

Controlling Complexity in Enterprise Architectures

Executive Overview Part I of III



**A Briefing Paper by Roger Sessions
ObjectWatch, Inc.**

Introduction

Just about everything in large organizations is becoming more complex. On the business side, regulatory requirements, partnership relationships, mergers, and acquisitions all serve to make business processes more complex. On the technology side, new distribution models, heightened demands for interoperability, and workflow automation all serve to make IT systems more complex.

In the midst of all of this mushrooming complexity comes a requirement for agility. Rapid changes in business models, customer expectations, and market conditions all put inexorable demands on organizations to keep IT systems agile.

This, in a nutshell, is the challenge: to build highly complex IT systems, ensure that those systems meet the needs of increasingly complex business processes, and do all of this in a way that allows everything to adapt quickly to changing market conditions.

How do we accomplish all of this? Clearly there must be a single vision that guides both business process evolution and IT investment. That vision is what we commonly call *enterprise architecture*.

But most existing enterprise architecture methodologies evolved in a much simpler era. Thus, while they address the traditional problem of IT/business alignment, they do not address the much more difficult problem of managing today's complexity.

This is the first of a three-part series that introduces a new approach to enterprise architectures. The process is called *simple iterative partitions* (SIP™). SIP is your best tool to manage complexity.

The Problem of Complexity

The failure to manage IT complexity is the single biggest reason that IT systems so often fail. And when complexity is the culprit, the failures are catastrophic, expensive, and usually highly visible. Examples of complexity-induced failures are pervasive across the public and private sector and range in cost from tens of thousands to billions of dollars. But if you are in a major organization, you don't need anybody to tell you this. You almost certainly have your own list of expensive failures.

The problem is that existing enterprise methodologies do not address complexity. In fact, they can *never* address complexity because they do not have a logical model for understanding what complexity means and how complexity behaves in IT systems.

Why do we need a model for complexity? As an analogy, imagine a moon launch. A moon launch is a highly complex undertaking. Suppose we did not have a model for gravity and planetary motion. Without such a model, every decision about the moon launch, from the amount of fuel, to the power of the thrust, to the angle of the launch would need to be based on gut feelings, experienced guesses, and raw intuition. How many such moon launches would be likely to succeed?

But since we do have models for gravity and planetary motion, we can ensure that every possible detail of the moon launch is correct within the predictive power of the relevant models. Such consistency may not guarantee the eventual success of the moon launch, but it considerably improves the odds.

The same holds true for enterprise architecture, many of which equal moon launches in complexity! Because existing enterprise architecture methodologies do not have a model for complexity, they cannot test proposed solutions against that model. Using such methodologies to manage the complexity of today's IT systems is much like trying to orchestrate a moon launch without understanding the mathematics of gravity.

SIP is the only enterprise architecture methodology that is based on a mathematical model for IT complexity. It is a model grounded in both probability theory and set theory. This model predicts IT complexity just as surely as physics predicts gravitational forces. In the same way that NASA scientists can use the model of gravity to predict fuel consumption, you can use the SIP model to predict the relative complexity of proposed IT solutions.

This briefing paper is an overview of SIP. For more information, read Part II, *Controlling Complexity in Enterprise Architectures: Mathematical Foundations*, which provides an explanation of the mathematical models that are used to control complexity in enterprise architectures. Part III, *Controlling Complexity in Enterprise Architectures: The SIP Methodology*, pulls all the pieces together and provides an overview of the entire SIP process. All of these briefing papers are by Roger Sessions and available from ObjectWatch (information@objectwatch.com).

The SIP Solution

The SIP approach is completely focused on controlling the complexity of an enterprise architecture. First, partitioning is used to dramatically reduce overall complexity, often by several orders of magnitude. Second, simplification algorithms are applied using defined mappings between business requirements and IT functionality. IT functionality that does not map to business requirements is eliminated. IT functionality that is redundant is consolidated. IT functionality that can be obtained is outsourced. Finally, the enterprise architecture is validated through a controlled process of prioritization and iteration.

The result is that high value-producing IT systems are delivered more quickly, overall costs are reduced, and the risk of failure is diminished. **High value, quick delivery, and reduced risk. These are the benefits/deliverables/fruits of the SIP meta-methodology.**

SIP is not intended to replace existing enterprise architecture methodologies. It is intended to provide a meta-methodology on top of them. This meta-methodology addresses the one issue that existing methodologies do not address: complexity.

The SIP Process

The SIP process usually occurs in five phases. These phases are as follows:

1. SIP Preparation – The organization is trained in the importance of complexity control, governance is put in place, and the methodology blend that will be used to guide the enterprise architecture is identified.
2. Partition Identification – The enterprise is partitioned into a collection of autonomous subsets using decomposition algorithms and equivalence relations. These subsets are tested according to the requirements of set theory.

3. Partition Simplification – The partition is simplified by removal of extraneous, redundant, or outsource-able functionality.
4. Partition Prioritization – The subsets of the partition are prioritized based on business value and technical risk.
5. Partition Iteration – The functionality of the subsets are delivered iteratively, using the blended methodology created in the SIP preparation phase.

The Result

There are a number of benefits that can be realized by applying the SIP meta-methodology. All of these stem from the complexity control that is the SIP focus. These benefits typically include the following:

- Reduced cost. Simpler systems are less expensive to develop and maintain than more complex systems.
- Faster delivery. Simpler systems can be delivered more quickly than more complex systems.
- Improved IT/business synergy. Simpler systems are easier to analyze and map to business requirements than more complex systems.
- Greater agility. Simpler systems are more flexible and easier to modify than more complex systems.

The greater the complexity of the enterprise, the greater the benefits that SIP provides. Faster delivery, higher value, reduced cost, and diminished risk. That is the SIP promise.

More on SIP

SIP was invented by Roger Sessions. Roger Sessions is the CTO of ObjectWatch. He has written six books (including *Software Fortresses*; *Modeling Enterprise Architectures*) and many articles. He is on the Board of Directors of the International Association of Software Architects (IASA), Editor-in-Chief of *Perspectives of the International Association of Software Architects*, and a Microsoft™ recognized MVP in Enterprise Architecture. He holds multiple patents in both software and enterprise architecture process. An accomplished key note speaker, Mr. Sessions has given talks at more than 100 conferences around the world covering a wide range of topics in Enterprise Architecture.

For more information on how SIP can help you manage the complexity of your IT projects, write information@objectwatch.com.

Legal Notices

The SIP methodology is protected by pending patents. SIP and Simple Iterative Partitions are trademarks of ObjectWatch, Inc. ObjectWatch is a registered trademark of ObjectWatch, Inc.