

Controlling Complexity in Enterprise Architectures

The SIP Methodology Part III of III



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

Introduction

This paper is about the methodologies needed to develop the overall architectural vision for an organization and specifically for managing its most pressing problem: complexity. Managing complexity is the primary responsibility of the enterprise architect. This is the architect who specializes in the broadest possible view of architecture within the enterprise.

As both IT systems and business processes become ever more complex, bringing these two sides together is an increasingly difficult challenge. Existing enterprise architecture methodologies do not address complexity in any meaningful way. They have no models for complexity against which architectures can be validated.

This is why SIP (Simple Iterative Partitions) is so important. It gives us models for complexity and a process for validating an architecture against those models in the early design phases, when changes can be made much more easily. It gives us a mathematically grounded approach for controlling complexity - the single biggest reason that large IT projects fail.

This briefing paper discusses the SIP process. It is the third of a three-part series. The first briefing paper gives an executive overview of the problem of complexity and the SIP approach to solving this problem¹. The second briefing paper discusses the mathematical underpinnings of SIP—namely, set theory, partitions, probability theory and equivalence relations². We recommend you read the first two parts of this series prior to reading this final section. Instructions for accessing the papers are below, in the footnote.

The purpose of this briefing paper is to give you a better idea of what it might look like to bring SIP into your organization. You will find many unique aspects of SIP that have no counterparts in other methodologies. These aspects are focused squarely on the problem of complexity.

SIP Process Overview

The SIP process has these goals for an enterprise architecture:

- Complexity Control – No surprise here. Complexity is the focal point. Our premise is that no significant business value is possible until complexity is controlled.
- Logic-Based Decisions – Most enterprise architecture decisions are based on instinct, gut feeling, politics, vendor loyalty, etc. SIP seeks to remove these emotion-based decisions and instead approach every decision from a rationally and mathematically grounded perspective.

¹ *Controlling Complexity in Enterprise Architectures - Executive Overview* by Roger Sessions, available by writing information@objectwatch.com.

² *Controlling Complexity in Enterprise Architectures - Mathematical Foundations* by Roger Sessions, available by writing information@objectwatch.com.

- Value Driven Deliverables – SIP is value driven. All deliverables are measured against quantifiable business value.
- Reproducible Results – SIP provides a methodology that is reproducible. Two SIP practitioners analyzing a similar enterprise should come up with architectural solutions that are similar.
- Verifiable Architectures – SIP produces architectures that can be verified logically and mathematically.
- Flexible Methodology – SIP can be used with most other enterprise architecture methodologies because SIP completes them by addressing concerns that the other methodologies do not address.

The SIP process, at a high level, is shown in Figure 1.

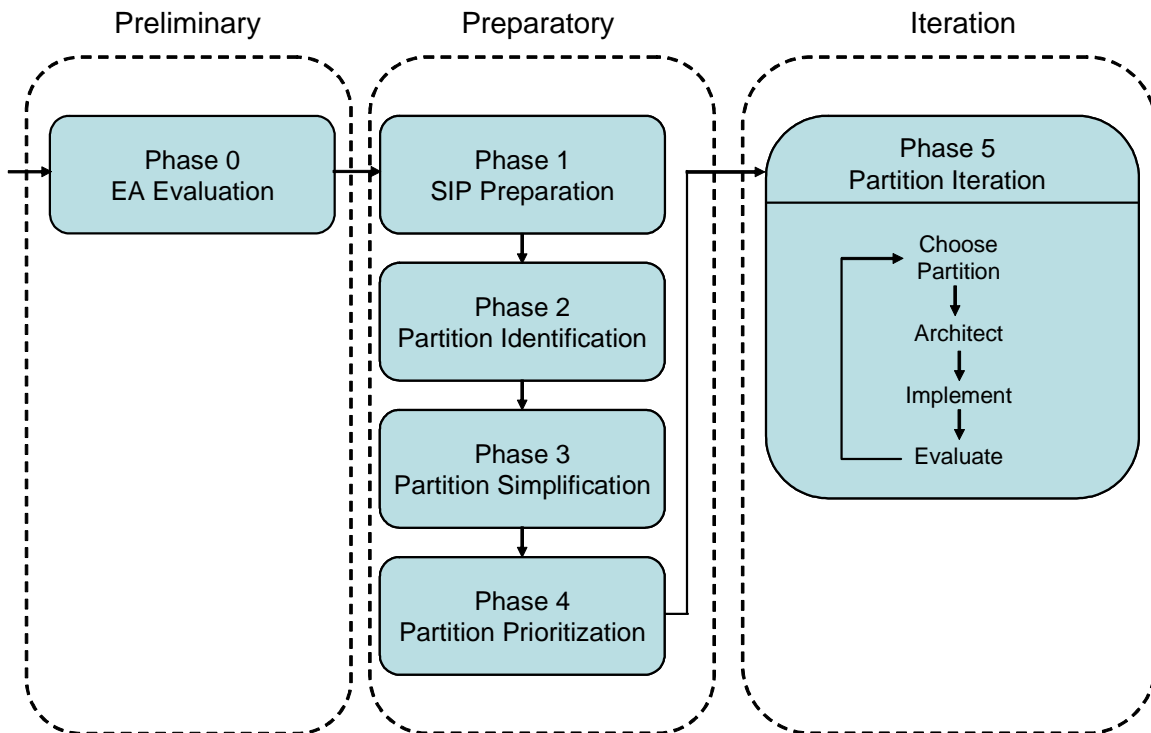


Figure 1. SIP Process (Overview)

As you can see in Figure 1, the SIP process consists of six primary phases that can roughly be divided into three categories. The categories are as follows:

- Preliminary – Activities that typically occur as an organization is evaluating its readiness to leverage an enterprise architecture, in general, and whether SIP, specifically, is a good match for its needs.
- Preparatory – Activities that culminate in the identification and prioritization of the enterprise partitions, guided by the mathematical models for complexity and partitioning.

- Iteration – The ongoing activities of designing, implementing, and delivering value-rich partitions.

Before we look at the individual phases, we need to examine more closely how partitions manifest themselves in the SIP methodology. This brings us to the concept of an *autonomous business capability* (ABC), our basic unit of enterprise partitioning.

ABC – The Enterprise Partition

The starting point for SIP is an autonomous business capability (ABC). *Autonomous* means that the functionality works independently of other ABCs. (Recall from the mathematical models of SIP that autonomy is a concept closely related to partitions and equivalence relations.) *Business* means that the ABC has some well-defined business purpose. *Capability* means that it is capable of producing some effect that is visible from the outside world. An ABC is thus a unit of the business that functions autonomously from other business units, but interacts with those units in well-defined ways.

An example of an ABC is *Shipping* or *Accounts Payable*. The ABC model greatly simplifies the overall business model for two important reasons. First, the model benefits from the simplicity that naturally results from partitioning and the SIP process. Second, the model slashes through implementation issues by looking only at *what* an ABCs does, not *how* it does it.

Figure 2 pictorially represents the ABC concept. Notice that an ABC typically includes both business process and technology elements.

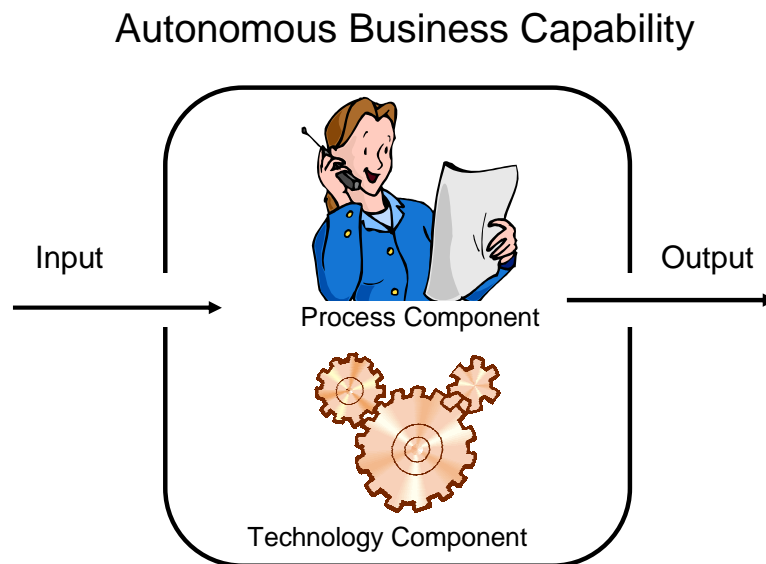


Figure 1. A Prototypical ABC

Once we understand the concept of the ABC, we model the business as a collection of ABCs. Figure 3, for example, shows part of a retail business model.

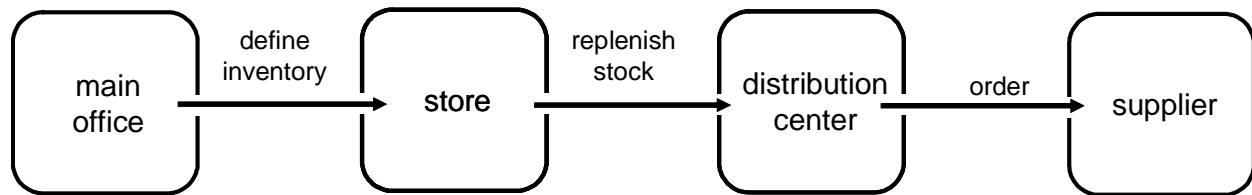


Figure 3. Hypothetical ABC Model for Retail Operation

The ABC is the most fundamental partition in the SIP methodology. Now that we understand ABCs, let's look more closely at the process that is responsible for their discovery.

Phase 0 – Path Evaluation

The starting point for a SIP (or any enterprise architecture) engagement is typically an internal investigation as to the potential value of creating an enterprise architecture. While every organization is going to have its unique style of investigation, we have several tools that can help in this effort. These tools include:

- *A Comparison of the Top Four EA Methodologies* by Roger Sessions. A 44-page white paper that compares the Zachman, TOGAF, FEA and Gartner/Meta methodologies in depth.
- A self-administered Value Graph Analysis that evaluates an organization's readiness for enterprise architecture, in general, and the applicability of the SIP methodology, specifically.
- A series of briefing papers on SIP, one of which you are now reading.

Phase 1 – SIP Preparation

At this point, an organization has concluded that it is in a position to leverage an EA effort and that the SIP focus on managing complexity resonates. It is in Phase I that we lay the groundwork for a successful SIP engagement. This typically includes the following deliverables:

- An audit of organizational readiness
- Description of an organization-specific SIP EA governance structure
- Description of how SIP will blend in with other organizational processes
- Internal training in the challenges of complexity and the organization-specific SIP approach
- Creation of organization-specific standardized forms and analysis tools

Phase 2 – Partition Identification

In Phase 2 partitions that will be implemented are identified. Partition identification involves taking an ABC (autonomous business capability), fractionating it into lower-level ABCs, and then using equivalence functions to repartition those fragments into lower level, logically autonomous ABCs.

We rarely use this technique on the enterprise as a whole but instead on those segments of the enterprise that are relevant to the project at hand. This process can often be speeded up by using our collection of over 900 representative enterprise ABC's adapted from the APQC's Process Classification Framework. Figure 4 is an example of a partial decomposition of an enterprise using our collection:

- Develop marketing, distribution, and channel strategy
- Develop and manage sales strategy
- Manage advertising, pricing, and promotional activities
- Manage sales partners and alliances
- Manage sales opportunity and sales pipeline
 - Identify and manage key customers and accounts
 - Develop key customer plans
 - Identify priority customers
 - Establish budgets
 - Develop sales/key account plan
 - Schedule calls to customers
 - Execute agreed-to sales plan
 - Prepare/Analyze/Evaluate sales results
 - Initiate sales order process
 - Sales order management
- Deliver Products and Services
- ...

Figure 4. Representative Partition (ABC) Breakdown of an Organization

Once we have identified the lowest level partitions of interest, we need to categorize the resulting collection of ABCs. These categories will be very helpful in the next phase as we seek opportunities for partition consolidation. Some of these categories will be enterprise-specific, but many can be taken directly from our extensive category collection.

Phase 3 – Partition Simplification

Major strides in reducing complexity are made as soon as the system is partitioned (as described in the *Mathematical Foundations* briefing paper). But there is more work to be done. In this phase, efforts are made to simplify the partitions even further.

There are three main simplification approaches we use in this phase. They are as follows:

- **partition elimination:** Check each ABC partition against the overall system requirements. If the ABC is not absolutely required, eliminate it. For example, in a marketing system, we might look hard at an ABC whose function is sending unsolicited email.
- **partition consolidation:** Check each ABC partition against each other ABC partition. If two ABC partitions are similar, consolidate them. An observation that two (or more) ABCs share the same category (as assigned in Phase 2) hints strongly that consolidation may be possible.

- partition outsourcing: For each ABC, ask whether that ABC is central to the mission of the organization or serves more of a support functionality. In the latter case, outsourcing should be investigated.

Phase 4 – Partition Prioritization

We now have a collection of ABCs that are ready to move onto the assembly line. A core facet of the SIP philosophy is that this roll-out should occur iteratively rather than en masse. This phase assigns the *order* of iteration.

The SIP philosophy advocates that rollout order should be based on two factors: *business value* and *risk*. Our goal is to rollout the high value/low risk ABCs first. This is in contrast to the philosophy of many methodologies that advocate attacking high risk aspects of the project first, on the theory that this helps risk management.

In our experience, the highest risk of all is organizational credibility. Most organizations suffer a credibility gap between the business and the technical sides. Any new enterprise architectural effort is likely to be met with initial skepticism. Early failures greatly feed this skepticism while early successes help turn skeptics into supporters.

We therefore want to ensure that the early SIP results are viewed as highly successful. The best way to achieve initial success is start with low hanging fruit: ABCs that are ripe with business value and have the least possible chance of failure.

It is important to use a consistent criteria in evaluating ABCs for rollout priority. Our approach is to use a Value Graph Analysis tool that automatically translates business value and risk into a visual rendition that facilitates rational comparison between ABCs. Examples of Value Graphs for two ABCs are shown in Figure 5.

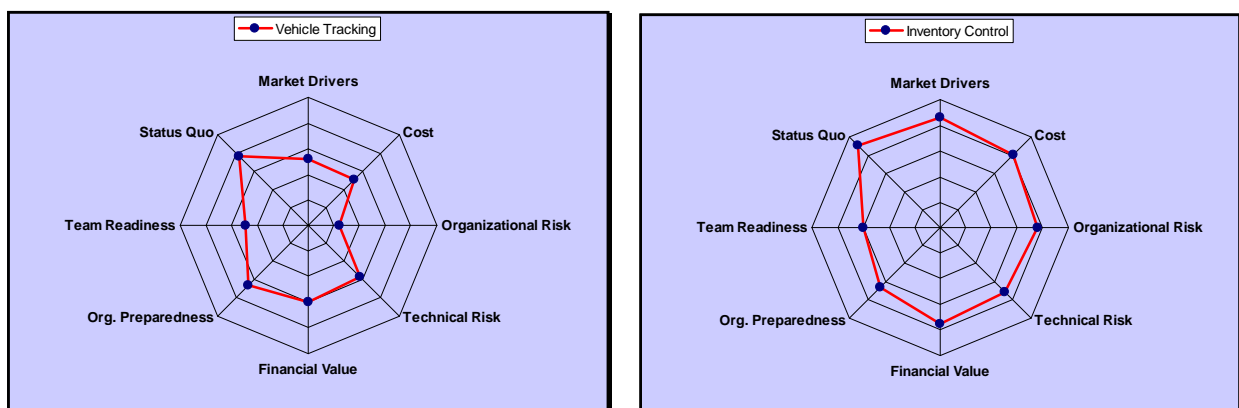


Figure 5. Value Graphs of Two ABCs

In Figure 5, the ABC on the left (*Vehicle Tracking*) is a lower priority than the ABC on the right (*Inventory Control*). This assessment can be made quickly from the visual coverage of their

respective targets. The first ABC has low target coverage, the one on the right much greater target coverage. The higher the coverage, the higher the priority.

It is important that the evaluation criteria be identical for all ABC assessments. A standard organizational evaluation criteria is thus one of the outputs from Phase 1, and that criteria will guide all of the priority evaluations. Our typical ABC priority assessment is based on more than 50 individual criteria organized in 8 broad groups, but the exact number of criteria and groupings will be organization-specific.

Phase 5 – Partition Iteration

The final phase is the most important. Here, iteration begins. The process iterates through the partitions that were first identified in Phase 2, then simplified in Phase 3, and finally prioritized in Phase 4. In Phase 5 the goods are delivered.

Surprisingly, this phase is the one in which SIP is most flexible and on which SIP has the least opinions. The reason for this is that most organizations already have in place good procedures for documenting business requirements, architecting IT systems, and delivering software.

There are multiple methodologies that work well in this phase, including The Open Group's TOGAF, ZIFA's Zachman framework, Microsoft's Solution Framework, and IBM's Unified Process. We don't believe that SIP should reinvent the wheel or dictate in areas that are not relevant to the problem SIP set out to solve (and hopefully, by now you know what that is—controlling complexity!).

By the time we hit Phase 5, SIP's work is largely done. It has transformed a world that that once seemed hopelessly complex into a world made up of manageable, autonomous, value-rich, bite-sized morsels. Implement these morsels anyway you choose.

Of course, we are not totally opinion-less on the topic of good implementation strategies. But our opinions are more in the nature of a check list. Do you have a process for documenting business requirements? For managing architectural artifacts? For establishing program requirements? For measuring milestones? For showing ROI? What do we think is the best way to do these things? Most likely, the best way to do these things is the way you have already been doing them.

Summary

This briefing paper is about the SIP process. SIP stands for Simple Iterative Partitions. The SIP process is straight-forward: partition the universe; simplify those partitions; and then iterate through them. Behind the SIP process is a world of mathematical models, tools, and methodologies.

SIP is about *reducing* complexity. It is not a methodology for creating *complex* architectures for complex systems. There are any number of enterprise architecture methodologies that can create complex architectures. SIP is about creating *simple* architectures for complex systems.

All enterprise architectural methodologies (including SIP) must ultimately be judged on how well they deliver business value. Most methodologies do this by first architecting the business systems and then architecting technology systems to support those business processes. Without an approach to controlling complexity, however, the resulting architectures can end up quite convoluted, as shown in Figure 6.

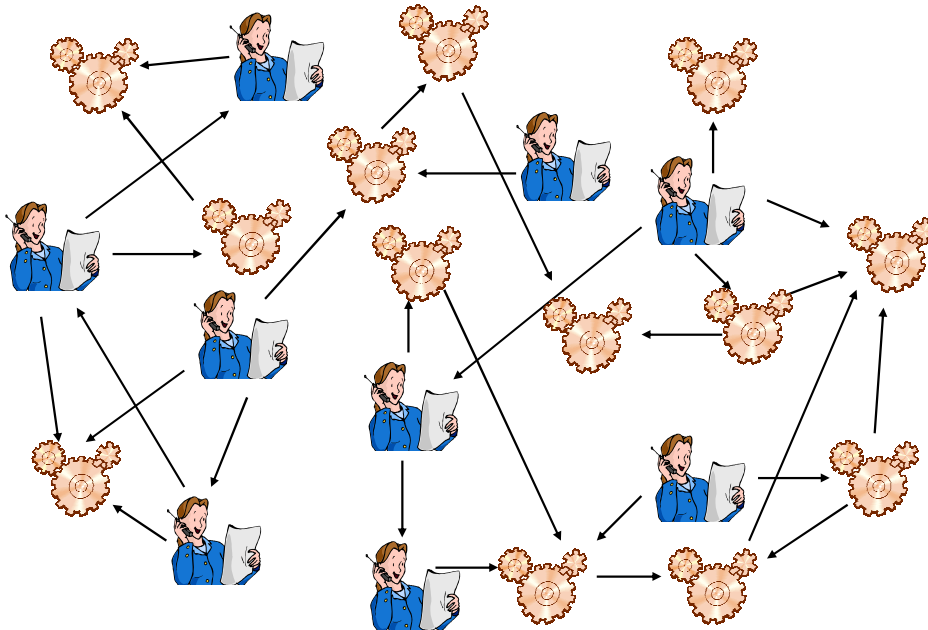


Figure 6. Typical Enterprise Architecture

The SIP enterprise architecture is much more manageable, as shown in Figure 7.

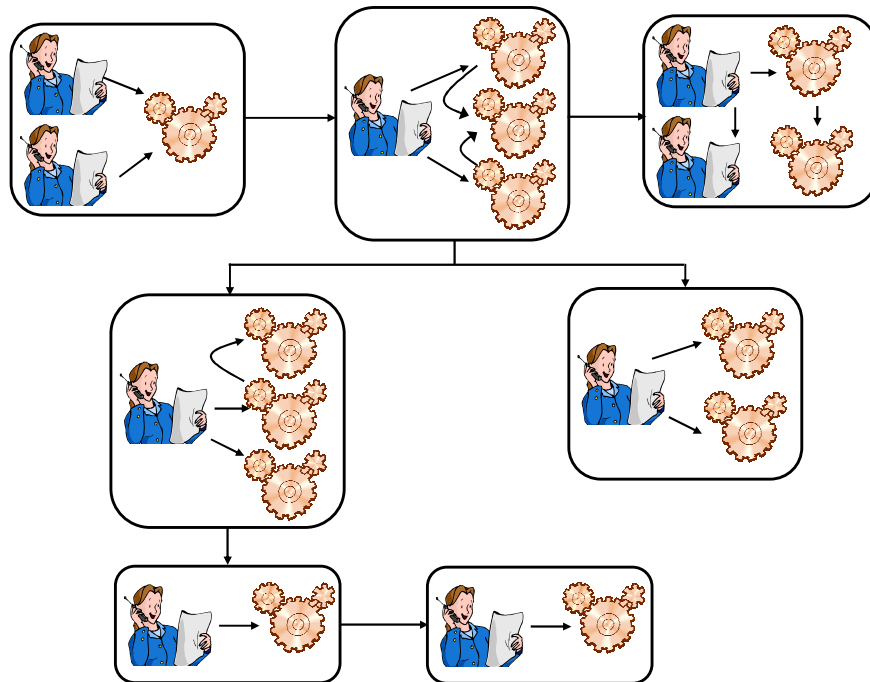


Figure 7. Typical SIP Enterprise Architecture

Simplicity is not free. It takes dedication, discipline, and experience to architect for simplicity. But the resulting architectures are *much* easier to implement, *much* easier to maintain, and *much* more likely to deliver value. Which system do you think is more likely to be delivering business value five years down the line — the typical system shown in Figure 6 or the SIP system shown in Figure 7?

We think the answer is simple. The answer is SIP.

More on SIP

SIP was invented by Roger Session. Roger Sessions is the CTO of ObjectWatch. He has written six books (including *Software Fortresses; Modeling Enterprise Architectures*) and numerous 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 to 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. This briefing paper is copyright© 2007 by ObjectWatch, Inc. All rights reserved.