the book is dedicated, was a Professor of Mathematics at Union College. One of his hobbies was using string models to visually illustrate complex mathematical relationships. In addition to the models he constructed himself, he painstakingly restored Union College's collection of the original Olivier Models dating from the nineteenth century. You can visit Union's web site at www.union.edu/Academics/Special/Olivier/ to learn more about Professor Stone and the Olivier models.

Acknowledgments

I would like to thank the many individuals who helped to make this work and this book possible. I give thanks to my mentors who helped me learn how to explore uncharted territory: John Reschovsky, Joel Sturman, David Oliver, David Musser, and Mukkai Krishnamoorthy. I thank my colleagues who have provided an intellectual foil for these ideas: Jonathan Levant, John Hutchison, James Rumbaugh, Michael Blaha, and William Premerlani. For their support of my enterprise



William Cady Stone. (Photograph by John Ross, Schenectady, NY. Reprinted with permission of the Stone family.)

methodology work, I thank Brian Pierce, Bruce Johnson, and Paul Beduhn. For their help in sharpening the real-world architectural concepts, I thank Paul Asmar, David Leigh, Saul Caganoff, and Janet Strong. For helping me turn a concept into a book, I thank Michael Blaha and William Premerlani. For helping me turn the book into a reality, I thank Paul Asmar, Ram Menon, Roger Strukhoff, Scott Fingerhut, Peter Gordon, and Donna Davis. Last, but far from least, I would like to thank my wife, Maria, for her patience, love, and support while bringing this book to life.

PCB
Schenectady, NY
January 2007