CHAPTER 1

Enterprise Architecture and Enterprise Engineering

In the preface we discussed the evolution of enterprises: from the Agricultural Age to the Industrial Age and then to the Information Age. We saw that we evolved from manual chaos using manual processes to automated chaos using automated processes. The manual processes were essentially automated as is, without effective redesign of those processes to take real advantage of the new technologies that were employed. The result today is that we have twenty-first-century enterprises with systems that use twenty-first-century technologies, yet most enterprises today still use processes that were originally designed in the eighteenth century!

The enterprise architecture methods of enterprise engineering as described in this book enable business experts and IT experts together to identify reusable business activities, reusable business processes, and integrated databases for business integration. These take advantage of the latest technologies for technology integration. The result is the evolution to integrated twenty-first-century enterprises that have been transformed through the introduction of reusable twenty-first-century processes!

Chapter 1 addresses the role of enterprise architecture for enterprise integration. To understand this, we will first discuss the evolution of enterprise architecture.

1.1 The Evolution of Enterprise Architecture

Enterprise architecture was developed by John Zachman while with IBM in the 1980s, after observing the building and airplane construction industries and the IT industry. He saw similarities between the construction of buildings, airplanes, and the information systems used by an enterprise.

These industries manage the design, construction, and maintenance of complex products by considering the needs of different people. Figure 1.1 illustrates the owner in the building industry, who uses architect’s drawings to decide that the building addresses specific requirements. For airplane manufacture, the owner uses the high-level work breakdown structure of the plane to determine requirements. For information systems, the owner uses a model of the business to determine the enterprise needs.

The designer, however, needs a different set of diagrams: architect’s plans for the building, sets of engineering design diagrams for the plane, or information system models for the enterprise.
The builder relies on still different types of diagrams: contractor’s plans for construction of the building, a manufacturing engineering design for plane construction, or technology models for information systems.

In addition, there are a number of different questions—called primitives or interrogatives or abstractions—that also need to be considered. These are illustrated in Figure 1.2.

What is needed is important to know. This is represented in Figure 1.2 by material, such as bills of materials for buildings and planes, and data models for information systems. How these are used is indicated by functions, such as functional specifications for buildings and planes, and functional models for information systems. Where is also important, as indicated by location—in drawings for building and plane construction and in network models for information systems.
Bringing these concepts together, the result is a matrix of five rows and three columns. These represent the perspectives of the planner, the owner, the designer, the builder, and the subcontractor, who are all interested in what, how, and where.

The last row addresses the functioning enterprise. The sixth row is not normally counted in the five main rows of the Zachman framework. Further, different documentation, models, or representations may also be utilized in each cell of the Zachman framework as shown by Figure 1.3. For example, reading down column 1—What (Data)—of this figure we see that:

- The cell formed by intersection of the objectives/scope row (of interest to the planner) and the data column shows that a “list of things” is relevant to this cell.
- The cell intersected by the owner row and data column is the “enterprise model”—also called the strategic model. We will discuss the role of strategic models in more detail in Part II of this book. For example, in Chapter 7 we will see that the strategic model enables enterprise-wide data integration to be realized.
- The cell for the designer row and the data column shows that “logical data model” documentation applies to this cell. This expands the strategic model to integrated logical data models with data attribute detail.
- The builder row and data column cell contains the “physical data model” for subsequent data implementation in target databases.

<table>
<thead>
<tr>
<th></th>
<th>What Data</th>
<th>How Function</th>
<th>Where Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANNER</td>
<td>List of Things</td>
<td>List of Processes</td>
<td>List of Locations</td>
</tr>
<tr>
<td>Objectives/Scope</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OWNER</td>
<td>Enterprise Model</td>
<td>Activity Model</td>
<td>Business Logistics</td>
</tr>
<tr>
<td>Conceptual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBCONTRACTOR</td>
<td>Data Definition</td>
<td>Program</td>
<td>Network Architect.</td>
</tr>
<tr>
<td>Out-of-Context</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUNCTIONING</td>
<td>Data</td>
<td>Function</td>
<td>Network</td>
</tr>
<tr>
<td>ENTERPRISE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.3 Different model representations exist in each of 15 cells to address the perspective of each row and the focus of each column.
• The subcontractor row and data column cell contain “data definition” scripts for the physical installation of these databases.

Reading down column 2—How (Function)—and column 3—Where (Location), we also see that each row has various representations in the cells for these columns as well. Several types of models may also be relevant to each cell. These models should all be well defined, but this complete definition is difficult to achieve in most enterprises.

For all things that we consider in business or day-to-day life—whether for buildings, for planes, or for complex enterprise systems—there are in fact six independent variables. These are based on the six primitive interrogatives: what, how, where, who, when, and why.

There are a further three columns—Who, When, and Why—in the complete Zachman framework for enterprise architecture. These additional interrogatives are shown in Figure 1.4, which illustrates a complete Zachman framework. In most enterprises, we will see that the models represented in these additional 15 cells are rarely well defined. We will also see in Part II that column 6—Why (Motivation)—is a very important column: It typically defines the business needs of an enterprise for the future.

Figure 1.4 shows examples of typical model contents for each cell. For example, the How column (column 2) shows that an Activity Model is relevant to the Owner row (row 2). This is a key cell, because it enables the return on investment (ROI) of alternative activities to be assessed through activity-based costing (see Chapter 8).

<table>
<thead>
<tr>
<th></th>
<th>What Data</th>
<th>How Function</th>
<th>Where Location</th>
<th>Who People</th>
<th>When Time</th>
<th>Why Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANNER</td>
<td>List of Things</td>
<td>List of Processes</td>
<td>List of Locations</td>
<td>Organization Structure</td>
<td>List of Events</td>
<td>List of Goals Objectives</td>
</tr>
<tr>
<td>Objectives/Scope</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OWNER</td>
<td>Enterprise Model</td>
<td>Activity Model</td>
<td>Business Logistics</td>
<td>Work Flow</td>
<td>Master Schedule</td>
<td>Business Plan</td>
</tr>
<tr>
<td>Conceptual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESIGNER</td>
<td>Logical Data Model</td>
<td>Process Model</td>
<td>Distributed Architecture</td>
<td>Human Interface</td>
<td>Process Structure</td>
<td>Business Rules</td>
</tr>
<tr>
<td>Logical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUILDER</td>
<td>Physical Data Model</td>
<td>System Model</td>
<td>Technology Architecture</td>
<td>Presentation Interface</td>
<td>Control Structure</td>
<td>Rule Design</td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBCONTRACTOR</td>
<td>Data Definition</td>
<td>Program</td>
<td>Network Architecture</td>
<td>Security Interface</td>
<td>Timing Definition</td>
<td>Rule Specifications</td>
</tr>
<tr>
<td>Out-of-Context</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUNCTIONING</td>
<td>Data</td>
<td>Function</td>
<td>Network</td>
<td>Organization</td>
<td>Schedule</td>
<td>Strategy</td>
</tr>
<tr>
<td>ENTERPRISE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.4 The complete Zachman framework for enterprise architecture is based on a further three columns, for a total of six columns and five rows—making up 30 cells. Each cell may contain various types of models, as illustrated in this figure.
As a further illustration, column 2, row 3—a cell of interest to the Designer—shows that a Process Model is relevant for this cell. We will discuss process models in Chapter 10.

In summary, the framework rows therefore indicate different views (or perspectives) of people in the enterprise, from the perspectives of the planner, owner, designer, builder, and subcontractor. (The last row, “the functioning enterprise,” is not normally counted.) The framework columns also address different primitive questions (also called interrogatives or abstractions) of What, How, Where, Who, When and Why.


1.2 Using the Zachman Framework for Enterprise Architecture

The focus of enterprise architecture initially should be based on rows 1 and 2, from the perspectives of the planner and the owner. These perspectives typically focus on the motivation as indicated by column 6 (Why), which represents business plans for the enterprise. Clear strategic directions can then be provided to row 3 (for the Designer), row 4 (for the Builder) and row 5 (for implementation by the Subcontractor).

The complete Zachman framework for enterprise architecture is illustrated now in Figure 1.5, showing representative models for each of the 30 cells. This

Figure 1.5  The Zachman framework for enterprise architecture illustrates how we have traditionally taken a bottom-up focus to the Designer’s row 3; we have then built down again from row 3. We have not typically taken the perspectives of the Planner or the Owner rows into account. (Courtesy of John A. Zachman.)
framework is available on the CD-ROM as a large foldout page [3]. Print this page now; leave it for easy reference as you read further.

Traditionally, in building enterprise systems we have taken a bottom-up view. We have looked at the existing systems—whether manual or automated—represented by the bottom row of the framework. From this view, we have looked at ways in which current manual or automated systems have been implemented. We examined ways to improve these systems: either by automating manual systems or by using new technology to improve existing automated systems. We have taken a design focus from the perspective of row 3 (designer) and then moved back down again to rows 4 and 5 (builder and subcontractor), using different technologies to bring about the desired improvements. This approach, however, is quite technical. Traditionally, it has been difficult to include the perspectives of the owner (at row 2) or the planner (at row 1). Parts I and II cover methods that involve senior managers and business experts in enterprise architecture.

1.2.1 The Difference Between Primitives and Composites

John Zachman makes the case that by addressing the six primitives (the interrogatives or questions of what, how, where, who, when, and why) very complex composites (such as buildings, planes, or enterprise systems) can be developed. Answers to these questions, he states, can be used to capture knowledge that is needed to construct any complex (composite) object.

He notes that by taking a top-down approach, building construction and airplane design have developed interchangeable parts that can be reused. He gives the example of standard doors and windows in buildings. He points out that “the Boeing 737, 747, 757 and 767 airplanes were designed so they all use a standard undercarriage. But it is hard to achieve reusability if each component is built from scratch each time” [2].

He develops this reusability principle further by saying that [2]:

The IT industry has tried to build reusable code or components by using object-oriented methods. But we have not been particularly successful to date. We do use O-O to build reusable components for screen design and other systems components. But we have not been very successful using O-O methods to identify many reusable activities and processes within an enterprise. Enterprise reusability is only achieved effectively by taking an enterprise-wide approach: not in detail across the enterprise, but broadly to encompass the whole enterprise.

This enterprise-wide view is illustrated in Figure 1.6 as horizontal “slices” in each cell. For example, a high-level view of the business plans for an enterprise is shown by the horizontal slice at the top of each cell for column 6 (Why) with row 1 (Planner), and column 6 (Why) with row 2 (Owner). We will see in Part I that strategic planning uses strategy analysis in Chapter 3 to identify these horizontal slices in column 6 as a high-level list of goals/objectives and high-level business plans. These comments introduce an initial three enterprise architecture principles—out of a total of six key principles:
Column 6 (Why) for both the Planner and Owner rows are two important primitive cells, used as a starting point focus based on the business plans defined for the future.

Each horizontal slice extends across the full width of its cell, to show that it is “enterprise-wide.” Because this horizontal slice is “high level,” it typically extends down to only approximately 10% of the depth of a cell.

The full depth of a cell represents “an excruciating level of detail,” to quote John Zachman [2].

Similarly, these high-level business plans are used to identify people in the organization structure (column 4—Who, row 1—Planner) who have business expertise in the areas addressed by those plans, together with knowledge of the high-level data that are required to implement the plans within the enterprise. This highlights the fourth principle:

Column 4 (Who) for the planner row is another key primitive cell. It identifies business experts in the organization structure who know the data and the processes that are suggested by the business plans.

Column 1 (What—Data) with row 1 (Planner) in Figure 1.6 shows a List of Things as a high-level horizontal slice in that cell. Column 1 (What—Data) with row 2 (Owner) further represents this data as an Enterprise Model. A high-level horizontal slice of an Enterprise Model—called a Strategic Model—applies to this cell. Chapter 7 describes how strategic modeling is used to develop a strategic model as horizontal slices for these two cells. The fifth principle is therefore:

---

<table>
<thead>
<tr>
<th>Horizontal Slice</th>
<th>What</th>
<th>How</th>
<th>Where</th>
<th>Who</th>
<th>When</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANNER</td>
<td>List of Things</td>
<td>List of Processes</td>
<td>List of Locations</td>
<td>Org Structure</td>
<td>List of Events</td>
<td>List of Goals/Objectives</td>
</tr>
<tr>
<td>OBJECTIVES/SCOPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OWNER</td>
<td>Enterprise Model</td>
<td>Activity Model</td>
<td>Business Logistics</td>
<td>Work Flow</td>
<td>Master Schedule</td>
<td>Business Plan</td>
</tr>
<tr>
<td>CONCEPTUAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHYSICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBCONTRACTOR</td>
<td>Data Definition</td>
<td>Program</td>
<td>Network Architect.</td>
<td>Security Interface</td>
<td>Timing Definition</td>
<td>Rule Specs</td>
</tr>
<tr>
<td>OUT-OF-CONTEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUNCTIONING</td>
<td>Data</td>
<td>Function</td>
<td>Network</td>
<td>Organization</td>
<td>Schedule</td>
<td>Strategy</td>
</tr>
<tr>
<td>ENTERPRISE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 1.6* An enterprise-wide approach is shown as a horizontal band across the full width of each cell. A high-level view of the models within each cell is shown as a narrow “slice” for the enterprise-wide band at the top 10% of each cell.
• Column 1 (What) for the Planner and Owner rows are two important primitive cells. They define the strategic model as the integrated data resource (of the enterprise model) that is required by the business plans.

Furthermore, horizontal slices in column 2 (How–Function) for row 1 (Planner) and for row 2 (Owner) represent a high-level List of Processes and high-level Activity Models. Part II will show how a data model can be used to identify a list of processes in column 2, row 1 as a List of Activities. This list is then used to identify and define activity models in column 2, row 2. Chapters 7 and 8 discuss how reusable activities can be identified from data models, for further documentation as activity models for ROI analysis through activity-based costing. The sixth principle of enterprise architecture is therefore:

• Column 2 (How–Function) for the Planner and Owner rows are two important primitive cells. They identify reusable business activities from the strategic model (in an enterprise model) and from the business plans.

Other methods are discussed in Part II for these primitives: column 3 (Where–Location); column 4 (Who–People), and column 5 (When–Time) for each of rows 1 and 2 (Planner and Owner).

### 1.2.2 Identifying Reusable, Priority Areas for Early Delivery

The high-level focus of the horizontal “slice” at the top of each cell, shown by Figure 1.6, enables priority areas to be identified that need to be implemented first. These are shown as vertical “slivers” in Figure 1.7, extending for the full depth of the cell at an “excruciating level of detail” [2] for the subset of the cell represented by the sliver.

<table>
<thead>
<tr>
<th>Horizontal Slice</th>
<th>What</th>
<th>How Function</th>
<th>Where Location</th>
<th>Who People</th>
<th>When Time</th>
<th>Why Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANNER Objectives/Scope</td>
<td>List of Things</td>
<td>List of Processes</td>
<td>List of Locations</td>
<td>Org Structure</td>
<td>List of Events</td>
<td>List of Goals/Obj</td>
</tr>
<tr>
<td>OWNER Conceptual Enterprise Model</td>
<td>Activity Model</td>
<td>Business Logistics</td>
<td>Work Flow</td>
<td>Master Schedule</td>
<td>Business Plan</td>
<td></td>
</tr>
<tr>
<td>FUNCTIONING ENTERPRISE</td>
<td>Data</td>
<td>Function</td>
<td>Network</td>
<td>Organization</td>
<td>Schedule</td>
<td>Strategy</td>
</tr>
</tbody>
</table>

**Figure 1.7** Priority areas for early delivery are identified as vertical “slivers” within each cell, which extend to the depth of the cell. Resources are allocated to these priority areas (slivers) so they can be defined, built, and delivered first.
From the planners’ and owners’ perspectives in rows 1 and 2 of Figure 1.7 we can see that vertical slivers in each cell enable greater detail to be defined in priority areas. These areas progress to detailed definition (represented by the full depth of the vertical sliver in each cell) for rapid implementation using appropriate tools and technologies. Thus, these areas can be delivered early, before other, less important areas that can wait until later. Part II discusses methods for rapid identification and definition of reusable activities and processes. Part III discusses technologies that are used to deliver these activities and processes rapidly into production as systems.

Rows 1 and 2—from the perspectives of the Planner and Owner—are critical for business transformation. These two rows are used to identify reusability opportunities within an enterprise.

Figure 1.8 highlights rows 1 and 2 (Planner and Owner) of the Zachman framework. A number of cells in these rows are vitally important. We have discussed that Business Plans in column 6 (Why) are the most important because such plans are used to set directions for the future. This is developed further in Chapters 3 and 7. We also discussed that column 4 (Who) identifies the Organization Structure in row 1. This enables business managers and business experts to identify relative priorities based on the business plans.

The business plans from column 6 are used to develop a high-level strategic model in column 1 (What). This is an important cell: It is vital in identifying the integrated high-level data that are needed to manage the progress of the enterprise toward the future. Activity models in column 2 (How) are also vital: These activity models are used to identify critical activities that should be carried out by the enterprise in the future.

John Zachman comments that “enterprise architecture is used for the management of enterprise change.” In fact, “if enterprise architecture is not used,” he says [2], “there are only three options for managing enterprise change: by trial and error; by reverse engineering; or by going out of business!”

<table>
<thead>
<tr>
<th>Planner</th>
<th>Owner</th>
<th>Designer</th>
<th>Builder</th>
<th>Subcontractor</th>
<th>Functioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives/Scope</td>
<td>Conceptual</td>
<td>Logical</td>
<td>Physical</td>
<td>Out-of-Context</td>
<td>Enterprise</td>
</tr>
<tr>
<td>What Data</td>
<td>List of Things</td>
<td>Logical Data Model</td>
<td>Physical Data Model</td>
<td>Data Definition</td>
<td>Data</td>
</tr>
<tr>
<td>How Function</td>
<td>List of Processes</td>
<td>Process Model</td>
<td>System Model</td>
<td>Program</td>
<td>Function</td>
</tr>
<tr>
<td>Who People</td>
<td>Org Structure</td>
<td>Human Interface</td>
<td>Prean Interface</td>
<td>Security Interface</td>
<td>Organization</td>
</tr>
<tr>
<td>When Time</td>
<td>List of Events</td>
<td>Process Structure</td>
<td>Control Structure</td>
<td>Timing Definition</td>
<td>Schedule</td>
</tr>
</tbody>
</table>

**Figure 1.8** The Planner and Owner rows (rows 1 and 2) are used for Reusability Definition. The Owner row is most effective in its ability to identify enterprise-wide reusability opportunities.
1.3 Enterprise Engineering for Rapid Development

From this brief introduction to enterprise architecture, we will look at typical systems development problems. The typical approach that is used to design and build enterprise systems with traditional systems development methods is summarized here:

- Systems requirements have typically been defined by IT staff, by interviewing users to determine their operational business needs.
- The designs that are established are then based on technology, with application design, database design, and object design reflecting that technology.
- These designs are then implemented to meet desired business performance requirements.

This traditional approach to systems development has been *technology dependent* and has resulted in problems:

- The *business needs* have been difficult to determine. If these needs are not understood or expressed clearly, the designed systems may not address the real needs of the users and management.
- The systems that are developed are typically not aligned with corporate goals that set directions for the future. This is one of the main problems with systems development today.
- But the strategic directions are not clear; yet they must be understood if IT is to design flexible systems that support the strategic directions.

In fact, problems with traditional development methods are much greater than suggested by the preceding list. The business needs have traditionally been decided by reviewing the operational processes of the business. These processes were determined based on strategic plans typically defined many years ago, sometimes more than a decade ago.

Yet in the early 1990s we had no idea—not even in our wildest predictions of the future—that we would today be able to communicate instantly with customers, suppliers, and business partners anywhere in the world, through the Internet. The environment that we accept today as the norm was way beyond our most fanciful imagination.

The strategic plans defined in the 1990s did not anticipate that these organizations would today communicate with each other in seconds. They assumed communication would be as it had always been, by mail—or later by fax—with responses received days or weeks later. The most rapid response these business processes assumed was at best in hours. The business processes we still use today were never designed to respond in seconds.

This point is vitally important and should be emphasized:

The traditional systems development approach—interviewing users based on existing business processes and then identifying their future needs—does not work well in periods of rapid change, such as today.
In fact, I will make this point stronger:

If we base our needs for the future on operational processes that we still use today, we are implicitly assuming that the future will be similar to the past. This is very dangerous; few industries and enterprises can say today that their future will be like their past. Most know that the future will be quite different. The only certainty we have is that the processes we will need then are quite different from the processes we use today.

This brings me to emphasize a very important principle for change:

We must design for tomorrow based not on operational processes still used today. We have to design for tomorrow by using new activities and processes tailored for the environment of the Internet—which represents our present and our future—so that enterprises can respond in seconds or minutes, not in days or weeks.

Enterprise engineering provides support for business transformation: a future where the only thing that is constant … is change itself. Businesses must change, to compete with other organizations in their relevant markets. This is true for commercial organizations that compete with other organizations. It is true for government departments that compete with other departments for government funding. And it is also true for defense departments, which compete with hostile defense forces, and also with friendly defense forces for limited resources.

Competition today demands systems that can change easily, to support rapid business transformation. Many business changes may need significant change or redevelopment of systems. Yet most of those systems were not designed for change. Existing systems may need massive modification to support essential business changes. Often it is faster to throw the existing systems away and start over again, developing new systems from scratch. This can still be slow and very costly.

The advantages and benefits of technology were not clear in the early 1990s to many senior managers. It was sometimes difficult to get funding approval for new projects and funding for the resources that are vital for success. But the Internet and the Y2K problem in the late 1990s demonstrated to management the dramatic impact—both positive and negative—that technology can have on the enterprise.

We discussed earlier that we have taken a bottom-up view with traditional methods in building systems for the enterprise. We looked at the existing systems—whether manual or automated—as represented by the bottom row of the Zachman framework for enterprise architecture.

From a bottom-up view, we looked at ways in which current manual or automated systems have been implemented. We then examined ways to improve these systems: either by automating the manual systems, or by using technology to improve existing automated systems.

As discussed with Figure 1.5, we have taken a design focus from the perspective of row 3 (Designer) using traditional methods, and then moved down again to rows 4 and 5 (for Builder and Subcontractor), using technologies to bring about the desired improvements. We saw that this approach is quite technical. Traditionally, it has been difficult to include the perspectives of the Owner (at row 2) or the Planner (at row 1).
How can we address these problems and involve the Planner and the Owner in setting transformation directions for the future? We will now consider solutions to these problems arising from the traditional approach to systems development:

- The systems that are to be developed for the future must support the corporate goals. This is the most common systems development problem today.
- We must therefore determine the goals for the future. But goals are expressed in business terms, not systems terms. What should we implement?
- We earlier discussed that IT departments must be aware of strategic directions so they can design for the future. In the 1990s this was difficult because most IT departments did not participate in strategic planning. However, this is changing; many CIOs now come from the business side rather than from IT.
- Yet we have seen that IT must build systems based on strategic plans if those systems are to be aligned with corporate goals. They must be based on activities and processes designed for the future, not the past.
- If this is done, technology can then offer competitive advantage: It can be used to help achieve the strategic plans and corporate goals, with new activities and processes that respond in seconds or minutes—not in days or weeks.

Today, enterprise engineering resolves these problems with systems development. It enables business experts and IT experts to work together in a design partnership using modeling tools (previously called CASE tools for computer-aided software engineering). Enterprise engineering utilizes modeling tools and methods for business transformation by business experts and IT experts to do the following:

- Build systems for the future that can support the corporate goals.
- Identify goals for the future in business terms, so that IT can determine what to implement in systems terms.
- Provide strategic business planning methods so that the IT department can participate in strategic planning with management.
- Enable IT to build systems based on the strategic plans so that those systems are aligned with corporate goals.
- Technology can then offer competitive advantage—used to help achieve the strategic plans and corporate goals.

We will now examine the business-driven enterprise engineering methodologies in more detail. These methods support all phases of the systems development life cycle (SDLC). Figure 1.9 illustrates that phases above the line are technology-independent methods and focus on the business. They apply to rows 1–3 (Planner, Owner, and Designer) of the Zachman framework. These methods are strategic business planning, data modeling, and function modeling:

- The strategic directions set by management provide input to strategy analysis, discussed in Chapter 3 for column 6 of the Zachman framework.
These plans indicate the information requirements of management that are input to data modeling in column 1. Strategic, tactical, and operational data modeling are covered in Chapters 6, 7, and 9.

Plans and data models define information usage as input to function modeling, for activity modeling, scenario modeling, and process modeling in column 2. These are covered in Chapters 8 and 10.

These phases of Figure 1.9 define technology-independent business requirements and address enterprise architecture rows 1–3 (for the Planner, Owner, and Designer). Phases below the bold line in the figure are technology dependent. They address enterprise architecture rows 3–5 (for Designer, Builder, and Subcontractor). These methods address component design and systems implementation:

- Technology and systems requirements of the business provide input to systems design. Internet technologies and object-oriented methods in this phase are used for application design, database design, and object design of systems to be deployed on corporate intranets and/or the Internet.
- Identified performance requirements then provide the input required by the systems implementation phase.

The first enterprise engineering phase is strategic business planning. This identifies the planning and requirements needs of the enterprise for the future. Strategic plans are represented by column 6 (Why) in the Zachman framework. Strategic business planning uses the method of strategy analysis to determine the strategic plans for the future, as discussed in Chapter 3. Strategy analysis is used to accomplish the following:
Identify goals from existing strategic plans, so that a clear understanding can be reached of the business needs of the enterprise.

Help develop business goals (where they do not yet exist)—or refine any goals that already exist—to ensure that business results are clearly stated.

Help develop project goals from business goals—or refine any project goals that already exist—to ensure that the business results and the project results can be clearly expressed and define what the project must achieve.

Consider alternative technologies for implementation as discussed in Part III.

Provide knowledge of strategic business planning methods and terminology to help IT experts and business experts provide technology input to the business plans.

Guide an organization’s technology agenda. Strategy analysis identifies priorities for early delivery. In conjunction with the other enterprise engineering methods, it supports a powerful rapid-delivery capability for large organizations.

After strategy analysis, strategic modeling methods use business plans to develop a strategic model. This is used to develop an enterprise architecture portfolio plan (EAPP) for project planning. We will learn in Chapter 7 how to develop project plans from data models. We will derive required project plans for enterprise architecture implementation. We then use technology for early delivery of priority systems.

Looking at the data modeling phase in more detail in Figure 1.9, Strategic business planning identifies the information requirements of management and provides input to this phase:

- Strategic business plans provide input to strategic modeling, to develop a strategic data model, called a strategic model. This is discussed in Chapter 7.
- Analysis of the strategic model produces an EAPP as mentioned earlier.
- The strategic model, the EAPP, and tactical business plans all provide input to tactical data modeling, to develop tactical data models.
- The EAPP, tactical models, and operational business plans also provide input to operational data modeling, to develop operational data models.
- Data modeling and business normalization methods—discussed in Chapters 6 and 9—are used to develop strategic data models, tactical data models and operational data models.

Data modeling is used to develop a strategic model from strategic plans for the rapid development of high-level business data models. These data models are used to develop project plans to deliver high-priority and high ROI systems early.

Data modeling also helps to identify various alternatives, leading to business benefits. This provides business justification for technology alternatives, funding approval for the technology, and resources for implementation.

The EAPP report is a deliverable from strategic modeling and strategic model analysis. We will see in Chapter 7 that this establishes clear project plans for priority
projects. It leads to detailed development of approved projects. Tactical and operational data models then define databases in detail, ready for implementation.

Function modeling addresses column 2 (How) of the Zachman framework. It is based on the information usage of management, as determined by the strategic plans defined by strategy analysis (in column 6—Why). Information requirements of management (from data models in column 1—What) also provide input to function modeling, which includes the following:

- **Activity modeling**: This indicates what has to be done to provide the required information to management. Activity models address column 2 (How) in rows 1 and 2 (for Planner and Owner) and are discussed in Chapter 8.

- **Process modeling**: This indicates how processes are to be carried out, based on required activities. This addresses column 2 (How) in rows 2 and 3 (for Owner and Designer) and is described in Chapter 10.

- **Scenario modeling**: This indicates who is involved in activities and processes. It identifies people from the organization structure (in column 4—Who, row 1—Planner) based on strategic alignment matrices in Chapter 8.

Function modeling is used to model business activities as activity models (also using activity-based costing) and as process models that define business processes. It aligns activities and processes to strategic plans to support corporate goals, project goals, and system goals. It is used for development of approved projects, to define business objects for object-oriented development. Function modeling ensures that systems can change rapidly.

We earlier discussed the component design phase. We saw the typical approach that has been used to design and build enterprise systems previously with traditional systems development methods. However, by using the prior technology-independent phases of enterprise engineering, the business needs for the future are now clearly defined from strategic plans and business plans as a result of these methods:

- **Strategy analysis** to define strategic business plans for the future.
- **Data modeling** to develop strategic, tactical, and operational data models.
- **Function modeling**, using activity modeling, activity-based costing, process modeling, and scenario modeling.

The business priorities are now clearly defined from business needs and project plans in Parts I and II. Data models are now fully developed at strategic, tactical, and operational levels to address future needs. Activity models and process models are now fully developed, with business processes defined as business objects for future needs and environments. Technology is then used for rapid development as discussed in Part III. This use of enterprise engineering is summarized in Box 1.1.

### 1.4 Using Enterprise Architecture for Enterprise Integration

At the start of this chapter we discussed that enterprise integration depends on business integration and also technology integration. Business integration is achieved...
through the use of enterprise architecture and related enterprise engineering methods. Technology integration is achieved with the use of XML, EAI, enterprise portals, Web services, and SOA, as discussed in Part III. We will now discuss some of the implications of business integration.

1.4.1 The Importance of Metadata

Enterprise integration is critically dependent on a clear definition of the metadata used in an enterprise. When asked to define the meaning of metadata, most IT experts respond with a definition of “data about data” or “information about information.” These definitions are meaningless to nontechnical business managers. They do not even begin to explain the meaning of metadata, let alone its vital importance for enterprise integration, business integration, and technology integration. Yet a clear definition of the metadata of an enterprise—referred to as enterprise metadata—is vital for success in each of these integration endeavors. A better definition of metadata is provided by the nontechnical analogy provided in Box 1.2.

As we discussed in relation to Adam Smith in the preface, enterprises have historically evolved with different terminology in various parts of the organization. The need for a common language for communication in an enterprise was not recognized. Consider the problems that arose as computers were introduced to automate processes and data. We discussed that this introduced problems of data redundancy, data maintenance redundancy, and process redundancy. To achieve business integration and technology integration for business transformation, common terminology must be used.

Data modeling is used to identify metadata and define what each term means. These definitions are captured by data modeling tools and stored in a repository. Agreed-on common terms, with other enterprise terminology, constitute the enter-

<table>
<thead>
<tr>
<th>Box 1.1: Enterprise Engineering for Enterprise Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>The use of enterprise engineering methods for enterprise architecture results in rapid definition of a strategic model for an enterprise—typically over 2 days—in a facilitated modeling session with business experts from relevant project areas of the enterprise. In Chapter 7 we discuss several strategic models developed in 2-day facilitated modeling sessions for a number of organizations.</td>
</tr>
<tr>
<td>From a strategic model defined in a 2-day facilitated modeling session, an enterprise architecture portfolio plan report is developed. The EAPP report identifies priority enterprise architecture areas for rapid delivery and implementation. For small and medium enterprises, the EAPP report is typically developed and documented in a total elapsed duration of 4 weeks, including the 2-day facilitated modeling session to develop the strategic model. For large enterprises, the development and documentation of the EAPP report typically takes 8–12 weeks because of the greater enterprise complexity.</td>
</tr>
<tr>
<td>The EAPP report is developed using entity dependency analysis methods covered in Chapter 7. These project plans are the basis for later development of tactical and operational data models (in column 1) and activity models and process models (in column 2) of the Zachman framework. This leads to rapid implementation of priority systems for the priority project areas.</td>
</tr>
<tr>
<td>With this analysis and the technologies covered in Part III, priority business activities and processes (as priority project areas) can typically be delivered into production in 3-month increments for small and medium enterprises. For large enterprises, these priority areas are delivered into production in 6-month increments due to greater enterprise inertia.</td>
</tr>
</tbody>
</table>
prise glossary that we discussed in Box 1.2. The enterprise glossary is the language
dictionary of a business, similar to the translation dictionaries used with different
spoken languages.

We discussed that data modeling methods are used by enterprise engineering.
These methods are described extensively in Part II. They define metadata. Their use
is vital to achieve business integration. Their use is also vital to implement enterprise
architecture for business transformation.

To illustrate the problems that arise from a lack of definition of enterprise
metadata, we will consider a hypothetical enterprise in Box 1.3: XYZ Corporation.
XYZ is a sales and distribution organization that purchases products and services

<table>
<thead>
<tr>
<th>Box 1.2: A Nontechnical Introduction to Metadata</th>
</tr>
</thead>
</table>
| Consider how we communicate by phone. Because all countries are interconnected by the global phone
| network, we can dial any number at random and a phone will likely ring somewhere in the world. However,
| if the person who answers it speaks a different language, communication may not be possible. But by
| using an interpreter or a translation dictionary, we are able to communicate regardless of the spoken lan-
| guage.
| Now consider that different “language” or terminology may be used in various parts of a business. We
| call this jargon. For example, finance people and engineering people may not understand each other
| because they use different terms to refer to their areas of knowledge.
| Consider also that different terms can mean the same thing in various parts of the business, such as
| “customer,” “client,” and “debtor.” These words are synonyms. They are used, respectively, by the sales
| department and order entry department, by the credit control department, and also by the finance depart-
| ment. Each synonym refers to a buyer of products or services from an enterprise. To communicate most
effectively, a common term must be agreed on and its exact meaning defined and documented so that all
parties know what that term means. A more appropriate definition of metadata follows:
| Metadata documents an organization’s terminology and meanings. It documents the enterprise language
typically as an enterprise glossary of terminology.
| This glossary is the enterprise equivalent of a translation dictionary as discussed earlier. |

<table>
<thead>
<tr>
<th>Box 1.3: XYZ Case Study Example</th>
</tr>
</thead>
</table>
| Both the sales department and the order entry department of XYZ accept orders from “customers.” They
| keep details of customers in a database table called CUSTOMER.
| The credit control department keeps similar details, but it uses different terminology. It refers to people
| who buy from XYZ on credit as “clients,” not customers. Details are kept in a CLIENT table.
| The accounts receivable section in the finance department uses different terminology also. It calls the people
| who pay for orders “debtors,” with details kept in a DEBTOR table. Multiple copies of each organization’s address are stored in these tables.
| If an organization that deals with XYZ as a customer, client, or debtor is also a supplier of products, then
| the purchasing department uses different terminology yet again. Such organizations are “suppliers,”
| with details kept in a SUPPLIER table. Payment by XYZ of the supplier’s account balance is managed by
| accounts payable in the finance department, who call these suppliers “creditors,” with details kept in a
| CREDITOR table.
| If an organization—known to XYZ variously as a customer, client, debtor, supplier and creditor—later
| changes its address, all of these address copies must be updated and synchronized so that they contain the
| same changed address. |
from its suppliers to sell to its customers. We will use this as a case study example throughout the book.

Redundant data present no problem if their values do not change. But if data values are volatile and hence can change—such as an address—then every redundant version of the address in each database table must be changed to contain the latest correct value.

We can see the problem that arises if different terminology is used throughout XYZ. Each of these terms represents the same organization: a “customer”—for sales and order entry; a “client”—for credit control; a “debtor” or “creditor”—for finance; a “supplier”—for purchasing. These synonyms all identify a buyer of products and services from XYZ, or a supplier to XYZ. To communicate effectively, a common term must be agreed on and its exact meaning defined and documented so that all involved understand what that term means.

The preceding example considers the various roles that an organization can take in dealing with XYZ. A common term should be used throughout XYZ Corporation. We will discuss this further in Chapters 6, 7 and 9 when we discuss data modeling methods to identify metadata.

1.5 Summary

The summary is as follows.

- We discussed the need to transform from today’s inflexible business environment to an enterprise that can change direction rapidly. Methods and technologies are needed for rapid business change—with systems that change in lock-step.
- Business change depends on enterprise integration. This includes business integration using enterprise architecture methods to define integrated data and reusable business activities and processes. Enterprise architecture methods are covered in Part II of this book.
- Enterprise integration also includes technology integration, that is, the process of using technologies to deliver integrated data and reusable processes rapidly into production as shared databases and systems. Enterprise architecture technologies are covered in Part III of this book.
- The problem is that today we have twenty-first-century enterprises that use twenty-first-century technologies … yet most enterprises today still use eighteenth-century disintegrated business processes!

The business processes—originally designed based on principles set by Adam Smith in 1776 as discussed in the preface—have not evolved to take advantage of the technologies we have today. We need integrated twenty-first-century enterprises together with transformed twenty-first-century processes!

- We discussed the problem of redundant data versions in most enterprises. When data values change, all redundant versions must be updated to synchronize with that change. With redundant data, we moved to data maintenance chaos!
• We saw that data modeling methods define metadata. Their use is vital for business integration. Their use is also vital to implement enterprise architecture.

• We discussed concepts of the Zachman framework for enterprise architecture. We discussed that the real architects of an enterprise are the senior business managers who set strategic directions for the future, based on business plans and strategies, and processes designed for that future and its technologies.

• We discussed enterprise reusability. We saw that the Planner and Owner rows of the Zachman framework are critical for reusability. These two rows enable reusability opportunities to be identified within an enterprise.

• We discussed the concepts of enterprise engineering, which is used to identify reusability opportunities based on business plans for rapid delivery of priority areas into production.

Summarizing enterprise engineering as used with the Zachman framework, the preferred way to implement for the needs of the future follows:

• We must design for tomorrow based on business plans for the future. We should use activities and processes tailored for the environment of the Internet—which represents our present and our future—so that enterprises can respond rapidly.

• Enterprise architecture should therefore first address rows 1 and 2, from the perspectives of the Planner and the Owner.

• Column 6 (Why) defines the business plans for the future. These plans are an important starting point.

• Column 4 (Who) is used to identify the business managers and business experts responsible for implementing the business plans.

• The business experts are used to identify the data needed for the future in column 1 (What). They also identify activities and processes in column 2 (How).

• Clear directions can then be provided to row 3 (for the Designer), row 4 (for the Builder), and row 5 (for implementation by the Subcontractor).

• The result is the development of flexible systems based on the needs of the future, to be implemented rapidly using Internet technologies and tools.

Endnotes


[3] The complete Zachman framework is available as a PDF file on the accompanying CD-ROM. Print this now, in color if possible, and use it for reference as you read through the book. This PDF file has been included with permission from Intervista Institute (http://www.intervista-institute.com).