

# Introduction to Enterprise Architecture

By Clive Finkelstein

## Contents

---

Introduction to Enterprise Architecture .....	1
Contents .....	1
Introduction .....	1
Why is Enterprise Architecture Important? .....	1
Systems Development for the 21 <sup>st</sup> Century Enterprise.....	3
The Need for Business Transformation Enablement.....	4
Enterprise Architecture Concepts .....	6
Clive Finkelstein .....	14

---

## Introduction



You may know me through the monthly column: “The Enterprise” that I previously wrote for DM Review magazine in the USA, or through my own newsletter, “The Enterprise Newsletter” (TEN) that I publish quarterly via email. You can read past issues of TEN at [www.ies.aust.com/](http://www.ies.aust.com/). Click on the **Papers** link or click on the **TEN Archive** link from any page. Past issues of TEN can be read online, or can be printed as PDFs. Use any issue of TEN to subscribe, if you want the free, quarterly TEN newsletter to be emailed to you.

I was recently approached by BEye Network to write articles for a monthly online column, which I agreed to. This is the first of a series of articles that I propose to write on Enterprise Architecture (EA). For these articles, I will draw on material from my recently published book: “[Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies](#)”, by Clive Finkelstein, Artech House, Norwood MA (March 2006). I will also draw on material from the public and inhouse courses that I present world-wide on rapid delivery EA methods and technologies. You can read a review of the book from <http://www.ies.aust.com/ten/ten32.htm>. This review can also be used to purchase the book if you wish to read more.

This first BEye Network article will start with a brief introduction of why Enterprise Architecture is important; I will discuss how it can help organizations to survive and prosper in the rapid business change environment of today. In later months, I will follow with articles on case studies of EA projects from commercial and government enterprises to show the application of EA principles to the problems of business.

## Why is Enterprise Architecture Important?

The most critical issue facing Government, Defense and Commercial enterprises today is the rapid pace of change in almost every industry. With the rate of technological change increasing, together with today’s budget and competitive pressures, enterprises must be able to change rapidly ... often just to survive - let alone succeed.

The need for transformation from today’s inflexible business environment to an agile enterprise that can change direction rapidly has never been greater. Yet the structures, processes and systems that we have today are inflexible: they are incapable of rapid change. More computer hardware, or software, or packages, or staff, or outsourcing are not the solution. They are part of the problem.

The solution needs methods for rapid business change – with systems that also change in lock-step. This is not a computer problem. It is a business problem. It needs strategic direction from senior management and strategic planners, with these directions then translated into rapid action by business experts working with IT experts.

The solution needs skills that enable senior managers, together with their planners, business managers, business experts and IT staff to work together to determine how to transform your enterprise – as each group has its specific expertise to contribute.

The methods to achieve this are being successfully applied by many enterprises today. As we will see in case studies over the coming months, these methods require new thinking. The tried and true ways no longer work. We need new ways to make the required business transformations.

Our current systems development methods have served us well in building, analyzing and drilling down through existing data resources with data warehouses, using OLAP and data mining tools. They have also served us well in developing operational information systems in the period of managed change that we had up until the 90s. But now the pace of change is much, much faster than we ever anticipated when those systems and data warehouses were first built.

Historically these systems and warehouses have been difficult to change. The systems and databases that we built in the early years of the Information Age to enable our organizations to be more responsive to change are now monolithic and resistant to change. Today, they inhibit the ability of our organizations to change rapidly in order to compete ... sometimes even to survive. We are chained to inflexible systems that can no longer respond to the rapid change environment of today – let alone the even greater change environment that we will soon find ourselves in tomorrow.

We need to build more flexible systems for the future that can change easily, rapidly, and often. To achieve this, the systems development methods that we use should take a different focus for the future. They must be able to identify potential future changes early. We must also build systems and databases differently ... so that they can be changed rapidly to support vital business changes. These changes must be capable of being made within weeks, even days – not in years as is the case today.

I will discuss some of the reasons for the dilemma that most enterprises find themselves in today:

### **Systems Development for the 21<sup>st</sup> Century Enterprise**

Our systems development methods have served us well so far, but we have now reached a point where we need to decide whether to take a different focus for the future of the 21<sup>st</sup> Century Enterprise.

### **The Need for Business Transformation Enablement**

There are fundamental issues that we need to address in the enterprise. An understanding of these will help us to determine whether the enterprise needs to be transformed, and if so ... how to achieve this transformation.

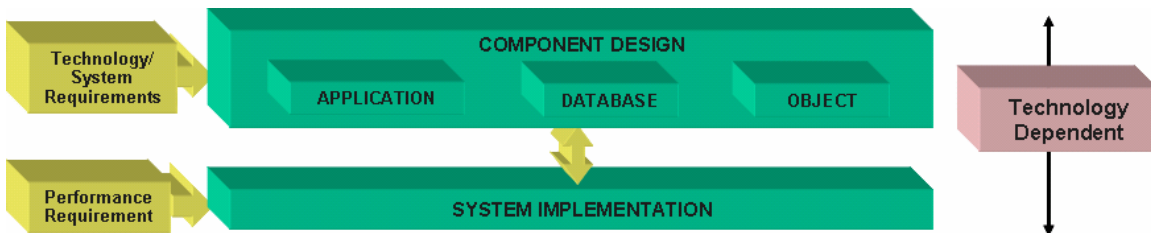
### **Enterprise Architecture Concepts**

I will discuss the concepts and methods of Enterprise Architecture and we will see how this discipline can assist business and IT in deciding how to go about business transformation.

## Systems Development for the 21<sup>st</sup> Century Enterprise

I will first discuss some typical systems development problems. The approach used to design and build enterprise systems with traditional systems development methods is summarized below and illustrated in Figure 1:

- 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. The technology is used to deliver the desired business benefits.



**Figure 1: Traditional systems development methods are very technology-dependent**

- These designs are then implemented to meet desired business Performance Requirements.
- This traditional approach to systems development has been *Technology-Dependent*.

But the traditional approach to systems development 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.

Problems with traditional development methods are even greater than suggested earlier. 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 90s 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. Fact has indeed been stranger than fiction, here.

The strategic plans defined in the 90s 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.

Competition today – in Government or Defense such as for funding resources – demands systems that can change easily to support rapid business change. Many of these 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 by developing new systems from scratch. This is slow and very costly.

The advantages and benefits of technology were not clear in the early 90s 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 Year 2000 problem in the late 90s both demonstrated to management the dramatic impact that Technology can have on the enterprise.

- We must build systems that are designed to provide support for a future where *the only thing that remains constant ... is change itself.*

Businesses must change, to compete with other organizations in their relevant markets. This is true for Commercial organizations, which compete with other organizations. It is true for Government Departments that compete with other Departments for government budget funding. And it is also true for Defense, which competes with hostile Defense forces, and also with friendly Defense forces for limited resources.

This brings us to three very important principles 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 based on the strategic plans that define what tomorrow will be.*
- *Our systems must be 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.*
- *Our systems must be designed and built so they can accommodate rapid change, if they are to be able to support the rapid pace of change that enterprises are experiencing today and tomorrow. Our systems need to be composed of “logic leggo building blocks” that can be assembled by point and click methods as process models or workflow models to generate executable code automatically. This will enable process changes to be made easily to accommodate changes in the business.*

## ***The Need for Business Transformation Enablement***

Technology alone is not the answer. To achieve any degree of success in enterprise integration requires that *Technology Integration* be used within a coherent, integrated enterprise, through *Business Integration*. But we still have a long way to go in most enterprises to realize business integration.

To appreciate what still has to be achieved, we need to review what I call the “*Process Engineering Bible*”. I describe the book in this way as it has had a dramatic effect in the way that organizations function. To consider its impact, we need to review its message. But first:

- What is its title?

- Who was the author?
- When was it published?

Perhaps we can identify the book by first considering its author:

- Was it Michael Hammer or James Champy: of *Reengineering the Corporation* fame? No, it was neither of them.
- Was it Ken Orr, Ed Yourdon or Tom de Marco: of *Software Engineering* fame? No, it was not them either.
- Well, was it Peter Drucker: of *Management and Strategic Planning* fame? No, not him.
- Was it Edwards Demming: of *Quality Control* fame? No, not him either.
- Was it Alfred Sloan, or Henry Ford? No, the book I am referring to was published long before all of these eminent people.

I have provided an extract in Box 1 from the initial pages of the book, in his own words – a quaint writing style, based on today’s books. I have included in brackets the technical terminology that we use today when referring to the environment he is talking about. He is using different words, but he is also talking about concepts that today are associated with Object-Oriented methods.

**Box 1: Extract from what I call the “Process Engineering Bible”**

“To take an example, therefore, from a very trifling manufacture; but one in which the division of labour has been very often taken notice of, the trade of the pin-maker ... a workman ... could scarce ... make one pin in a day, and certainly could not make twenty. (*today’s terminology: serial operation*)

But in the way in which this business is now carried on, not only the whole work is a peculiar trade, but it is divided into a number of branches, of which the greater part are likewise peculiar trades. (*today’s terminology: object-oriented methods*)

One man draws out the wire, another straightens it, a third cuts it, a fourth points it, a fifth grinds it at the top for receiving the head; to make the head requires two or three distinct operations (*object-oriented encapsulation*); to put it on is a peculiar business, to whiten the pins is another; it is even a trade by itself to put them into the paper; and the important business of making a pin is ... divided into about eighteen distinct operations ... (*object-oriented methods*)

I have seen a small manufactory of this kind where ten men only were employed ... they could, when they exerted themselves, make among them about twelve pounds of pins in a day ... upwards of forty-eight thousand pins in a day. Each person, therefore ... might be considered as making four thousand eight hundred pins in a day. (*object-oriented reusability*)

But if they had all wrought separately and independently... they certainly could not each of them have made twenty, perhaps not one pin in a day (*through serial operation*) ... ; that is, certainly, not ... the four thousand eight hundredth part of what they are at present capable of performing, in consequence of a proper division and combination of their different operations. (*object-oriented reusability*)”

So which book am I talking about? As soon as I give you the author and its title – with its publication date – its significance will become apparent. The reference is as follows ...

- The extract is from: *Adam Smith, "Wealth of Nations", (1776)*

Adam Smith's breakthrough was the foundation for late 18th Century – early 19th Century industrial age enterprises. With the focus at that time on manufacturing physical items, this period also saw the same concepts applied to knowledge-based processes for bank loans and for insurance policy applications.

By the middle of the 20th Century, the Industrial Enterprise had evolved into a complex series of manual processes. The pace of progress had seen most enterprises evolve to use increasingly complex business processes, with rapidly growing transaction volumes to be manually processed. And what was the result?

- These enterprises found they were operating in a continual state of *Manual Chaos!*

Starting from the late 1950s – through the 60s, 70s, and right up to today – we saw manual processes being automated by computer. The processes were automated, but we took the existing manual processes and then automated them essentially AS-IS: without much change. That is, the automated processes were being executed as for the manual processes, but faster, and more accurately.

- In so doing, *we moved from manual chaos ... instead to automated chaos!*

The problem is much worse than this! Most automated processes today assume that the technologies of the past still apply. The manual processes that they automate required paper-based forms that were mailed, or later faxed. So their automated counterparts are based on forms that are also printed to be mailed or faxed. On receipt at their destination, the data in these forms are manually reentered into relevant systems: with manual work; with extra staffing to do that reentry; with delays; with errors; and with associated costs.

The problem is that automated systems that assume inter-communication with printed forms and manual reentry over weeks and days do not work well when asked to inter-communicate with electronic forms that bypass the need for manual reentry – and that complete in minutes, or seconds across the Internet. And the reason?

- *Today we have 21st century enterprises that utilize 21st century technologies ...*
- *Yet most enterprises today still use 18th century non-integrated business processes!*

## ***Enterprise Architecture Concepts***

I will now discuss the concepts and methods of enterprise architecture and we will see how this discipline can assist business and IT in deciding how to go about business transformation, utilizing the most effective rapid delivery enterprise architecture methods and SOA technologies.

Enterprise architecture was initially 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 as illustrated in Figure 2 and discussed following the figure.

## DIFFERENT PERSPECTIVES

Buildings	Airplanes	Enterprise
Architect's Drawings	<b>OWNER</b> Work Breakdown Structure	Model of Business
Architect's Plans	<b>DESIGNER</b> Engineering Design	Model of Info System
Contractor's Plans	<b>BUILDER</b> Manufacturing Engineering Design	Technology Model

**Figure 2: Enterprise Architecture addresses different perspectives**

- The *Owner* in the Building industry uses *Architect's Drawings* to determine if the building will address specific requirements. For Airplane manufacture, this is achieved by the high-level *Work Breakdown Structure* of the plane. A *Model of Business* is needed for Information Systems.
- 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* or blueprints for construction of the building; *Manufacturing Engineering Design* for plane construction; or *Technology Models* for information systems.

In addition, there are a number of different questions – called *Primitives*, *Interrogatives* or *Abstractions* – that need to be considered, shown as columns in Figure 3.

## DIFFERENT ABSTRACTIONS

WHAT	HOW	WHERE
Material	Function	Location
Bill of Materials	Functional Specifications	Drawings
Data Models	Functional Models	Network Models

**Figure 3: Enterprise Architecture addresses Primitive Interrogatives or Abstractions**

- WHAT is needed is important to know. This is represented above by *Material*, such as *Bill 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 as shown in Figure 3. These represent the perspectives for each of:

- The *Planner*, who is interested in *What, How* and *Where*
- The *Owner*, who is also interested in *What, How* and *Where*
- The *Designer*, who is interested in *What, How* and *Where*
- The *Builder*, who is interested in *What, How* and *Where ...* and
- The *Subcontractor*, who also is interested in *What, How* and *Where*

The last Row in Figure 4 addresses the Final Structure: the final product to be produced.

	What	How	Where
Planner			
Owner			
Designer			
Builder			
Subcontractor			
Final Structure			

**Figure 4: The first three columns of the Enterprise Architecture Framework**

Different documentation, models or representations may also be utilized in each cell of the Zachman Framework as shown in Figure 5. For example, reading down Column 1 – WHAT (*Data*) 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 strategic models in next month’s article.



- The cell for the *Designer* row and the *Data* column shows that “*Logical Data Model*” documentation is appropriate for this cell.
- The *Builder* row and *Data* column cell contains the “*Physical Data Model*”
- The *Subcontractor* row and *Data* column cell contains physical “*Data Definition*” scripts for installation of databases.

	<b>WHAT Data</b>	<b>HOW Process</b>	<b>WHERE Network</b>
<b>Objectives/ Scope</b> <i>Planner</i>	List of Things	List of Processes	List of Locations
<b>Enterprise Model</b> <i>Owner</i>	Conceptual Enterprise Model	Business Process Model	Business Logistics System
<b>System Model</b> <i>Designer</i>	Logical Data Model	Application Architecture	Distributed System Architecture
<b>Technology Model</b> <i>Builder</i>	Physical Data Model	System Design	System Architecture
<b>Detailed Representations</b> <i>Subcontractor</i>	Data Definition	Program	Network Architecture
<b>Functioning Enterprise</b>	Data	Function	Network

**Figure 5: Different models apply to each cell**

Reading down Column 2 – HOW (*Process*) and Column 3 – WHERE (*Network*), we also see in Figure 5 that each row in these columns has various representations in the cells for those columns as well. Several types of models may also be relevant to each cell.



- Artifacts are defined by independent variables, represented by Columns based on the Primitive Interrogatives:
  - What, How, Where, Who, When, Why
- Rows are different Views of Enterprise:
  - From Perspective of : Planner, Owner, Designer, Builder, Subcontractor (or Out-of-Context specification)

ZACH-MAN framework.wmv [1m58s]

	What	How	Where	Who	When	Why
PLANNER Objectives/Scope						
OWNER Conceptual						
DESIGNER Logical						
BUILDER Physical						
SUBCONTRACTOR Out-of-Context						
FUNCTIONING ENTERPRISE	Data	Function	Network	Organization	Schedule	Strategy

**Figure 6: The six columns of the Zachman Framework for Enterprise Architecture**

John Zachman makes the point in Figure 6 that for all things we consider in business or day-to-day life – whether for buildings, for planes or for complex enterprise systems – there are actually six independent variables. These are based on six Primitive Interrogatives: *What, How, Where, Who, When, Why*.

There are therefore a further three columns: WHO; WHEN; and WHY in the complete *Zachman Framework for Enterprise Architecture*. These questions are shown in Figure 6 by all six columns in the Zachman Framework.

The Perspectives of the *Owner, Designer* and *Builder* were discussed earlier. The Perspective of the *Planner* defines the scope, while the *Subcontractor's* Perspective is to focus on components.

- The result is a matrix of five rows and six columns, for a total of thirty cells as shown above in Figure 6.

	3	5	6	2	4	1
	What Data	How Function	Where Location	Who People	When Time	Why Future
PLANNER Objectives/Scope	List of Things	List of Processes	List of Locations	Org Structure	List of Events	List of Goals/Obj
OWNER Conceptual	Enterprise Model	Activity Model	Business Logistics	Work Flow	Master Schedule	Business Plan
DESIGNER Logical	Logical Data Model	Process Model	Distrib. Architect.	Human Interface	Process Structure	Business Rules
BUILDER Physical	Physical Data Model	System Model	Technol. Architect.	Presn Interface	Control Structure	Rule Design
SUBCONTRACTOR Out-of-Context	Data Definition	Program	Network Architect.	Security Interface	Timing Definition	Rule Specs
FUNCTIONING ENTERPRISE	Data	Function	Network	Organization	Schedule	Strategy

**Figure 7: Models that apply to all 30 cells of the Zachman Framework for Enterprise Architecture**

Figure 7 shows examples of typical model contents for each cell. For example, the How column (Col 2) shows that an *Activity Model* is relevant to the Owner row (Row 2). As a further illustration, Col 2 Row 3 – a cell of interest to the Designer – shows that a *Process Model* is relevant for this cell.

These are defined ideally in the sequence indicated by the numbered columns in Figure 7:

1. Col 6 (Why) focuses on the business plans for the future, to understand the directions for that future.
2. Col 4 (Who) identifies managers in the enterprise responsible for implementing the business plans. It identifies the staff who will help them (as business experts) to transform the enterprise for that future.
3. Col 1 (What) uses the managers and business experts from Col 4 to help identify the data and information needed to support the enterprise based on its transformation plans for the future.

4. Col 5 (When) identifies major business events that initiate business activities. This focus on business events based on the business plans can be used to identify reusable business activities. These can have the greatest potential transformation impact on the enterprise.
5. Col 2 (How) addresses business activities identified from Col 5 and also Col 1. It defines activity models and then uses activity based costing to evaluate costs and other benefits of alternative strategies for business transformation that were defined in Col 6.
6. Col 3 (Where) then addresses the locations of the enterprise to participate in this transformation.

In summary, the Framework rows indicate different Views (or Perspectives) of people in the Enterprise, from the Perspectives of the: *Planner, Owner, Designer, Builder, and Subcontractor*. The Framework columns also address different primitive questions (also called interrogatives or abstractions) of *WHAT, HOW, WHERE, WHO, WHEN* and *WHY*. An understanding of these perspectives and primitives is essential to achieve managed and effective business transformation.

A high-level view of an enterprise is shown by a horizontal slice at the top 10% of each cell as shown in Figure 8.

		3 ↓	5 ↓	6 ↓	2 ↓	4 ↓	1 ↓
		What Data	How Function	Where Location	Who People	When Time	Why Future
<b>Horizontal Slice</b>	PLANNER Objectives/Scope	List of Things	List of Processes	List of Locations	Org Structure	List of Events	List of Goals/Obj
<b>Vertical Sliver</b>	OWNER Conceptual	Enterprise Model	Activity Model	Business Logistics	Work Flow	Master Schedule	Business Plan
	DESIGNER Logical	Logical Data Model	Process Model	Distrib. Architect.	Human Interface	Process Structure	Business Rules
	BUILDER Physical	Physical Data Model	System Model	Technol. Architect.	Presn Interface	Control Structure	Rule Design
	SUBCONTRACTOR Out-of-Context	Data Definition	Program	Network Architect.	Security Interface	Timing Definition	Rule Specs
	FUNCTIONING ENTERPRISE	Data	Function	Network	Organization	Schedule	Strategy

**Figure 8: High-level priority areas are identified as “vertical slivers” for rapid delivery**

- Col 6 (Why) addresses Business Plans with Row 1 (Planner) and with Row 2 (Owner). Business managers and business experts from Col 4 (Who) that are responsible for these high-level plans are used to identify high-level data, activities and processes that are required to implement those plans within the enterprise.
- Col 1 (Data) with Row 1 (Planner) shows a “List of Things” as a high-level horizontal slice in that cell. Col 1 (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. I will cover strategic models next month.

- Furthermore, horizontal slices in Col 2 (How) for Row 1 (Planner) and for Row 2 (Owner) represent a high-level “List of Processes” and high-level “Activity Models”. This list is then used to identify and define Activity Models in Col 2, Row 2.

The high-level focus of the horizontal “slice” at the top of each cell (as discussed above) enables priority areas to be identified by management for transformation – that need to be implemented first; these are called “vertical slivers” that extend for the depth of each cell.

From the Planners’ and Owners’ perspectives in Rows 1 and 2 we can see that vertical slivers in each cell enable greater detail to be defined in priority areas, typically in the sequence as numbered in Figure 7.

These areas progress to detailed definition: this is represented by the depth of the vertical sliver in each cell, for rapid implementation using appropriate tools and technologies as I will discuss in later columns covering Web Services and SOA. Thus systems and processes for priority business transformation areas can be delivered early: before other, less important areas that can wait until later.

Rows 1 and 2 – from the perspectives of the Planner and the Owner – are critical. These two rows are used to identify reusability opportunities for transformation within an enterprise. Figure 9 highlights these two rows for reusability using the Zachman Framework.

**Reusability Definition**

	What Data	How Function	Where Location	Who People	When Time	Why Future
PLANNER Objectives/Scope	List of Things	List of Processes	List of Locations	Org Structure	List of Events	List of Goals/Obj
OWNER Conceptual	Enterprise Model <b>Key</b>	Activity Model <b>Key</b>	Business Logistics <b>Key</b>	Work Flow <b>Key</b>	Master Schedule <b>Key</b>	Business Plan <b>Key</b>
DESIGNER Logical	Logical Data Model	Process Model	Distrib. Architect.	Human Interface	Process Structure	Business Rules
BUILDER Physical	Physical Data Model	System Model	Technol. Architect.	Presn Interface	Control Structure	Rule Design
SUBCONTRACTOR Out-of-Context	Data Definition	Program	Network Architect.	Security Interface	Timing Definition	Rule Specs
FUNCTIONING ENTERPRISE	Data	Function	Network	Organization	Schedule	Strategy

Identify priority processes for delivery into production in 3 - 6 month increments

**Figure 9: Reusability is defined in the Planner and Owner rows**

A number of cells in these rows of Figure 9 are vitally important, in the sequence below:

1. *Business Plans* in Col 6 (Why) are *Key* to setting directions for the future.
2. Col 4 (Who) identifies the *Organization Structure* in Row 1. This is also *Key*, as it enables key managers with *Workflow* knowledge to identify relative priorities based on the *Business Plans*.
3. The *Business Plans* from Col 6 are used to develop a high-level *Strategic Model* in Col 1 (What). This is a *Key* cell as it is vital in identifying high-level data and information that are needed to manage the progress of the enterprise towards the future.

- Col 5 (When) shows that a *List of Events* and the *Master Schedule* are also Key. This can be used to identify potential reusable activities in Col 2 (How).
- Activity Models* in Col 2 (How) are also vital: these *Activity Models* are *Key* to identifying critical reusable activities that should be carried out by the enterprise in the future.
- Col 3 (Where) representing the *Business Logistics* of the enterprise is also *Key*.

Rapid delivery of priority areas – represented earlier by “vertical slivers” – is realized by key technologies as illustrated by the highlighted, numbered bands in Figure 10, discussed after the figure:

		Automatic Generation	Reusability Definition				
		What Data	How Function	Where Location	Who People	When Time	Why Future
PLANNER	Objectives/Scope	List of Things	List of Processes	List of Locations	Org Structure	List of Events	List of Goals/Obj
<b>Reusability Definition</b>							
OWNER	Conceptual	Enterprise Model <b>Key</b>	Activity Model <b>Key</b>	Business Logistics <b>Key</b>	Work Flow <b>Key</b>	Master Schedule <b>Key</b>	Business Plan <b>Key</b>
DESIGNER	Logical	Logical Data Model <b>1</b>	Process Model <b>3</b>	Distrib. Architect.	Human Interface <b>2</b>	Process Structure	Business Rules
BUILDER	Physical	Physical Data Model	System Model	Technol. Architect.	Presn Interface	Control Structure	Rule Design
SUBCONTRACTOR	Out-of-Context	Data Definition	Program	Network Architect.	Security Interface	Timing Definition	Rule Specs
FUNCTIONING ENTERPRISE		Data DBMS DDL Generation	Function DB Code Patterns	Network	Organization BPM Generation	Schedule	Strategy

Identify priority processes for delivery into production in 3 - 6 month increments

**Figure 10: Rapid Delivery Technologies for Enterprise Architecture**

- Col 1 (Data) uses data modeling methods to develop an enterprise model as a strategic model in Row 2. This is expanded into greater data detail as a logical data model in Row 3. Today, most enterprises use modeling tools to develop the physical database design in Row 4 and data definition language in Row 5 for installation using target DBMS products. Row 5 is now automatically generated as DBMS DDL scripts by most modeling tools today.
- Workflow models in Col 4 Row 2 are the focus of an emerging class of Service-Oriented Architecture (SOA) business process management (BPM) languages. These include: business process execution language for web services (BPEL); web services choreography interface (WSCI), business process modeling language (BPML) and ebXML business process specification schema (BPSS). These are all XML-based executable languages that are automatically generated from workflow models for immediate execution. I will cover these BPM languages in later articles.
- Database code pattern generators are available today that automatically generate >80% of the code that previously was 100% written by hand. This code is generated from DDL scripts that define the structure of a database. These database code patterns are generic for all programs that need to access and manipulate these databases. They can be generated in a variety of languages, including Visual Basic (VB-6 and VB .Net), Active Server Pages (ASP and ASP .Net) and C# .Net for example.

The result is rapid generation and delivery of databases and systems into production in a fraction of the time that was previously required. These technologies enable priority business activities or processes representing transformation areas (identified earlier in Figure 8 as vertical slivers) to be delivered typically in 3-month increments. I will cover all of these rapid delivery technologies in later articles.

Further, the resulting systems are far more responsive to change than the traditional systems they replace. Business changes can be made by point-and-click changes directly to the data models and workflow models, which are then regenerated, resulting in systems that can rapidly change to support required business changes. These utilize web services, which are the “logic leggo building blocks” that I referred to at the start of this article.

I have now covered the fundamental principles that underpin enterprise architecture. In my article next month I will introduce the concepts of strategic modeling. I will follow that with a case study of a rapid delivery enterprise architecture project for a bank in the next month. After that, I will describe another enterprise architecture project completed for a government department.

### **Clive Finkelstein**



Clive Finkelstein is acknowledged worldwide as the "father" of information engineering, and is Managing Director of Information Engineering Services Pty Ltd in Australia. He has more than 45 years of experience in the computer industry. Author of many books and papers, his latest book, *Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies*, brings together the methods and technologies for rapid delivery of enterprise architecture in 3-month increments. Read the book review at <http://www.ies.aust.com/ten/ten32.htm>. Project references, project steps and descriptions are available from <http://www.ies.aust.com>. Click on the **Projects** link from any page. Clive may be contacted at [cfink@ies.aust.com](mailto:cfink@ies.aust.com).