論文 / 著書情報
Article / Book Information

題目(和文)	都市計画アプローチによる企業情報システムの再構築
Title(English)	City planning approach for rebuilding enterprise information systems
著者(和文)	南波幸雄
Author(English)	
出典(和文)	学位:博士（学術）, 学位授与機関:東京工業大学, 報告番号:甲第6207号, 授与年月日:2005年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor of Philosophy, Conferring organization: Tokyo Institute of Technology, Report number:甲第6207号, Conferred date:2005/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

City Planning Approach
for Rebuilding Enterprise Information Systems

Yukio Namba

Submitted in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Supervised by: Professor Junichi Iijima
Graduate School of Decision Science and Technology
Tokyo Institute of Technology
January, 2005

Acknowledgements

I would like to express my sincere gratitude to Professor Junichi Iijima, whose insightful advice and guidance have been indispensable throughout my period of study at Tokyo Institute of Technology. Thanks are also due to the other members of my thesis committee—Professors Motoshi Saeki, Norihiro Nakai, Kunihiro Higa Dai Senoo, and Hiroyuki Umemuro—who have offered valuable suggestions throughout the composition process. In addition, I would like to thank Ms. S. Hou and the members of Iijima laboratory for their warm support.

I would also like to express my appreciation to Mr. Ayumi Teshima and the late Mr. Hiromichi Iwata for their thoughtful advice and encouragement. They have strongly supported this research from the standpoint of object orientation and architectural planning. I deeply appreciate the help of Mr. Takayasu Shigeno and the involved parties of KDDI, and thank Mr. Tshuji and Mr. Oda of Ebara Corporation for their case provision and practical advice. I also express my thanks to Mr. James P. Brown for his suggestions regarding English composition.

Finally, I wish to thank my wife Eimi and my sons Keita and Koji for their warm and continuing support and patience over the past two years.

Yukio Namba

January 2005

Contents

Chapter 1

Introduction: Current Status and Issues for Firms' Information Systems

1.1 Background of the Study

1.1.1 Changing Business Environment and Information Technology

Rapidly changing business climates force firms to change toward a flexible and agile management that is capable of adapting to a changing market. Firms must realize that information systems are one of the most significant challenges in sustaining a business and maintaining a competitive edge. To cope with this, firms' information systems must provide support so that business can adapt to change with flexibility.

For the business domain, the major changes are expressed with specific keywords as follows:

- Extension of operational boundaries of firms as a supply chain management (SCM) or a business-to-business electronic commerce (B2B).
- Restructure of firms themselves through business process re-engineering (BPR).
- Merger and acquisition (M&A) of the firms themselves.
- Formation of closer relationships with customers through customer relationship management (CRM).

On the other hand, for the domain of information technology (IT) and the IT environment, which are the foundation of firms' information systems, changes are as follows:

- Speed of technologies progress is symbolized by Moor's Law.
- Diversification of technology utilization is symbolized by Ubiquitous Technologies.
- Progress and diffusion of Open Standard Technology is backed by the Internet world.
- Rapid expansion of the Internet is occurring both as a backbone network and also in terms of the ratio of the broadband network to the home.
- Emergence of a public infrastructure such as the Internet and various other shared services is taking place, along with the emergence of common frameworks such as RosettaNet and ebXMI.

Both the changing business environment and the progress of IT have driven, enabled, and accelerated the pace of change in these areas for key business transformation. As a result, possessing the ability to adapt to such change is essential if a firm is to maintain a competitive business advantage.

1.1.2 Reality of Firms' Information Systems

Since business requirements have always been both urgent and mandatory, more than a few firms' systems have been maintained and integrated using ad hoc or makeshift solutions, with little regard to the long-range effects. Accordingly, these firms are burdened with increasing systems complexity, resulting in high maintenance costs and insecure structures. This embroiled aspect of information systems is termed "spaghetti," a term initially used for programs with confusing control structures and for a state of uncontrollable applications. To make things worse, the term is now used to describe systems beyond the single-application level. IBM (1999) mentions, "The problem gets worse as you try to connect more and more applications…This type of solution has built-in inflexibility and a high cost of maintenance and we characterize it as inter-application spaghetti."

Belady and Lehman (1976), studying the history of successive releases of a large operating system (OS/360), found that the total number of modules increases linearly with each release number, but that the number of affected modules increases exponentially with each release number. All repairs tend to destroy the structure, increasing the entropy of the system, unless specific measures are executed to maintain or reduce disorder. This law of entropy may apply beyond single applications and operating systems, expanding to a firm's overall systems. Therefore, maintenance or enhancement makes systems more complex and degrades their structures unless proper measures are followed. To reduce systems complexity, application design, development techniques, and management methods need to meet and adapt to new business and IT environments.

By contrast, businesses require the connection of more and more application functionalities, not only intra-enterprise but also inter-enterprise, in accordance with extensions of the business domain, such as SCM, B2C and B2B. Consequently, firms' systems must cover all operational boundaries, extending beyond their legal boundaries. This results in making a firm's information systems more rigid and thus harder to maintain.

According to Namba and Iijima (2004a), many firms have implemented, or are now implementing, an integrated software package such as enterprise resource planning (ERP) to rebuild their systems. This approach represents one example of the move toward commercial-off-the-shelf-software (COTS). Since ERP is intended to be an all-in-one solution, firms receive maximum benefit if they implement all the functionalities. ERP solutions, however, (a) do not cover the whole activity of a firm, (b) cannot contain a full set of tailored sub-packages, and (c) have not offered all the functionality needed to support the business processes. As most firms implement ERP selectively, they have to complement missing functions not covered by ERP with existing architectural applications or with other newly implemented applications. Accordingly, a mechanism is required to integrate whole-system functions. In addition, many firms need to connect their applications and build interfaces for partnering with companies or customers outside of the firm. Multiple architectures in a firm's system affect maintainability and operation capability unless proper mechanisms or structures have been implemented. As firms have to involve black boxes in their information systems, they have no other choice but to manage architectural heterogeneity in a software environment that consists of various structured systems or applications. From optimal and inter-operable viewpoints, a comprehensive architectural approach that can integrate heterogeneity as a discipline is required in order to rebuild a firm's systems.

As mentioned above, most firms' systems are "built from multiple architectures, multiple designs, different vendor products, and different versions of those products" integrated with varying degrees of success "across all aspects of computer systems" (Putman, 2001). They become a combination of heterogeneous architecture and hardware/software products such as mainframe systems, client/server systems, Web applications, etc. Accordingly, firms' systems have become "a vast array of computers and applications that are linked together through a variety of ad hoc mechanisms" (Cummins, 2002). Firms are therefore burdened with the complexity of aggregated applications and information infrastructure.

1.1.3 Issues for Firms' Information Systems

Prior to discussing the required feature of the information systems, the direction of information systems should be considered. In the last ten years, the progress of information technologies has driven the transition from centralized mainframe systems to distributed client/server systems (Cummins, 2002). There are several changes related to the direction of this technology:

- The move from stand-alone or stovepipe applications to functional integrated applications,
- The move from central processing to distributed client server processing,
- The move from closed proprietary techniques to open architecture techniques

developed in the Internet world, and

- The move from "make" solutions to "buy" solutions and also from "have" solution to "use" solutions that include outsourcing.

In the process of the above mentioned change, a firm's information systems have been further accelerated to have both a complex and complicated structure. This results in the following:

- Information systems with spaghetti or inter-application spaghetti structure;
- A highly complex and complicated data structure;
- Inconsistency of data among applications due to integration of information systems developed in different places and times;
- Diversification of applied technologies for information infrastructure with adopting open technologies;
- Increasing maintenance and systems operation cost for information infrastructure; and
- Necessity of building a renewed framework for a hybrid systems operation that includes open systems, lack of know-how, and lack of human resources in this field.

In addition, firms have to connect their information systems with various affiliated or partner firms and customers through networks as well as from inside of the firm. This leads firms to deal with security issues because risks come from outside of the firm through these networks. Moreover, adapting open technologies and open networks for a firm's systems enlarges security risks. Firms have to implement effective provisions to cope with these issues, which makes Enterprise Information Systems (EIS) more complex.

Since firms' systems cannot avoid having a structure composed of an aggregation of heterogeneous architectures, they must view implementing comprehensive frameworks (meta-architecture) as a discipline based on heterogeneity. In other word, firms' systems that support businesses must have a structure (architecture) capable of quickly adapting in order to follow the speed of business. This involves a comprehensive approach not in the granularity of each information system, but in that of the level of firm, to solve the above mentioned issues.

1.2 Research Subjects and Methodology

The current status of firms' information systems and of the surrounding business environments, as mentioned above, forces firms to rebuild their information systems. This is because both business drivers and technology enablers make firms' information systems more complex and structures more complicated, unless firms take proper precautions. This dissertation tries to present ways to rebuild the firms' information systems to cope with architectural perspectives and also to propose an EIS city planning approach that grasps EIS at enterprise level and attempts to verify the applicability and effectiveness of the framework. The following four subjects are addressed:

- The necessity of IT architecture framework at the level of firm and the kind of studies that have been done in this field;
- The proposition of IT architecture framework and EIS city planning approach to withstand criticism in the prior studies;
- The discussion of the applicability and effectiveness of the framework through studying three cases; and
- The discussion of a more efficient EIS city planning approach with a post-modern city planning concept, which may break through the issues included in the concept and methodologies of current city planning.

In this dissertation, the author tries to create the IT architecture framework at the level of firm by applying the concepts and methodology of physical city planning.

The scope of this framework, however, should cover a wide area and a versatile subject matter, and the author has not had an opportunity to either develop or implement an entire firm's systems. Thus, it is difficult to verify the applicability and effectiveness of the framework directly, as the object of this study is to examine a firm's information system. Instead, a case study approach is taken. Since the author, however, did not know of a single case that covered the whole area of the framework, the following methods are adopted:

- Picking up a case that illustrates a similar concept or a similar structure to the real world.
- Extracting a success or failure factor from a case and examining a common point within the framework.
- Focusing on a structure of success factor or a state of implementation of the case, and weighing applicability and effectiveness while comparing common points or structures between the framework and the case.

In this method, the basis on which the case is selected is of importance. Thus, the author has decided the rule for selection of cases is as follows (except in the case of online securities firms, to which the author's work is directly related):

- Relevance: The case must include contents, concepts, and methodology relevant to the theme of the dissertation.
- Representation and generality: The main theme of the case should be well-known and easy to understand.
- Concreteness and openness: The case should be published in an academic journal, a commercial magazine, or be presented in a public or an academic conference, and should illustrate the contents concretely. In addition, it should be able to provide, in detail, contents with intensive interview, etc., and disclose additional information.

The theme of this dissertation pertains to information systems rather than to element technologies. According to Nagata (2001), objectively describing information systems in an article in order to illustrate effectiveness is more difficult than to do the same for element technologies. This is mainly because information systems cannot exist apart from a vast context surrounding the study. Because effectiveness is the most important measure of assessment, information systems should be examined in the context of a firm's social activities and needs to be described in logical and plain language. Thus, in this dissertation, the author follows Nagata's proposed methodology to evaluate applicability and effectiveness with specific cases.

1.3 Outline of the Dissertation

Including this introductory chapter, this study has six chapters. The subsequent four chapters correspond to each of four research subjects as mentioned above.

In Chapter 2, preceding studies on this subject are stated, which include an IT architecture framework to grasp a firm's information systems at the level of enterprise comprehensively. Then, a city planning concept that uses a metaphor to describe a firm's information systems is mentioned. After stating criticisms, the themes on this dissertation are expressed.

In Chapter 3, an EIS city planning approach composed of an EIS architecture and an

EIS scenario is introduced. EIS architecture is described with three viewpoints according to the IEEE model. This framework is characterized by employing not only a metaphor, but also the concepts and methodology of physical city planning of civil and architectural engineering.

In Chapter 4, three cases that try to prove the applicability and effectiveness of the framework are shown. These cases include KDDI Corp., Ebara Corp., and Monex, Inc. First the characteristics of each case are shown, then the concern and the applicability and effectiveness of the framework with the cases from the three viewpoints and EIS scenario described above is discussed.

In Chapter 5, additional discussion on the framework focusing on a more effective and applicable EIS city planning approach in practice is provided. First some criticism that physical city planning has not worked well is introduced. Then a renewed approach to city planning based on a new concept of post-modern city planning, which has as its prerequisite a structure that distributes autonomous parts that form a harmonious whole is shown. The direction of this study will take in the future is outlined and, finally, the structure of EIS that enables the renewed concept of the EIS city planning approach in the case of KDDI is discussed.

References

Belady, L.A., and Lehman, M.N. “A model of large program development,” *IBM Systems Journal*, Vol.15, No.3, pp.225-252, 1976.

Cummins, F. A. *Enterprise integration*. Wiley Computer Publishing, 2002.

IBM, “IT architecture: Building your IT city plan”, 1999.
http://www-1.ibm.com/service/its/us/source/cwp-itcityplan.pdf

Nagata, M., “A proposal for writing and reviewing papers on information systems,” *The Special Interest Group on Information Systems Notes of IPSJ*, Vol.2001, No.062, pp.25-30, 2001. (in Japanese)

Namba, Y., and Iijima, J. “Evaluating enterprise resource planning implementation to enterprise information system through city planning approach,” *Proceedings of the Japan Society for Management Information, Spring 2004*, pp.98-101, 2004a. (in Japanese)

Putman, J., *Architecting with RM-ODP*, Prentice Hall, 2001.

Chapter 2

IT Architecture Framework and IT City Planning

2.1 Introduction

Firms should have a structure to respond to the change of speed in the business environment, as mentioned in Chapter 1. Such a structure is required for both the business itself and the information systems that sustain the business. It follows then, that firms' information systems need architecture at the firm level as well as at each information systems level.

This chapter begins by summarizing and reviewing prior studies on IT architecture framework. IT architecture framework, in this dissertation, comprises a framework to describe the architecture of firms' information systems comprehensively. Next, it discusses the concept and meaning of "city planning." The city planning concept is included in the category of IT architecture framework, but it is unique in that it classifies a difference of granularity between the firm level and each information systems level. Last it will present some criticism on this topic.

2.2 IT Architecture Frameworks

The growing size of EIS and the increasing number of subsystems allow EIS to be a more complex and complicated structure. Moreover, closed relationships between applications required by business make EIS hard to maintain and operate. These situations have fostered in firms the necessity to understand EIS comprehensively from the perspective of IT architecture framework beyond each information system. The following section illustrates a typical IT architecture framework.

2.2.1 Zachman Framework

Zachman (1987) presents an Information Systems Architecture (ISA) framework in which he tries (a) to grasp complex and complicated firms' information systems as a logical structure (an architecture) and (b) to build its framework. He states, "With increasing size and complexity of the implementation of information systems, it is necessary to use some logical construct (or architecture) for defining and controlling the interface and the integration of all of the components of the system." This ISA framework focuses on the development of large-scale systems without providing a comprehensive view of EIS. This idea, however, was extended in 1992 (Sowa & Zachman, 1992) to cover the entirety of a firm's activities.

The enhanced Zachman framework is organized as 36 cells arranged in a six-by-six matrix—six-layered model types (perspectives) and six elements of each model (aspects). A perspective is the result of an ordered logical method used to break an issue or topic into unique viewpoints or frames of reference (O'Rourke et. al., 2003, pp.5-16). Every upper layer is a meta-framework of a lower layer. Figure 2.1 shows the Zachman framework (ZIFA, 2003), and the relationship between perspective and aspect.

A s p e c t s

P e r s p e c t i v e	*What* Data	*How* Function	*Where* Network	*Who* People	*When* Time	*Why* Motivation
Scope (conctextual) *Planner*						
Business Model (conceptual) *Owner*						
System Model (logical) *Designer*						
Technology Model (physical) *Builder*						
Detailed Representation (out-of-context) *Sub-Contructor*						
Functioning Enterprise						

Figure 2.1 The Zachman Framework for Enterprise Architecture

The six perspectives are as follows:

- The Scope (Contextual) Viewpoint - aimed at the planner
- The Business Model (Conceptual) Viewpoint - aimed at the owner
- The System (Logical) Viewpoint - aimed at the designer
- The Technology (Physical) Viewpoint - aimed at the builder
- The Detailed Representations (Out-of-Context) Viewpoint - aimed at the subcontractor
- The Functioning Enterprise Viewpoint

The six aspects and the interrogatives to which they correspond are as follows:

- The Data aspect - What?
- The Function aspect - How?
- The Network aspect - Where?

- The People aspect - Who?
- The Time aspect - When?
- The Motivation aspect - Why?

The Zachman framework was the first structured IT architecture framework to grasp information systems from an integrated architecture viewpoint. According to IBM Business Consulting Service (2004, pp.26-29), “Zachman thought that it was important to fill in the gap between business strategy and information system (IT architecture) to maintain consistency between them. As a result of his study, he presented this framework to enlarge reliability of systems development and maintenance and to reduce overlapping cost.”

2.2.2 Reference Model of Open Distributed Processing (RM-ODP)

The International Standards Organization (ISO) and the International Telecommunication Union (ITU) defined a reference model—the Reference Model of Open Distributed Processing—after working on a joint standardization effort termed the Open Distributed Processing (ODP). RM-ODP is standard for modeling object-based distributed processing architectures; it separates concerns and simplifies the specification of the heterogeneous, open-distributed processing system (Putman, 2001). Putman states, “RM-ODP provides a generic decomposition of a system into five viewpoints…. The viewpoints each have different interests of the system at hand, different requirements from that system, different evaluation criteria and therefore contain different information and have different specifications for the system.” Since these viewpoints are not layered, they are independent of each other.

Five generic and complementary viewpoints are as follows:

- The *enterprise viewpoint* focuses on the purpose, scope, and policies for the system. It describes the business requirements and how to meet them.
- The *information viewpoint* focuses on the semantics of the information and the information processing performed. It describes the information managed by the system and the structure and content type of the supporting data.
- The *computational viewpoint* enables distribution through functional decomposition on the system into objects which interact at interfaces. It describes the functionality provided by the system and its functional decomposition.
- The *engineering viewpoint* focuses on the mechanisms and functions required to support distributed interactions between objects in the system. It describes the

distribution of processing performed by the system to manage the information and provide the functionality.

- The *technology viewpoint* focuses on the choice of technology of the system. It describes the technologies chosen to provide the processing, functionality, and presentation of information.

The enterprise through computational viewpoints are independent of distribution and the enterprise through engineering viewpoints are independent of implementation, as shown in Figure 2.2.

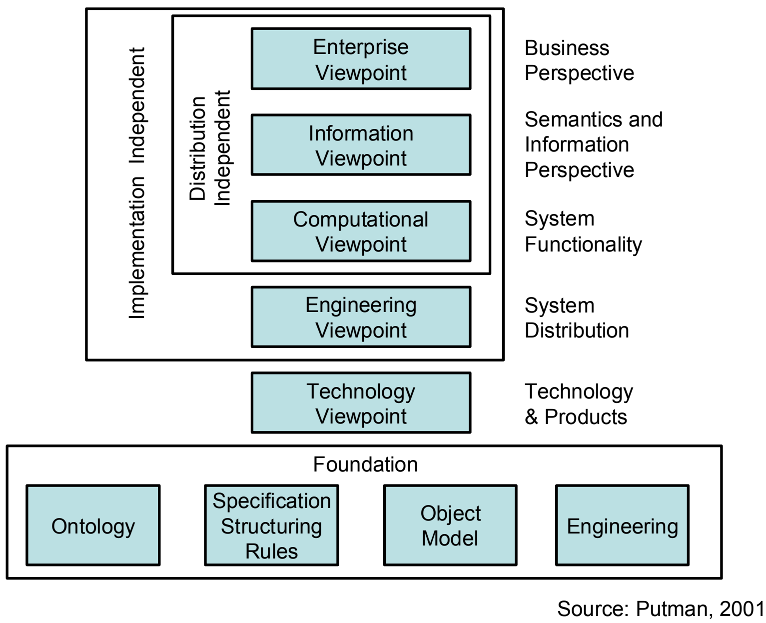


Figure 2.2 Viewpoint Foundation

As each viewpoint is an independent existence, viewpoints can apply singly or collectively. For instance (Putman, 2001):

- The information and computational viewpoints focus on software architecture.
- The enterprise, information, and computational viewpoints focus on system components or a related RM-ODP standard specification.
- All viewpoints focus on Domain-Specific Software Architecture and instantiations of that, or complete specification for a system.

RM-ODP provides a clear concept of the viewpoints that describe architecture, and it focuses on a reference model that "provides a durable framework to structure decision-making about an architecture…and guides the choices made in an architecture which, in turn, guide the choices made in an implementation" (Putman, 2001). This model is characterized by clarifying a viewpoint and focusing a reference model to describe an architecture.

2.2.3 TOGAF and Other frameworks

The Open Group (2002b) has developed The Open Group Architecture Framework (TOGAF). It is an architectural framework that enables the user to design, evaluate, and build the right architecture for an organization. The key to TOGAF is the TOGAF Architecture Development Method (ADM)—a reliable, proven method for developing an IT architecture that meets the needs of a firm's business. TOGAF includes four types of architecture as subsets of an overall enterprise architecture:

- Business architecture,
- Data/information architecture,
- Application (systems) architecture, and
- Information Technology (IT) architecture.

The Open Group provides two available versions of TOGAF:

- TOGAF Version 7 ("Technical Edition"), published in December 2001 (The Open Group, 2002b).
- TOGAF Version 8 ("Enterprise Edition"), first published in December 2002 and republished in updated form as TOGAF Version 8.1 in December 2003 (The Open Group, 2003).

TOGAF Version 8 uses the same underlying architecture development method that evolved, with particular focus on technical architectures, in the transitions up to and including TOGAF Version 7. TOGAF Version 8 applies this architecture development method to all domains of an overall enterprise architecture; including business, data, and application architecture, as well as technical architecture. The framework originated in large part from the Zachman framework, and The Open Group describes the relationship between the Zachman framework and TOGAF as "mapping the TOGAF ADM to the Zachman framework" (The Open Group, 2002a).

2.2.4 Enterprise Architecture (EA): Federal Enterprise Architecture Framework

The Federal CIO Council (1999) has proposed a federal enterprise architecture framework (FEAF) intended to help architects, the agency head, and the Chief Information Officer (CIO) develop, maintain, and facilitate the implementation of top-level enterprise architecture for federal enterprise systems. The FEAF provides an organized structure and a collection of common terms by which federal divisions can integrate their respective architectures into the federal enterprise architecture.

The purposes of the FEAF are (The CIO Council, 1999) as follows:

- Organize federal information on a federal-wide scale;
- Promote information sharing among federal organizations;
- Help federal organizations develop their architectures;
- Help federal organizations develop their IT investment processes quickly; and
- Serve customer needs better, faster, and more cost effectively.

The goal of the CIO Council is to develop a framework to prepare an enterprise architecture description. The framework consists of various approaches, models, and definitions for communicating the overall organization and relationships of the architecture components required for developing and maintaining a federal enterprise architecture. The framework must (a) be flexible to allow for new activities, (b) focus on common federal enterprise architecture activities, (c) address the realities of the federal workplace, and (d) provide a measure of immediate success. The FEAF is an organizing mechanism for managing the development and maintenance of architecture descriptions. The FEAF also provides a structure for organizing federal resources and describing and managing federal enterprise architecture activities.

Figure 2.3 shows the concept of federal enterprise architecture framework. As shown in the figure, the FEAF defines architecture, the transition process, and related concepts as follows:

- *Architecture Drivers* represent two types of external stimuli or change agents for the enterprise architecture: business and design.
- *Strategic Direction* guides the development of the target architecture and consists of a vision, principles, and goals and objectives.
- *Current Architecture* defines the “as is” enterprise architecture and consists of two parts: current business and design architectures. This is a representation of current capabilities and technologies, and is expanded as additional segments are defined.
- *Target Architecture* defines the “to-be-built” enterprise architecture and consists of two parts: target business and design architectures. This represents the future capabilities and technologies resulting from design enhancements to support changing business needs.

- *Transitional Processes* support the migration from the current to the target architecture. Critical transition processes for the federal enterprise include capital IT investment planning, migration planning, configuration management, and engineering change control.

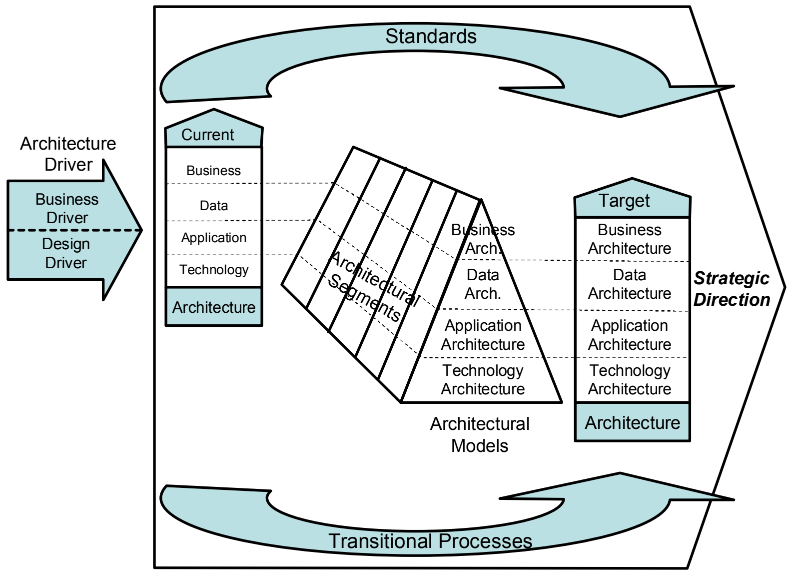


Figure 2.3 Structure of the FEAF Components

As shown in Figure 2.3, the FEAF partitions a given architecture into business, data, applications, and technology architectures. Thus, the FEAF defines (a) the current, (b) the target architecture, and (c) the transitional processes in the domain of defined enterprise, and focuses on promoting interoperability within the realm of enterprise. In 1991, the CIO council issued "A Practical Guide to Federal Enterprise Architecture Version 1.0." This guide provides a more concrete definition of the concepts and migration plan.

2.2.5 Enterprise Architecture (EA): METI

In Japan, the IT Associate Council sponsored by the Ministry of Economy, Trade and Industry (METI) published an interim report on enterprise architecture (EA) in November, 2002. This report was followed by a design guideline for EA (IT Associate Council, 2003) in December, 2003, based on the federal enterprise architecture. Prior to designing EA, Murakami (2004), a METI officer in charge of EA framework, conducted a hearing on concerns of IT and management from 18 management members in the Information Technologies and Business Strategy Council. As the designated key words were "customer-oriented management," "optimization," and "information sharing and personnel affairs," METI started to design a framework for complete IT management. Thus, the Japanese EA model is based on an optimization perspective, while the U.S. model is based on interoperability. According to EA guidelines, "EA is a design and management guideline in which operation processes and systems are modeled within an entire organization using a common language and a unified method. It aims to reform them to the direction of a customer orientation from optimization point of view."

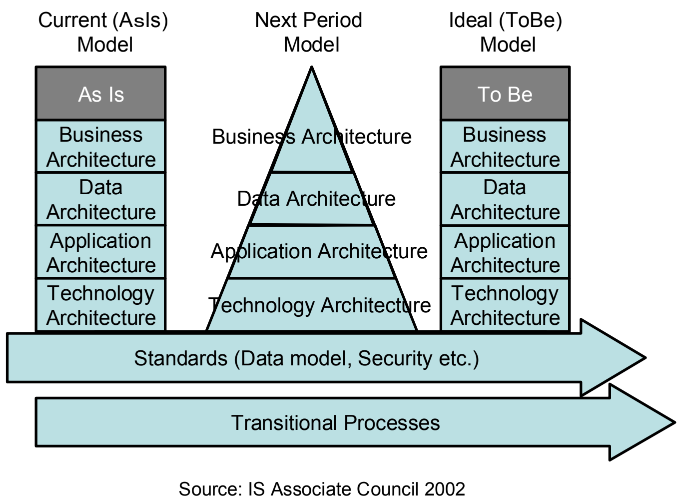


Figure 2.4 METI Model Framework

EA is composed of an architecture, a governance, and a migration plan. The architecture is described in four layers: a business architecture, an application architecture, a data architecture, and a technical architecture (see Figure 2.4). EA sets a next period model as the actual target in order to clarify the time-line relation from a current state to an ideal goal. Firms only execute the transitional process to realize the next period model.

A significant aspect of METI's EA framework is that it is the first authentic architecture centric framework at the enterprise level for the Japanese IT industry—an industry that has had a poor climate from the architectural point of view. This framework demonstrates that firms have to plan this kind of architectural framework and implement the systems along with the framework. Since vendors who bid to the government or local government for procurement of information systems are required to adopt the framework, the EA is responsible for generalizing and popularizing the concept.

2.3 City Planning Concept

2.3.1 City Planning Metaphor

Meanwhile, an approach toward grasping not just individual architecture, but an aggregation of applications, and a method of dealing with whole enterprise systems comprehensively, was originated by Gartner Group. In a report arguing that "the failure to distinguish between architectural 'blueprint' level issues and macrocosmic 'city planning' issues is the primary reason that IT architecture has not worked in the past," the Gartner Group (Schulte, 1997) advocated this concept in order to present a complete picture of EIS. As this approach depends on a metaphor for city planning as is practiced in civil engineering, it is called IT city planning. According to the Garter Group report, "The design of a building or an application system is an architectural issue; one set of blueprints can describe the structure in detail because there is one developer." This concept has become known as the city planning concept of IT architecture, in which an analogy is drawn between EIS and cities. The more a business environment becomes complex, the more important it is to integrate various independent information systems that organize EIS and to exploit them. The Gartner Group's approach aims to integrate and utilize the disparate and diversified individual applications that make up the entirety of enterprise systems.

Other groups (e.g., IBM, 1999) have proposed similar city planning concepts using the same analogies. Iwata (2002) reported on an architectural model of EIS that are used as a prerequisite for city planning. In this report, he proposed an information systems architecture that consists of four models: an application portfolio model, a processing pattern model, an information technologies reference model, and a platform type model. He also provided a reference model for each model in order that the audience could easily understand what it was.

2.3.2 Analogy Between Information Systems and Building Construction

Prior to stating the city planning concept, an analogy between building construction and information systems is illustrated. The term “architecture” comes from civil and architectural engineering, and denotes a philosophy or style of building. A more thorough discussion of this topic will follow in Chapter 3.

There are many articles, reports, and books that use the origin of the term “architecture” to offer an analogy between software architecture and building design. Sewell and Sewell (2001, p.2) state, “There is a perfect, profound analogy between software and building construction…The analogy between software and building construction is far more than an academic exercise—it is an indispensable tool, a template.” They insist on the effectiveness of using this analogy to inform non-technical individuals about the software construction process. They continue, “With the analogy in mind, it swiftly becomes clear that an entire branch of knowledge has been missing from software construction. This branch of knowledge comprises one of the oldest professions in human history.” They also restate the effectiveness of utilizing ideas from building architecture in the world of software, though their major concerns were with the architect rather than the architecture itself.

They also cite Vitruvious’ work that states that all architecture is comprised of three elements known as the Vitruvian triad (Figure 2.5). “*Utilitas* represented the need for a structure—the function of the structure.” “*Venustas* is the design. The design is created to meet the functional need of the client and represents the organization and artistic arrangement of the systems and materials.” “*Firmitas* represents the means, materials, and logistics of construction.” “The Vitruvian triad translates perfectly to software construction and is an elegant cognitive map of the essential roles and responsibilities engaged in the creation of a structure.”

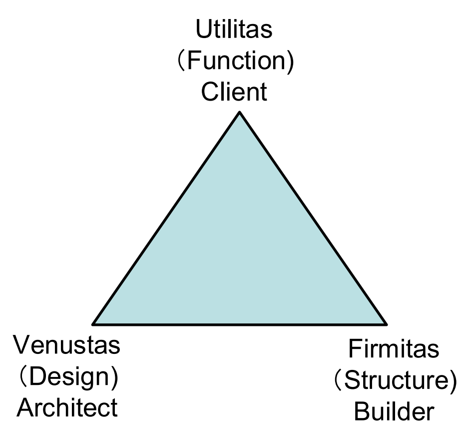


Figure 2.5 Vitruvian Triad: Three Elements of Architecture

Alexander (1979, pp. 19-40), a building architect, mentions “the quality without a name.” He says, “There is a central quality which is the root criterion of life and spirit in a man, a town, a building, or a wilderness. This quality is objective and precise, but it cannot be named.” According to Sewell and Sewell, “It lies at the heart of the questions ‘What is good architecture?’ and ‘What is good design?” They continue, “A tenet of Mr. Alexander’s philosophy is based on patterns: Our lives are patterns of events, done over and over again; a town evolves according to the human patterns of its inhabitants; a building is a collection of design patterns that either bring the occupants to life or thwart them.” Though he did not intend to write this literary work for a software architect, builder, and engineer, Alexander’s architecture pattern influences them.

2.3.3 Meaning of Analogy

As Meta Group (Burke, 2002) state, “City planning is an easily understood metaphor that architects can employ to communicate more effectively the nature and value of architecture by relating unseen enterprise architecture to real-world concepts that are well understood.” Sewell and Sewell (2001, p.2) suggested that this metaphor made clear the entire branch of knowledge, as mentioned above. In his article “Scientific philosophy for discovery,” Noe (2000) mentions the functionality of this metaphor in scientific discovery. According to Noe, one aspect we cannot neglect for discovery is the metaphorical usage of the concept and terminology when a new theory is submitted. When a scientist faces an irregular case, the most effective way is to leverage a traditional concept or terminology as a metaphor to describe the case.

A metaphor can be used in various ways. Surely, leveraging a metaphor is helpful not only in communication between architects and their clients but also to find some missing functions or knowledge in the new field. When seeking a new framework, metaphor and analogy have an important role.

2.4 Critical Analysis on City Planning Metaphor

2.4.1 Criticism from Brooks

Brooks (1995, p.201) states, “The building metaphor has outlived its usefulness. It is time to change again. If, as I believe, the conceptual structures we construct today are too complicated to be accurately specified in advance, and too complex to be built faultlessly, then we must take a radically different approach." He continues, "Let us turn to nature and study complexity in living things, instead of just the dead works of man…The brain alone is intricate beyond mapping, powerful beyond imitation, rich in diversity, self-protecting, and self-renewing. The secret is that it is grown, not built."

After all, the biological analogy may be effective in describing each information system, insofar as it involves discussing a complex and complicated conceptual structure. It requires, however, a different granularity or degree of abstraction to comprehend EIS structure within the framework of city planning and apply this methodology for EIS as a meta-structure of each information system. Moreover, it is also an issue of what is to be described in the EIS context. For this purpose, an EIS city planning approach that uses the concept and methodology of physical city planning would be an effective measure for CIOs and EIS architects who are required to envisage EIS architecture and execute it while ensuring interoperability and consistency among EIS.

2.4.2 Criticism from Ross

In an article discussing her four stages of a maturity model of IT architecture, Ross (2003) argues that, although "the city plan concept has given birth to a breed of IT architecture," it often provides "only the technologist's perspective of the relationship between IT and business processes…but it does not highlight the few IT capabilities critical to enabling the firm's strategic objectives." She continues, "These capabilities are the objectives of the IT architecture, specifying what the architecture enables the business to do." Ross's criticism of IT city planning models focuses on the idea that they have not fully exploited all IT capabilities. "Accordingly, the city plan metaphor has failed to capture the strategic potentials of enterprise IT architecture." Rightly, Ross calls attention to "the objectives of the IT architecture, specifying what the architecture enables the business to do."

In addition, she also elucidates the competency of enterprise IT architecture and the associated challenges. The logical sequence for developing the architecture is assumed to be as follows:

- Define the firm's strategic objectives.
- Define key IT capabilities for enabling those objectives.

- Define the policies and technical choices for developing the IT capabilities.

The major difficulty in the first step is obtaining the firm's strategic objectives. In the third step this difficulty is the trade-off between policies and technical choices for developing IT capabilities.

2.4.3 Direction of the Dissertation

City planning concepts focus on giving stakeholders a concrete image of EIS architecture with a metaphor, because the architecture is highly conceptual and abstract with no immediate visible attributes. The Meta Group states that this metaphor is helpful in communication between architects and their clients. Metaphor, however, is effective and available for those who find it hard to understand intangible or abstract things such as architecture. This is because metaphors serve to exploit the similarity of structure between abstract and concrete things. This does not always ensure the relationship between objects compared metaphorically is sound, but it sometimes includes or suggests profound insight. Excess usage of metaphor, however, can serve to mislead or confuse the target audience.

The criticism from Brooks is based on the presumed complexity of the software product. Because the granularity of EIS is much courser than that of software, the complexity of EIS may be seen as simpler with respect to abstraction and size. Ross's criticism is intended for conventional city planning, and the author fully agrees with her outlook on IT architecture's capability in that context. Likewise, the issues raised by Sewell and Sewell are significant. Thus, an IT architecture framework that can apply to an entire firm's information systems comprehensively must be presented. It includes the role of enabling IT capability, as well as that of the business driver. Furthermore, it must be understood easily by the non-IT person. The author tries to compose an IT architecture framework at the firm level in the following chapter.

Consequently, the IT architecture framework should incorporate the following:

- It must be a comprehensive framework and cover the domain of a firm;
- It should utilize the IT capability and not the mere technical standard or buy list;
- It should be easily understandable for the non-IT person, especially management members; and
- It should be applicable for rebuilding EIS and effective for use.

2.5 Summary of the Chapter

This chapter introduced the preceding studies on IT architecture framework, including the Zachman framework, RM-ODP, TOGAF, FEAF, and METI's enterprise architecture. It also showed the city planning approach that is characterized by exploiting (a) the relationship between information systems and building construction and (b) the comparison of EIS to a city.

Following the introduction of IT architecture framework, the author offered an analogy between software products and building constructs. First, Sewell and Sewell's work to describe the effectiveness of this metaphor was referred. Further, Alexander's philosophy based on pattern was brought up. Some criticism on the concept or the methodologies on city planning from Brooks and Ross were also introduced. Brooks questions the effectiveness of architectural metaphor because he thinks the conceptual structures are too complex and complicated to be specified accurately or to be built faultlessly. He suggests using not the building metaphor, but living things to illustrate the conceptual structure. Ross objects that the city planning IT architecture framework is treated as a mere technical standard and has failed to capture the potentials of enterprise IT capability. Finally, the theme of the dissertation with reference to prior related studies in this field and their strengths and weaknesses was discussed.

References

Alexander, C., *The timeless way of building,* Oxford University Press, New York, 1979.

Brooks, F.P., *The mythical man-month: Essays on software engineering, Anniversary edition*, Addison-Wesley, 1995.

Burke, B. "Enterprise architecture or city planning?" 2003. http://www.metagroup.com/

IBM, "IT architecture: Building your IT city plan", 1999. http://www-1.ibm.com/service/its/us/source/cwp-itcityplan.pdf

IBM Business Consulting Service, *Enterprise architecture*, Nikkei BP, 2004. (in Japanese)

IT Associate Council, "Optimization plan for business & systems ver. 1.1", 2003. (in Japanese)
http://www.meti.go.jp/policy/it_policy/itasociate/it.associate.htm

Iwata, H., "Introduction to information systems architecture," *Tutorial of the Information Processing Society of Japan, Spring 2002*, pp.1-10, 2002. (in Japanese)

Murakami, K. "Enterprise architecture," in "How should we grasp EA?" Symposium, 2004. (in Japanese)

Noe, K., "Scientific philosophy for 'discovery,'" in Morishita, S. and Miyano, S. (Ed.) *Discovery science and data mining*, Kyoritsu Shuppan, 2000.

O'Rourke, C., Fishman, N., and Selkow, W. *Enterprise architecture using the Zachman framework*, Course Technology, Boston, 2003.

Putman, J., *Architecting with RM-ODP*, Prentice Hall, 2001.

Ross, J. W., "Creating a strategic IT architecture competency: Learning in stage", *CISR Working Paper* No.334, April 2003.

Schulte, R., "Architecture and planning for modern application styles," *Strategic Analysis Report*, Gartner Group, April 1997.

Sewell, M. T., and Sewell L.M. *The software architect's profession: An introduction*, Prentice Hall PTR, 2001.

Sowa, J.F., and Zachman, J. F. "Extending and formalizing the framework for information systems architecture" *IBM Systems Journal*, 31, pp.590-616, 1992.

The Chief Information Officers Council, *Federal enterprise architecture framework*, September 1999.

The Open Group, "Mapping the TOGAF ADM to the Zachman framework," 2002a.
http://www.opengroup.org/architecture/togaf8-doc/arch/p4/zf/zf_mapping.htm

The Open Group, "The Open Group architectural framework version 7." 2002b.
http://www.opengroup.org/togaf/

The Open Group, "Welcome to TOGAF 8: Enterprise edition," 2003.
http://www.opengroup.org/architecture/togaf/

Zachman, J.A., "A framework for information systems architecture," I*BM Systems Journal*, Vol.26, No.3, pp.226-292, 1987.

ZIFA, 2003.
http://www.zifa.com/

Chapter 3

EIS City Planning Approach for Enterprise Information System

3.1 Introduction

As mentioned in Chapter 2, METI (IT Associate Council, 2003) published enterprise architecture (EA) based on the federal enterprise architecture. These activities triggered vendors or consulting firms to start offering their solutions to firms. These situations, in the background, have made firms increasingly interested in EA or in this kind of IT architecture framework. Since most large-scale EIS are composed of independent, heterogeneous systems developed at different times by independent teams, they often have a complicated structure of disparate architectures. Planning an architecture for such aggregate information systems thus requires a comprehensive approach different from that taken when planning an architecture for individual systems.

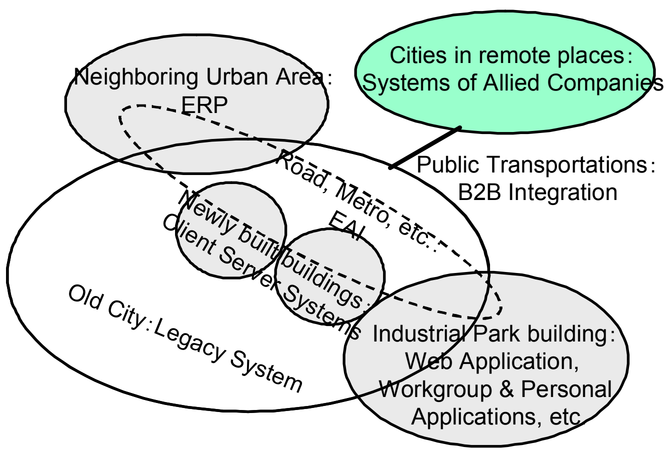


Figure 3.1 EIS as Metaphor for City Planning

Namba and Iijima (2003a, 2003b) proposed the Enterprise Information Infrastructure (EII) Meta-model, which offers an integrated information infrastructure framework for EIS. That paper illustrates EIS as a city planning metaphor. Figure 3.1 shows the relationship between enterprise systems and the metaphor for city planning. The entire enterprise system, in this figure, includes a legacy system, a client/server system, ERP, and Web applications; while the city includes an old and a new city area, a neighboring urban area, and an industrial park. Each system is connected by an application integration infrastructure (in the case of intra-enterprise connection) and a B2B integration (in the case of inter-enterprise connection), as each construct is connected by a road or a public transportation system.

As long as one adheres to merely a technologist's perspective or a metaphor, these kinds of IT architecture frameworks for enterprise or city planning will lose their usefulness over time. To combat this, Namba and Iijima proposed the EIS city planning approach (2003c, 2004b) and presented other relevant articles (2004a, 2005) based on the EII Meta-model. The framework in this dissertation differs from the city planning concept that Ross criticized, insofar as:

- It defines the conceptual model of EIS architecture based on the IEEE architecture model.
- It exploits the concept and the methodology of physical city planning that have been cultivated in the long history of civil engineering for the framework of EIS city planning.
- It introduces an "EIS scenario" that stipulates the migration plan.
- It provides three viewpoints to describe the EIS architectures, rather than discussing architecture in general. The purpose of classifications is to clarify the necessity of including all stakeholders' concerns and to describe the EIS architecture with views that correspond to each vantage point.

This chapter aims to describe an EIS architecture framework using the analogy between cities and EISs. The EIS city planning approach is comprised of both an EIS architecture and an EIS scenario. First, an EIS architecture with reference to the architectural description provided by IEEE Computer Society is illustrated. Then an EIS scenario that includes a migration plan is discussed that has a mechanism of program management that achieves an aggregation of each project included in the EIS. The term "program management" comes from a key concept in Project and Program Management (P2M). "Program," in this context, means an organic combination of projects that focuses on a specific mission, and it is distinguished from the term "multiple projects," in which the interdependency of each project is low or none. Finally, the applicability and effectiveness of the framework is discussed.

3.2 EIS City Planning Approach

3.2.1 Granularity

As mentioned in the city planning section in Chapter 2, the granularity of an object is different between designing architecture for each information system and for a whole

EIS. Just as several levels of planning exist for physical city planning, several levels of planning exist for EIS city planning. Figure 3.2 illustrates the relationship between physical city planning (Shirvani 1985, p.142) and EIS city planning. Physical city planning involves various scales or frames of reference, from an object level to a national level. Shivrani points out that policy-oriented attitudes dominate at courser levels, while product-oriented attitudes dominate at finer levels. The right side of Figure 3.2 gives the corresponding unit for EIS city planning. For the EIS, comprehensive and interoperable characteristics correspond with courser granularity, while analytical and specific characteristics correspond with finer granularity. Thus, consistency among EIS architectures should be considered. The boundary of the EIS is shown as the box of the internal systems in Figure 3.2 to classify the ins and outs of the enterprise system.

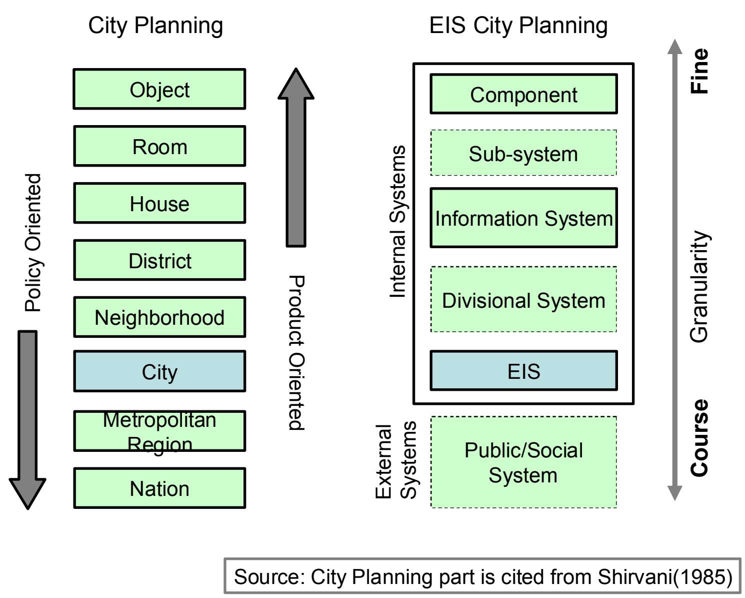


Figure 3.2 Granularity of Planning Levels

3.2.2 EIS Architecture

Before discussing EIS city planning, several key concepts must be defined. Both the federal enterprise architecture framework (The CIO Council, 2001) and the EA Guideline (IT Associate Council, 2003) define an "enterprise" as "an organization (or cross-organizational entity) supporting a defined business scope and mission." This definition, however, seems a bit ambiguous for the firm that adopts the business unit system, since the definition is applicable to the whole company or to a single business unit. The author instead defines enterprise as the highest level of business domain governed by unified policy. When the lower business domains are highly independent and the business relationship to the upper level is weak, the independent business unit can also be defined as, substantively, an enterprise. Enterprise includes not only business enterprises but also non-business enterprises, such as a governmental one or a municipal corporation.

The term "architecture" traditionally refers to a building construction, such as a building or a bridge, and it has been used in the fields of computer science and information systems to form an analogy between civil and systems engineering. When IBM's Systems 360 came onto the market in 1964, the term "architecture" was defined as "the definition of the external specification that is independent from implementation." Since then, various definitions have been proposed (i.e., Software Engineering Institute SEI, 2004; The Open Group, 2001), but "there is no standard, universally-accepted definition of the term, for software architecture is field in its infancy, although its roots run deep in software engineering" (SEI, 2004).

Sewell and Sewell (2001), for instance, note the similarity between building constructions and software (see Chapter 2). Namba and Iijima (2003c, 2004b) applied this similarity to devise an analogy between a city and an EIS structure. As a city is an aggregation of buildings and various constructs, an EIS is composed of independent information systems structures. In this sense, EIS architecture can be called a "meta-architecture."

For the purpose of drawing out the analogy, IEEE Std 1471-2000 (IEEE Computer Society, 2000) has been referred to, which illustrates an architectural description using a class diagram showing the relationship between stakeholders, concerns, viewpoints, and views (See Appendix). The author modified the IEEE diagram by arranging some elements in line with the nature of the EIS, changing the definition of terms, simplifying the architectural description, and describing the class diagram. This is shown in Figure 3.3.

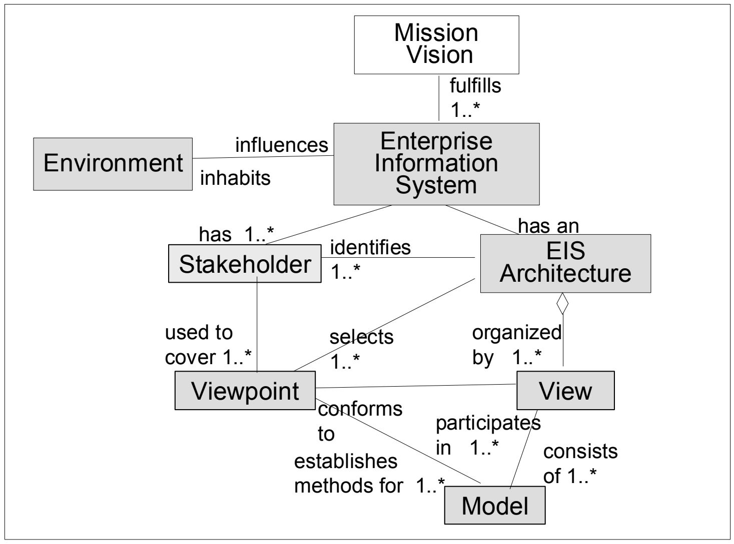


Figure 3.3 Conceptual Model of EIS Architecture

An EIS that fulfills the vision or mission of a firm takes into account both the EIS architecture and stakeholders. Planning an EIS architecture, in other words, requires identifying stakeholders, selecting one or more viewpoint(s), and organizing the plan by views. A view conforms to a viewpoint and may consist of one or more model(s). Each model is established with a method defined by the corresponding viewpoint. A viewpoint, in turn, establishes the associated modeling method and analysis techniques. The architect's role is to design the EIS comprehensively from viewpoints defined according to the concerns of stakeholders.

Generally, stakeholders in information systems include IS staff, end users, and management (Ewusi-Mensah, 1997). As RM-ODP (Putman, 2001) states, "Stakeholder is a term to represent any customer, user, owner, administrator, acquisition authority, or program manager." According to IEEE Std 1471-200, the stakeholders that architects should consider when they define an architectural concept are systems users, systems owners, systems developers, and systems maintainers. In the context of EIS, stakeholders include management members, CIOs, local management, EIS architects, and other persons who have an architectural concern in the outcome of the EIS. Insofar as an EIS must connect customers (business to customer electronic business (BtoC)) and/or aligned or partner companies (business to business electronic business (BtoB)), the definition of stakeholders can be extended to include persons or parties of the relevant information systems, such as customers, allied companies, and business partners. Generally, these stakeholders may not have any concern in designing EIS architecture. Usually the social environment, CSR, or competent authorities exert influence on an EIS, but these do not have the same level of concern in it. As a result, all of these factors are not stakeholders of an EIS, but rather make up its environment. Through the EIS, they have a direct or indirect effect on each stakeholder.

An EIS must meet the vision and/or mission of the enterprise, as shown in the top box of the class diagram in the Figure 3.3. When a firm plans an EIS, the architect must map out the vision of the EIS architecture, which becomes a blueprint for the future enterprise system. In most cases, it may be difficult to migrate directly from an existing EIS architecture (as-is EIS architecture) to this ideal future architecture (to-be EIS architecture) because of constraints such as human resources, technological capabilities, the current status of effected information systems, the social environment of a firm, or CSRs. Within these constraints, a firm is tasked with setting the actual or pro tempore target outcome for the EIS architecture (live-to-be EIS architecture). Since the nature of a live-to-be EIS architecture varies with time, the architect has to periodically review and redesign it, in accordance with actual performance, degree of environmental change, and the progress of technologies.

The FEAF defines terms with meanings similar to those given in the framework (see Chapter 2). The current architecture is defined as the "as is" enterprise architecture and consists of two parts: current business and design architectures. The target architecture is defined as the "to-be-built" enterprise architecture and consists of two parts: target business and design architectures. In the FEAF, the transitional processes support the migration from the current to the target architecture. The difference between the FEAF and the framework in this dissertation is that the author clearly separates the live-to-be architecture from the to-be architecture, although the FEAF does describe an in-between architecture in "A Practical Guide to Federal Enterprise Architecture" (The CIO Council, 2001). The EIS scenario for the framework corresponds with the "transitional processes of FEAF," which target the live-to-be instead of the to-be architecture.

3.2.3 EIS Scenario

"Development plan" and "zoning" are fundamental rules for concrete land utilization in realizing a physical city plan (Fukukawa, 1997, p.18). Since city planning is closely related to the law and/or the regulation systems of a country, the usage of terms differs, to some extent, among countries. In the U.S., for instance, a development plan is called a "comprehensive plan," a "regional plan," or a "master plan" (Hall, 1988; Catanese & Snyder, 1988). "Traditional zoning ordinances deal with this aspect of physical form by setting specifics of height, setbacks, and coverage. There are also more elaborate and involved regulations (Shirvani, 1985, p.11)." Zoning, however, can have various meanings, such as "incentive zoning" or "performance zoning." It too is closely related to the relevant laws and regulations of a country. In general, the particular meaning of a term varies from country to country. In this dissertation, the meaning follows Fukukawa's work (1997), which compares the systems of city planning in the U.S. and Japan.

A development plan is a long-range master plan that shows a target feature of a city's structural fabric and a directionality for urban policy. In this sense, a development plan is a measure intended to solve issues proactively. Zoning is a rule or regulation enacted to control land utilization, which classifies an area, delineates usage of land, or prohibits a particular land usage. Typically, zoning aims at harmonizing land utilization in accordance with the characteristics of an area. In this way, zoning may be characterized as a passive planning technique.

Applying the analogy to EIS, a development plan corresponds to a mid- or long-range plan focused on realizing a live-to-be EIS architecture, while zoning corresponds to a migration plan with daily governance intended to iteratively achieve a live-to-be EIS architecture. Zoning formulates the EIS service portfolio with the aim of creating an integrated information infrastructure and executes this portfolio in accordance with regulations or pre-existing standards, which are analogous to a land utilization program or a building code.

City planning includes two methods of execution: new development, which may develop, for instance, a neighborhood or shopping center on vacant parcels; and redevelopment, which develops land housing old buildings or old infrastructures. Redevelopment applies to, for instance, slum clearance and includes scrap-and-build, rehabilitation, and conservation approaches (Sakamoto, 2000, pp.151-152). The goal of rehabilitation is to repair a building or infrastructure, while maintaining an existing community; while conservation aims at inhibiting deterioration. Redevelopment methods are characterized by the utilization of present infrastructures and assets, which are converted iteratively. Just as an EIS city planning approach necessitates consideration of cost, time, and risk, firms have to select a suitable method within the given situation and promote the project appropriately.

An EIS scenario implies a program management technique that manages an aggregate of projects at the level of the whole firm rather than at the level of each project. It includes both a long-range and short-range perspective, while it envisages both the impact of the architecture of each information system and the likely lifecycle phases of each project, with an emphasis on future optimization. On this account, the combination of an iterative approach with a zoning-like idea and a small-sized big bang is possible. In the words of Ewusi-Mensah, "The most obvious advantage of using the phased-lifecycle approach is to help the project team realize what the deliverables for each stage are and to know if they have been satisfied. The iterative nature of systems development notwithstanding, the phased-lifecycle approach has been instrumental in helping to manage and control the development of large, complex systems successfully."

Parr and Shanks (2000) present three categories for ERP implementation. "Comprehensive" implements all functions to the multiple sites with full business process re-engineering (BPR), "vanilla" implements only core function to the small sites with the least BPR, and "midway" is in-between. Thus, ERP implementation varies drastically, depending on the BPR at the respective firm levels. It is a very severe task that requires sufficient time and money as well as systems implementation. These points are also major factors for planning an EIS scenario.

When writing an EIS scenario, it is necessary to consider a firm's relation to new technologies. Rogers (1995, pp. 261-266) outlines five categories for technology adoption on the basis of innovativeness (see Figure 3.4). He gives each category a single word explanation: "venturesome," "respect," "deliberate," "skeptical," and "traditional" for innovators, early adapters, early majority, late majority, and laggards, respectively. When selecting technologies, firms should consider which category they occupy. When a firm needs new technology for a competitive edge, the firm should be willing to take on additional risk. When the focus is on stability or reliability, the firm should place itself in a more mature category. The placement of a firm in a category should result from its business model.

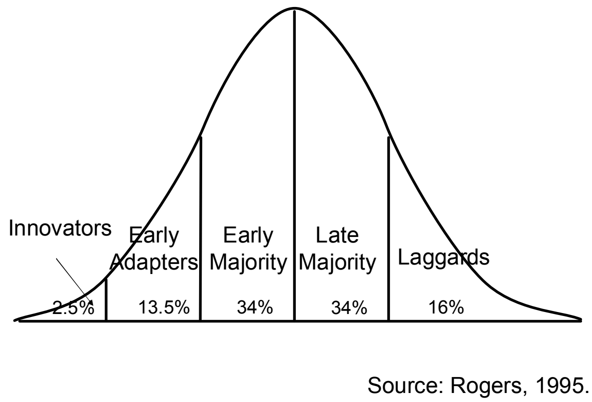


Figure 3.4 Adopter Categorization on the Basis of Innovation

Furthermore, an EIS scenario includes a migration plan, which delineates an EIS strategy based on the given enterprise's business strategy. The target migration is a live-to-be EIS architecture that maintains consistency among all information systems in the enterprise. An EIS scenario also takes into account program management, which includes EIS activities such as maintenance, systems monitoring, and operation tasks. As a framework, an EIS scenario includes processes, activities, and tasks of software products. It has to cover the entire lifecycle of the model, from the design phase to the disposal phase.

3.2.4 EIS City Planning Approach

The EIS city planning approach provides a methodology for designing an EIS architecture as a meta-architecture, for planning an EIS scenario, and for executing the migration required to realize the scenario. Live-to-be architecture is designed taking into consideration the gap between current (as-is) architecture and future (to-be) architecture, as well as actual constraints surrounding the EIS. The EIS scenario is a path to realizing live-to-be architecture that uses the methodology of physical city planning. Figure 3.5 shows the relationship between the firm's business model, the EIS architecture, and the EIS scenario.

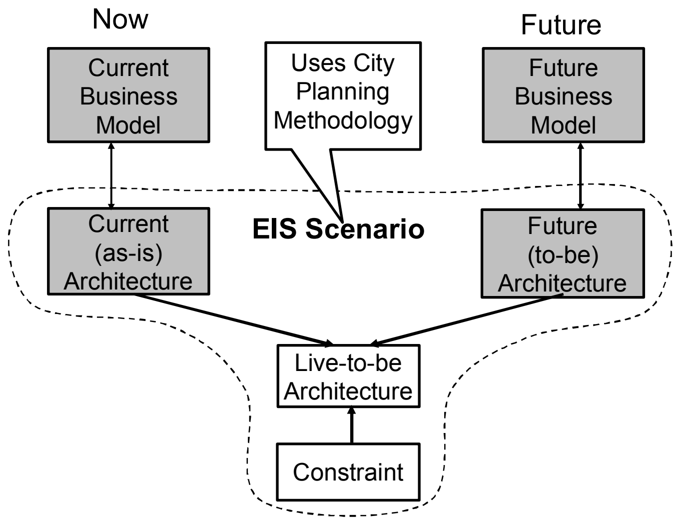


Figure 3.5   Framework of EIS City Planning Approach

3.3 Three Viewpoints of EIS Architecture

3.3.1 Selecting Viewpoints

As shown in Figure 3.3, architecture selects viewpoints. Since viewpoints determine associated modeling methods and analysis techniques, it is necessary to select a proper viewpoint for describing a projected EIS architecture. Every architectural viewpoint is a description intended to capture aspects of the planned objects (e.g., the drawing of a plane view, side view). In an EIS architecture, a viewpoint describes the structure of EIS and includes such modeling techniques as an entity relationship diagram (ERD) or a class diagram. Accordingly, viewpoints focus on each aspect of the system and are orthogonal with each other. A view, therefore, is an actual figure based on a viewpoint. Consequently, an architecture is described as an integration of views.

In physical city planning, there are layers. The infrastructure layer includes roads, water supply, sewage, communication lines, railroads, etc. The architectural layer includes buildings, houses, shopping malls, and other facilities. Surrounding or encompassing these two layers is a third, the social environment. Thus, city planning has a layered structure. Similarly, EIS architecture has a layered structure that corresponds to the structure viewpoint.

A city plan serves to mediate between the concerns of an individual or a particular area (the part) and the public (the whole). Specifically, a city must solve such problems as developments that do not profit the whole or not-in-my-back-yard (NIMBY) facilities, such as a garbage incinerator or a highway in a residential area. The conflict over these types of developments is that between the part and the whole.

Most cities are penetrated by national or interstate highways and must have a close relationship with neighboring cities and with the prefecture or state government. Thus, city planning takes place within a city, but must maintain a relationship with structures outside of the city. Similarly, EIS city planning involves those inside the enterprise, but, at the same time, it must include those outside of the enterprise: customers, vendors, and/or partner companies. This is accomplished through private or public networks and the Internet.

The author portrays EIS architecture from three viewpoints: “structure,” “part and whole,” and “ins and outs” (Figure 3.6), which come from the analogies of physical city planning mentioned above. These viewpoints are based on the concerns of stakeholders.

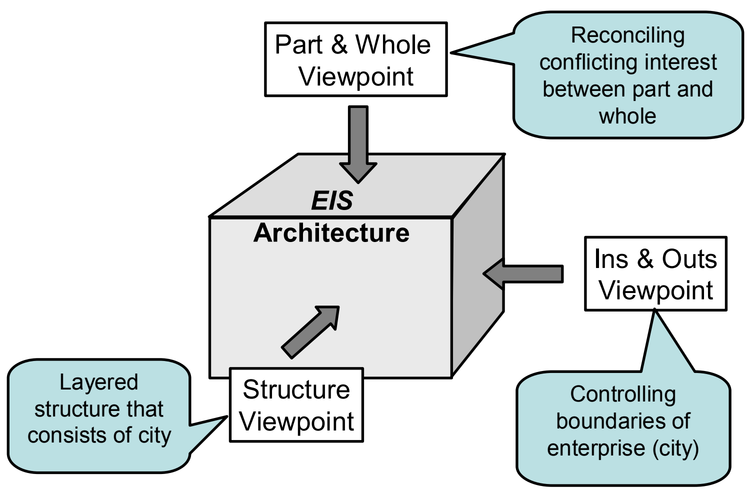


Figure 3.6 Three Viewpoints of EIS Architecture

3.3.2 The Structure Viewpoint

A physical city plan encompasses two domain types. The civil engineering domain focuses on developing a social infrastructure, which includes roads, water supply, sewage, communication lines, railroads, etc. The architectural engineering domain aims at (re)developing buildings, houses, shopping malls, and other facilities. The social environment provides the context for these two domains. Together they comprise a layered structure. Analogously, EIS architecture has the structure viewpoint shown in Figure 3.7. The top layer is the business layer, which is followed by the information systems services layer, and then the integrated infrastructure layer. These layers have a relationship such that the upper layer drives the lower layer and the lower layer enables the upper layer.

Layering aims to minimize the influence of changes in each layer to the other layer(s) by making each layer independent. Thus, once a layered structure has been achieved, technologies appropriate for information infrastructure can be selected and deployed without influencing the other layers, especially the information systems service layer. Additionally, the structure for each application can be implemented with a common information infrastructure.

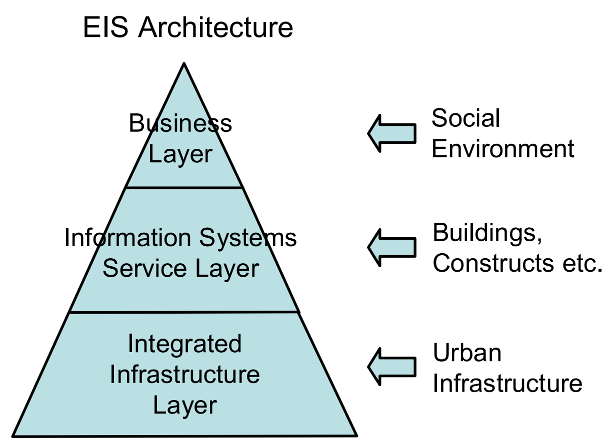


Figure 3.7 Structure Viewpoint

Since the business model of a firm is generally complex in nature, it is often difficult to describe it just as a business layer. The viewpoint of the stakeholders, however, can be rendered with modeling techniques, such as conceptual data modeling or business process modeling. The information systems service layer is an information systems services portfolio comprised of an aggregation of applications that consist of data and process specifications and/or information systems services. Here, the term “service” means the same as “service-oriented architecture.” The integrated infrastructure layer includes shared service and maintenance, monitoring and systems operations on the systems platform, middleware, DBMS, common interfaces for both intra- and inter-enterprise connections, and gateways (Namba and Iijima, 2003a; 2003b). The stakeholder, then, is a person or party who has a strong interest or concern in designing and implementing the EIS architecture.

Namba and Iijima (2003a; 2003b) proposed the EII Meta-model. In the article, they state that the integrated information infrastructure (In3) can serve as a guide in the selection of applicable IT. The EII Meta-model consists of an integrated information infrastructure (In3) map, a service framework, and an IT scenario. The IT scenario corresponds to a subset of the EIS scenario in the EIS city planning approach. The In3 map is prepared based on the requirements of the integrated information infrastructure and consists of applied technologies, a short list for implementation and maintenance, and an operation structure. The requirements classify three functions for integration, information infrastructure and non-functional requirements (NFRs), and system maintenance. NFRs refer to the requirements of data accuracy, availability and scalability of the systems, security, performance, etc. The IT scenario should be utilized to avoid rigidification of the list.

Table 3.1 illustrates the integration function as a reference model to describe and classify the integration technologies (Namba and Iijima, 2005) in an In3 map. This table gives viewpoints for classification. Because the viewpoints are independent from each other, this taxonomy can be used in a combined form. The category refers to the logical relation or role-sharing of sender and receiver. Topology refers to the configuration of collaborating functions for implementation. Tier refers to the layer at which collaboration proceeds.

Table 3.1 Classification of Integration Technology for Information Systems

Category	Topology	Tier	Protocol	Others
・Request / Response	・Point to Point	・Presentation	・Proprietary	・Sync
・Messaging	・Hub and Spoke	・Business Logic	・Industry Standard	・Async
・Data Transfer		・Data		・Batch
・Publish & Subscribe	・Bus, Pipeline		・Common	・Tight Coupling
・Reconcile				・Loose Coupling
・Data Share				

The points to consider with respect to the structure viewpoint are:

- Each layer is defined independently from the others,
- The information systems service layer reflects the structure of the business layer,
- Each application or service can collaborate with the others through a common interface that is provided by the integrated information infrastructure layer.

3.3.3 The "Part and Whole" Viewpoint

According to Sakamoto (2000), a physical city plan serves to mediate between the concerns of an individual or a particular area (the part) and the public (the whole). Part and whole viewpoints in EIS architecture are related to the ability to accommodate an interest (such as governance) or the responsibility of sharing between a part and a whole of a firm. As in physical city planning, in EIS architecture, if a firm employs a business unit system under a decentralization policy, it serves to regulate the relationship between a business unit and the corporate whole.

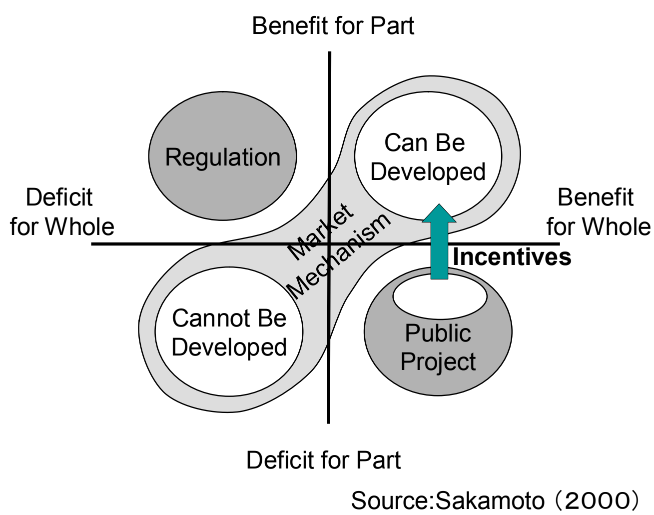


Figure 3.8 Part and Whole Viewpoint

Figure 3.8 shows the relationship of benefit between the part and the whole. Development in the right-upper region (first quadrant) and the left-lower region (third quadrant) do not generate major conflict because both the part and the whole, obeying market law, share a mutual interest. For the development in the left-upper region (second quadrant), however, it is often the case that the part benefits while the whole receives a deficit. In a case where the whole receives the deficit, some regulation or rule is required to compensate the whole. Developments in the fourth quadrant often lead to NIMBY-type conflicts, which require a plan to compensate the affected group.

When a NIMBY-type application or information infrastructure is implemented, a conflict of interest between the part and the whole arises. For instance, when a firm deploys a security policy and enhances its security infrastructure in line with the policy, this measure may give little benefit to a particular business unit from that unit's perspective. In such cases, firms may have a fund—corresponding to a city planning tax—from which the whole can draw to benefit the part. The stakeholders in this context are the people who have a responsibility to control the conflict of interest between part and whole, such as CIOs or management members.

The points to consider with respect to part and whole viewpoints are:

- Whole function is defined apart from part function,
- The function in the part and that in the whole are implemented separately,
- The organization or governance system ensures the whole authority.

3.3.4 The "Ins and Outs" Viewpoint

The ins and outs viewpoint is concerned with how to control boundaries between the inside and outside of the enterprise. Frequently, boundaries among legal, business operations, and information systems in a firm do not coincide. The business requirement for collaboration and alliance between firms accelerates the expansion of operational boundaries beyond the original boundaries of the enterprise as a legal entity. Simultaneously, the boundaries of an EIS also expand. Figure 3.9 shows the relationship between the boundaries of a firm's legal entity, information systems, and business activity. Outside of or along with these boundaries, competent authorities regulate the activity of firms, while the community and/or society require CSR. customers, business partners, customer companies, and vendors to be connected with some business activity through BtoC or BtoB. The boundaries of business activity and information systems cover a wider area than those of legal entity and do not coincide with each other because of their different natures and responsibilities. Even within firms, the necessity for collaboration beyond the boundaries of each individual application or information system is rapidly spreading.

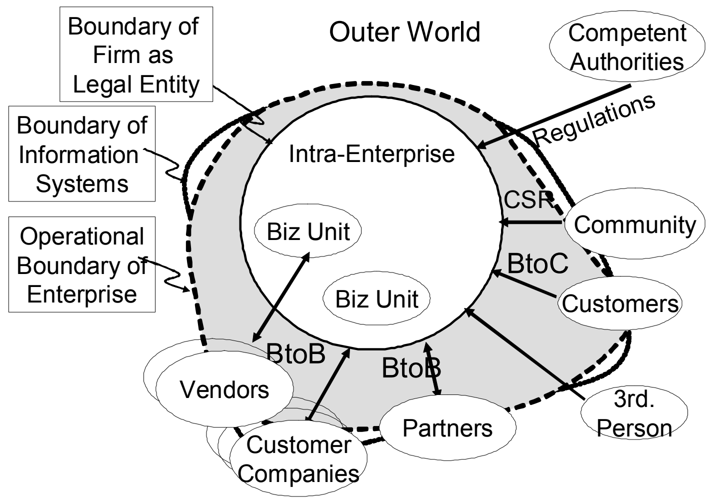


Figure 3.9 Ins and Outs Viewpoint

This viewpoint is deeply involved in the definition of an enterprise. Thus, the stakeholder of this viewpoint is a person or a party who has a concern regarding a boundary of a firm. For IS concerns, these are architects or persons who have to design or be responsible for the system's operation, as well as the maintenance required to connect information systems among those inside the firm and those outside it.

The points to consider with respect to the ins and outs viewpoint are:

- The boundary of the enterprise is defined,
- The systems operation reflects controllability based on the ins and out viewpoint,
- The balance between controllability and systems connection should be designed and implemented properly.

Each structural layer can have a different boundary as well. For instance, in an integrated information infrastructure, the boundary of an information infrastructure as a firm's asset may differ from the range of operational responsibility. The architect therefore has to design an intra- and inter-enterprise system structure and decide whether it should be implemented in the information systems service layer or in the shared information infrastructure layer. This decision should take into account role-sharing between the part and the whole, as well as the ins and outs viewpoint. Such factors are closely related to the organizational architecture of the firm.

Another important issue for the ins and outs viewpoint is security. Security is closely related to the issue of governance at the enterprise level. This is because security always concerns not just the average capability, but the weakest point of an enterprise. A firm, then, should always try to address security from a holistic perspective. From the ins and outs viewpoint, security relates to how to control a system outside the firm, or the enterprise involves a potential security hole that cannot be controlled directly. An enterprise generally cannot control persons or third parties outside the firm. Because events outside the firm are black boxes for the enterprise, it must be prepared for the worst-case scenario. In order to avoid such scenarios, a firm should create service level agreements or contracts in which outside parties assure security. In addition, a firm should separate its interface from intra-enterprise systems and/or deploy a dedicated interface system.

3.4 Discussion

3.4.1 Concerns for Stakeholders

During the period of legacy systems, the users of the systems were primarily employees and infrastructures were assets internal to the firm. The situation, however, has changed as firms now increasingly employ BtoC models and as customer use of firm systems has become a primary source of profit. Thus, both the user and the management domain of firm infrastructures are expanding to an area beyond the firm's boundaries. Moreover, the objectives of information systems now lie outside the firm's own systems, as is the case when firms employ BtoB type e-business or when interconnected systems become part of supply chain management.

The automatic teller machine (ATM) serves as a case in point: though the user is a customer, the bank itself owns the ATM and operates and manages the system. For an online bank, however, the terminal that a customer uses, as well as the network access from the terminal to the Internet, is within the direct purview of the customer. The only network that the firm can control is the access line to the firm's servers, but the firm still has to contact the customer's computing environment with a peer-to-peer base, often to extend support. Until technology reached this point, firms operated on a closed structure ATM model, controlling and owning the primary access points to their information systems.

In the case of a BtoC application such as an online banking interface, the customer becomes a stakeholder. From an EIS architecture point of view, however, it may be better to conduct planning without considering the customer as a part of the environment, since a customer does not relate to the system as a whole. In the event that a partner or allied company is included as a stakeholder, the relationship may become more critical since they are stakeholders whose systems interact with the firm's systems in BtoB transactions. In order to cope with these situations, top management or the CIO should bring these factors into EIS planning and implementation, considering them as imperative environmental concerns.

3.4.2 Viewpoints

When an ERP software with a modular type of production system is implemented in a company that conducts an integral-type business model, such as Toyota, there is a possibility that the firm's core competency could be lost. This is because only a change in the information systems service layer with COTS will have a closed relationship to the business layer. As a result, the new information system would become a disabler because the business layer would fail to portray the actual business of the firm.

Remaining attentive to attributes of the part and whole helps to clarify and compartmentalize the functions between them. This issue may be viewed differently by each layer of the structure. For instance, in a business unit with high managerial independency from headquarters, a business flow and/or physical distribution flow may nonetheless depend on a headquarters function, if the unit uses common systems or a common database in the information systems service layer. An independent information system function, in turn, may rely on a common networking function, data center, and monitoring and operation. Thus, the degree of actual independence varies depending on the layer being considered. When a business unit is defined as an enterprise, the relationship between the part and the whole is key.

While the ins and outs viewpoint may be seen as an extension of the parts and whole viewpoint, each address different concerns related to controllability. Within an enterprise, a whole can manage a part, while offering incentives or giving orders under the authority of headquarters when vigorous action is instituted. Such incentives, however, are not effective when the affected entity is outside the enterprise's immediate authority. For this reason, a public works project analogy is not valid when entities outside the enterprise are considered. The ins and outs perspective applies in those instances where the entities communicate with a common data interchange protocol or through a common interface. When the activities of a business unit are independent from the corporate headquarters, the business unit should be considered an enterprise and the headquarters an entity of outs.

Table 3.2 shows a summary of viewpoints to describe an EIS architecture. Each viewpoint articulates a stakeholder(s)' concerns and results in "view or output."

Table 3.2 Viewpoints and Views

Viewpoint	Stakeholder	Concern	View or Output
Structural	- EIS Architect - CIO	- Business driven & technology enabled - Layer's independency from other layers	- Conceptual Data Model - Independent layered structure - Service Portfolio - In3 Map (*)
Part & Whole	- Top Management - CIO - Local Management	- Distributed IT governance - Consensus Making	- IT Regulation - IT Standard - Incentives (Funding)
Ins & Outs	- Line of Business - Info. Infra. Designer	- Interface - Controllability of systems operation and maintenance	- Data Interchange Standard - Operation Design - Service Level Management - Security Policy

* Source: Namba and Iijima, 2003b

As each viewpoint has an independent existence, viewpoint can apply with combined form. Since a viewpoint in an EIS architecture is also defined as independent, a combination form of viewpoint (e.g., the integrated information infrastructure layer of structure viewpoint and the ins and out viewpoint) can be applied as a viewpoint.

Several methods are available to integrate information systems or applications, such as file transmission, a tight coupling with socket communication, a loose coupling through a message broker, or communication with XML, as illustrated in Table 3.1. These methods, however, are not applicable in every case, because there is a constraint inherent in the relationship between system controllability and the degree of coupling. Figure 3.10 shows the relationship between "degree of coupling" and "system controllability." The degree of coupling illustrates the strength of the collaboration, and system controllability shows the ease of control for targeted applications. The difference, however, does not vary continuously, because there is a gap between application integration and BtoB integration. In the case of application integration, it is a matter of affecting the inside of firms, while BtoB affects the outside. This gap corresponds to the gap of ins and outs.

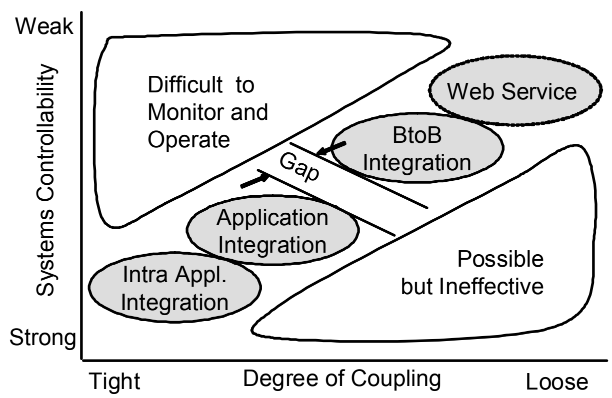


Figure 3.10 Collaboration of Information Systems

In the case of integration within an application that requires the tightest coupling, a structure of monitoring and an operation with tight communication is required. Conversely, in the case of tight coupling without controllability, it is difficult to maintain stable monitoring and operation. Loose coupling may be a useful solution to address controllability issues in this area.

3.4.3 EIS Scenario

It is difficult for most enterprises to replace their mission critical systems through a big bang, total reconfiguration approach, except in cases where the goal is to build new systems in an emerging business domain. This is partly because firms' information systems have become so big and complicated that the expense and the time of rebuilding are prohibitive. This is mainly because firms have no other choice than to utilize existing systems (sometimes legacy systems), even in the world of EIS. Replacing systems with an iterative approach requires program management to oversee related projects in the aggregate. This oversight enables the firm to maintain business continuity. Each project, such as replacing an interface and infrastructure or the phased releasing of new systems, is controlled by the project management. Such a project management technique is analogous to redevelopment in a specific area. In the case of redevelopment, the following sequence is required. First, an alternative area for companies who do business in the area and for people who live there must be secured. Second, once these stakeholders are moved, the building begins. In the case of EIS, it is necessary to switch to new systems, while keeping the existing systems running. Program management is required in the case mentioned above, in which several projects run simultaneously and each project relates to the others.

3.4.4 Case of Dow Corning and FoxMeyer

Scott and Vessey (2002) studied ERP implementation projects at both Dow Corning and FoxMeyer that were carried out in almost the same time period by a comparable method. Dow Corning succeeded in implementing with difficulty, but FoxMeyer failed to do so and went bankrupted. According to this analysis, there are some risk factors for FoxMeyer. A major factor is that FoxMeyer made a contract with Universal Healthsystem Consortium (UHC) to improve its depressed revenue caused by the bankruptcy of its major customer Phor-Mor. The other factor is that FoxMeyer failed to build an interface to the automatic warehouse system that had been initiated prior to the ERP project. Risk factors for both companies are summarized in Table 3.3.

Table 3.3 Risk Factors in Dow Corning's and Foxmeyer's Implementation

Risk Factors	Dow Corning	FoxMeyer
External Business Context Characteristics	Threatened by breast implant lawsuits; competitive pressures	Unstable; cut throat competition; Acquisitions and mergers
Information Systems Characteristics ES	Concerns with processing large numbers of global users	Attempted to assess transaction volume; in the end, proved to be inadequate
Non-ES	Omitted bolt-ons, for example, to bar scanners, and so forth, to meet project deadlines	Conversion problematic; numerous data errors (large $ losses) Problematic; numerous data errors (large $ losses)
ES Project Characteristics	Reduced scope when needed Strong project leadership	Inappropriate reaction to new customer; compressed schedule; did not test adequately Who was in charge? Project leadership inadequate

Namba and Iijima (2004a) analyzed this case from a city planning framework. Concerning the first factor, they wrote that FoxMeyer failed to reflect an essential change to the information systems service that occurred in the marketing architecture, which is a part of the business layer. The latter factor they attribute to the fact that two projects were treated as a separate independent project. In this instance, FoxMeyer should have treated them as related projects and managed them coherently. From the viewpoint of EIS architecture, the former factor is an issue of the structural viewpoint. Unless the information systems service layer matches the business layer, the information system cannot support the business. The latter case corresponds to the part and whole viewpoint. The two projects should have a close relationship and actually interface to effectively collaborate with each other.

Generally this kind of case is analyzed in the context of project management. At the same time, however, it involves architectural issues. On this account, an architectural perspective is necessary, or the project will not fulfill its primary goal even if it succeeds. Therefore, an EIS city planning approach that includes not only architectural issues but also involves migration planning will be applicable and effective. This is because architecture and migration planning are inseparable.

3.4.5 Dissimilarities

The author has discussed the framework of the EIS city planning approach in terms of the similarities between city planning and EIS architecture. There are, however, some essential dissimilarities between them. First, there is the obvious difference between real objects (such as buildings) and cyber information systems. Though analogy has been applied only as a means to visualize EIS architecture, the limitations of this analogy must be understood. When EIS forms spaghetti and IS staff are burdened with maintaining the systems, users often do not realize the seriousness of the situation because spaghetti structures in information systems are unseen objects for them.

A second difference is related to the time scale involved in each project type. City planning focuses on a 10- or 100-year life span for the resulting product. In the world of information systems and information technology, where Moor's law reigns supreme, no such lifespan is expected. The speed of change for information systems is determined by the pace of change in business. Furthermore, the success or failure of information systems can immediately affect the achievements of the enterprise. To respond to this situation, a system to review the plan every several months and to execute appropriate changes is required. Since executing new processes with zero-based review is impractical, formulating an EIS scenario that allows for adjustment to the actual situation is not only effective but necessary.

A third difference between EIS architecture and city architecture is one of profitability. As a city plan is typically a public works project, citizen satisfaction, consensus building among stakeholders, and resource allocation are the major issues of concern. EIS planning, however, must account primarily for profitability, as this is the main purpose of a business enterprise. While acquisition of land in an early stage of the project in a physical city planning scenario might be acceptable, the acquisition of technology prior to starting an EIS project could be unwise because the technology may become obsolete before the project is completed. In EIS planning, the timing of investment matters.

3.5 Summary of the Chapter

This chapter proposes a framework for an EIS city planning approach comprised of an EIS architecture and an EIS scenario. The approach leverages the methodology of city planning in civil engineering to provide a framework for creating an EIS. First, an EIS architecture that follows the architectural guidelines provided by the IEEE is illustrated. Architecture is described by view(s) that conforms to viewpoints rendering stakeholders' concerns. Three specific viewpoints are focused on: structure, part and whole, and ins and outs. To achieve an organic aggregation of each project and each stakeholder concern included in an EIS, an EIS scenario that specifies a migration plan as a mechanism of program management is outlined. Then the validity of the framework and discuss its applicability is discussed.

The framework is intended to offer management and CIOs a useful methodology for designing EIS architectures from a top-level view of the enterprise that takes into consideration several stakeholder viewpoints. The framework is also intended to provide a metaphor for visualizing otherwise unseen EIS architectures. Since it is only an analogy, the city planning concept will lose its meaning or value in due time. Furthermore, the author also has to clear the criticism of Ross discussed in Chapter 2. For this reason, the EIS city planning approach that introduces an architecture point of view and takes over the "entire branch of knowledge" that has been missing in software construction has been proposed. In addition, viewpoints to define architecture and create an EIS scenario are used, taking over the methodology and know-how of physical city planning.

To illustrate the framework, the author uses the case of ERP implementation for Dow Corning and FoxMeyer. These cases show that un-matching layers sometimes cause a firm to go bankrupt and also show the vulnerability of parallel projects among parts without coordination by the whole.

References

Catanese, A.J., and Snyder, J.C. *Urban planning*, McGraw-Hill, Inc. 1988.

Ewusi-Mensah, K. "Critical issues in abandoned information systems development projects," *Communication of the ACM*, Vol.40, No.9, pp.74-80, 1997.

Fukukawa, Y., *Zoning and master plan: Planning and control of land-use in the United States*, Gakugei-Shuppan, 1997. (in Japanese)

Hall, P. *Cities of tomorrow*, Blackwell, 1988.

IEEE-Computer Society, "IEEE recommended practice for architectural description of software-intensive systems," *IEEE Std 1471-2000*, 2000.

IT Associate Council, "Optimization Plan for Business & Systems Ver. 1.1," 2003. (in Japanese)
http://www.meti.go.jp/policy/it_policy/itasociate/it.associate.htm

Namba, Y. and Iijima, J. "Proposal of 'EII meta-model' as integration framework for enterprise information systems," *Journal of the Japan Society for Management Information*, Vol.12, No.1, June, pp.15-32, 2003a. (in Japanese)

Namba, Y. and Iijima, "EII meta-model on integration framework for viable enterprise systems," *Journal of Systems Science and Systems Engineering,* Vol.12, No.1, March, pp.111-126, 2003b.

Namba, Y. and Iijima, J. "Evaluating enterprise resource planning implementation to enterprise information system through city planning approach," *Proceedings of the Japan Society for Management Information, Spring 2004*, pp.98-101, 2004a. (in Japanese)

Namba, Y. and Iijima, "A study on integration of enterprise information systems based on EIS city planning approach," *Journal of Information Processing Society, Japan*, Vol.46, No.3, in press, 2005. (in Japanese)

Namba, Y. and Iijima, "A study on IT city planning approach for enterprise information systems," *Proceedings of the Japan Society for Management Information , Spring 2003*, pp.268-271, 2003c. (in Japanese)

Namba, Y., and Iijima, J., "Evaluating enterprise resource planning implementation to enterprise information system through city planning approach," *Proceedings of the Japan Society for Management Information, Spring 2004*, pp.98-101, 2004a. (in Japanese)

Namba, Y., and Iijima, J., "City planning approach for enterprise information systems," Proceedings of the Eighth Pacific Asia Conference on Information Systems, Shanghai, China, pp.169-180, 2004b.

Parr and Shanks "A taxonomy of ERP implementation approach" Proceedings of the 33rd Hawaii International Conference on Systems Science 2000, 2000.

Putman, J., *Architecting with RM-ODP*, Prentice Hall, 2001.

Rogers, E.M., *Diffusion of innovations: Fourth edition*, The Free Press, New York, 1995.

Sakamoto, I. *Urban and life space engineering*, Housou-Daigaku-Kyouiku-Shinkoukai, Tokyo, 2000. (in Japanese)

Scott, J., and Vessey, I. "Managing risks in enterprise systems implementation," CACM, Vol.45, No.4, pp.74-81, 2002.

Sewell, M. T., and Sewell L.M. *The software architect's profession: An introduction*, Prentice Hall PTR, 2001.

Shirvani, H., *The Urban Design Process*, Van Nostrand Reinhold Company, New York, 1985.

Software Engineering Institute, "How Do You Define Software Architecture?" 2004. http://www.sei.cmu.edu/architecture/definitions.html

The Chief Information Officers Council, *A Practical Guide to Federal Enterprise Architecture Ver 1.0*, February 2001.

The Chief Information Officers Council, *Federal Enterprise Architecture Framework*, September 1999.

The Open Group, "Other Architectures and Architectural Frameworks," 2001. http://www.opengroup.org/architecture/togaf7-doc/arch/p4/others/others.htm

Chapter 4

Case Study to Focus on Evaluating the Framework

4.1 Introduction

In Chapter 3, the framework of the EIS city planning approach was explained conceptually. In this chapter, the author illustrates the framework using three cases—KDDI Corp. (KDDI), the Ebara Corp. (Ebara), and Monex, Inc. (Monex)—and discusses the applicability and effectiveness of the framework. KDDI builds its systems with the concept of "object orientation." The company designs a conceptual data model first and then builds systems along with it to realize its business model. Ebara integrates its information infrastructure based on a fundamental policy of reforming the firm's information systems. Monex illustrates an actual case in which various network connections exist across the boundary of the firm.

These cases are based on the information from the presentation materials in the special interest group (SIG) or the symposium in the academies to which the author belongs, as well as articles in magazines and an intensive interview with the key person from each of the corresponding cases.

4.2 The Case of KDDI Corporation

KDDI has undertaken several mergers and acquisitions since 1998 to keep the firm competitive. KDDI was established in 2000 with the merger of KDD (an international phone and network company) and DDI (a local phone and network company). In 2001, the company merged "au" (an mobile phone company) that was itself created by a merger of seven local cellular phone companies. These business events forced the repeated integration of KDDI's information systems. Accordingly, the systems have grown large, complex, and rigid. This has resulted in increased:

- Systems development time and cost for new development,
- Cost for systems maintenance and operation, and
- Systems failure.

Moreover, it has led to:

- Obstacles to sharing knowledge and know-how,
- Increased man-power and undermined morale, and
- The information systems division being considered by the management as an inhibitor to the firm's reformation (Shigeno, 2004a; 2004b).

Having recognized that the essential issue was an aging effect of the applications caused by uncontrollable complexity, KDDI designed, developed, and implemented information systems to adapt to the changing business environment. Consequently, KDDI introduced the idea of object orientation, designed its information systems based on the conceptual data model, and rebuilt its mission critical back-end systems as a part of structural reformation for the enterprise information system. This concept and methodology incorporated the idea that the enterprise information system is a simulator of the real world as a "universe of discourse" (Shigeno, 2004a).

KDDI has divided elements of information systems into "variant" and "invariant" in order to realize its idea. In practice, the company classifies itsr information systems into three types: presentation systems, back-end systems, and informational systems. On that basis, it designs the conceptual data model in the back-end systems that control data directly as an invariant element based on its business model, and builds subsystems along with the data model. Using these structures, the company can modify and enhance its information systems per subsystem, adapting to change within the business environment. By consolidating user-interfaces as presentation systems, KDDI can accommodate the detailed change requirement for screens layout or design and allow users to print forms promptly. In addition, the informational systems include data-warehouse and data-mart systems that reorganize data in order to exploit it effectively.

To be fair, however, KDDI's local phone system was based on a DDI system built in 1987, which was, technically, the first system to apply this concept. This original system was a sort of experimental system. KDDI called this methodology "evolutional prototyping." Before then, almost all systems were called "silo" (Ross, 2003) or "stovepipe" (Schulte, 1997) applications, which were isolated independent applications.

Figure 4.1 shows the procedure used to build the company's information systems by the methodology and the concept of object orientation (Umeda, 2003).

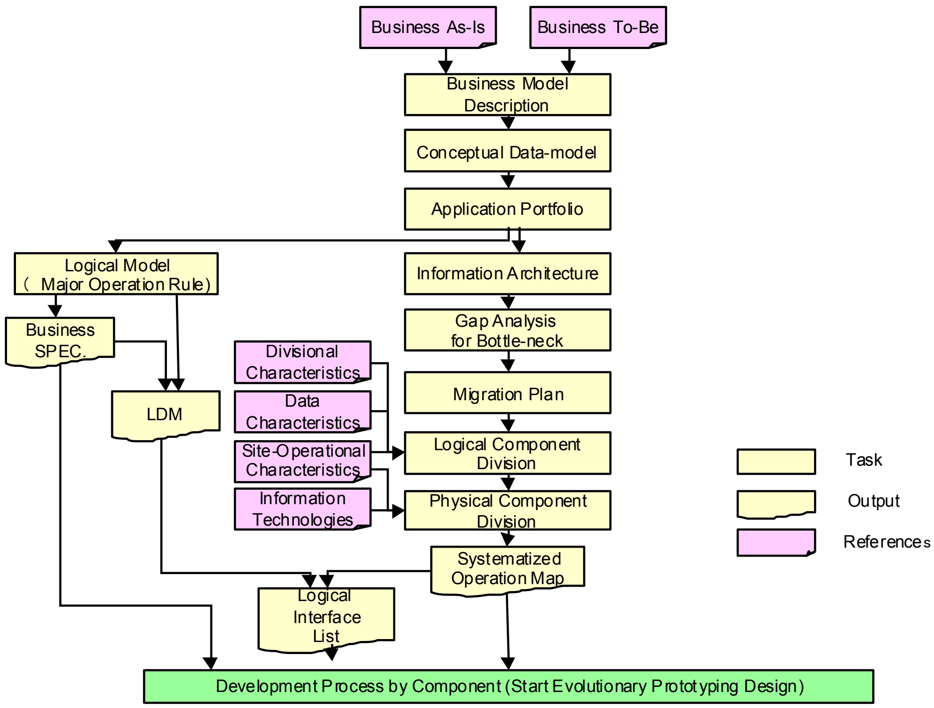


Figure 4.1 Flow from Designing Conceptual Data Model to Implementation

The steps for designing the conceptual data are as follows:

- An outline of the business model is described.
- A conceptual data model—which consists of a static model, a dynamic model, and an interaction model—is described, taking various entities and activities into consideration.
- The static model is portrayed in the entity relationship diagram. This model is based on entities, defining identifiers, major attributes, and relationship data abstracted at the EIS level.
- The dynamic model is described as the lifecycle of the static model from its generation to its dissolution.

In order to design an application portfolio, the conceptual data model is divided and partitioned after the administrative responsibilities have been clarified for each part of the static model and the transition of responsibility has been established based on the state of transition. The entities that have a close relationship in the conceptual data model are grouped and corresponding subsystems are defined. Then these subsystems are mapped on the application portfolio, which is defined by business characteristics and expected technology trends. On that basis, the relationships between subsystems are defined, collaboration functions between subsystems are deployed as collaboration middleware, and the information systems architecture is defined.

The gap in Figure 4.1 between the information systems architecture (future EIS) and the bottlenecks in the actual condition are analyzed and organized, and then the specific migration plan is made. KDDI selects not a big-bang approach but an iterative one, and integrates whole functions with a loose coupled technique. In this way, the company can avoid concentration of risks, proceed with its structural reformation in the ordinary operation, and maintain and enhance the systems per subsystem. When partitioning subsystems logically, optimal condition is defined by considering the requirements from divisional (organizational) characteristics and data. After defining the logical subsystems, physical partitioning is executed by considering the requirement of operation and monitoring current trends in technology.

KDDI defines subsystems from an adaptation-centric rather than a function-centric perspective by assuring independency in exploiting the fundamental concept of object orientation (information hiding and encapsulation). Thus, they use everything they can imagine. This includes:

- Hiding the entity that the back-end system controls from user-front systems,
- Hiding the physical configuration of the back-end system by using the hub,
- Hiding physical implementation by designing systems interfaces based on the conceptual data model, and
- Prohibiting direct data referring and updating between subsystems in the back-end system.

KDDI has implemented a “hub and spokes” type integrated message-interchange infrastructure (message hub) (Namba and Iijima, 2005). As Figure 4.2 illustrates, a basic structure of the message hub is its role in connecting each subsystem in the peer-to-peer base as a shared information infrastructure. Repository in the figure involves common processing methods and rules. KDDI deploys both internal and external hubs, the former for internal applications and the latter for external applications, including such systems as customer connection or settlement. Since Figure 4.2 is a logical structure, it is illustrated using one box. But in reality, it is implemented with several processes based on considerations of performance and ease of systems operation. The external hub includes a physically independent front-end processor (FEP) that transforms protocols between internal systems and external systems outside of fundamental hub functions.

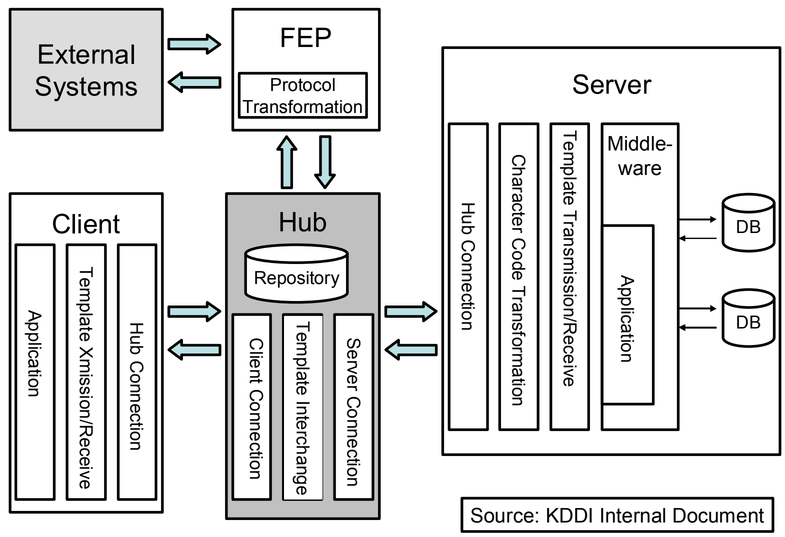


Figure 4.2 Message Hub of KDDI

The role of the hub, as an integrated information infrastructure, basically involves interchanging data between each subsystem. Moreover, it ensures the independence of each subsystem, since the subsystems are used only to interface to the hubs. KDDI's hub has a standard message form for transmission as a data interchange standard. The standard message is composed of a header part, a common message part, and a template. The hub routes messages and transmits them to each destination application. This template function corresponds to a building code in physical city planning and is one of the basic functions of zoning. In order to assure interoperability in the enterprise, infrastructure is constructed first, and then buildings are constructed in connection with the infrastructure.

Unlike ordinary procedure designing and implementing applications, KDDI's system is characterized by taking the path shown in Figure 4.3. KDDI first creates a conceptual data model that can describe the business model of the firm in order to simulate a structure of the real world. Second, the company builds the back-end part of the systems based on the data model. Third, the application program as a variant element is built, which serves as a prerequisite for the existence of the back-end system. Partitioning subsystems and interchanging messages only through the fixed interfaces (hubs) enables not only scalability but the capability to delete, which assures the elimination of unnecessary applications and/or subsystems from the systems (Nikkei Computer Editorial Desk, 2004).

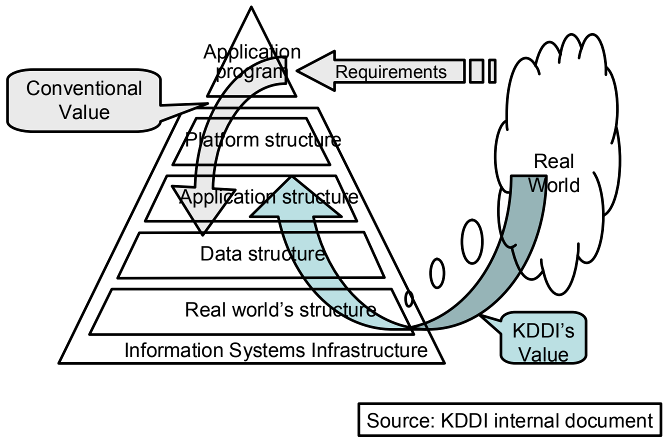


Figure 4.3 KDDI's Value of Systems Development

4.3 The Case of the Ebara Corporation

The Ebara Corporation has developed its business based on a technology of manufacturing pumps. Recently, its business includes not only pump manufacturer but a prime contractor of plant construction. In the process of developing business, the company has included multiple production style such as:

- Job order production,
- Mass production,
- General contract for plant design and construction, and
- Design-to-order production for plant equipment.

Moreover, applications have been developed for each plant that has a different production model (i.e., business model) individually. Information infrastructures have also been deployed for each application. These situations have become obstacles to the expansion of the company's business, due to the complexity of the information systems. This has triggered a project to rebuild the corporation's information systems, in order to create an architecture that maintains the consistency of its business activities.

Ebara aimed at integrating the systems at the level of all applications. The company started a project named e-HUMAN’s (Engineering Hub of Management and Action for New Generations) for this purpose. The project included four sub-projects:

- Project Management System (PMS),
- Material Management System (PMS),
- Engineering Document Management System (EDMS), and
- Information infrastructure support team.

Since changing all relevant application systems with consistency and coherence is difficult, the company made two items a priority, so that the project could proceed. These items were:

- Integrate the information infrastructure, and
- Then integrate data.

According to Tsuji and Oda (2004), Ebara chose to migrate its information processing environment from an existing mainframe-centric server environment to an open architecture server environment as the first stage of the project. Beginning in 2001, Ebara implemented a physical integration of database servers and UNIX application servers in order to consolidate the uniform relational database management system (DBMS), OS, middleware, and programming language. In the process, however, the company had to improve its management level of systems operation, since the different service time, the target person of the service, and troubleshooting each system could bring confusion and trouble to the systems operation. In order to improve the situation, the company decided to reorganize the information infrastructure as an open system, including improvement of the systems monitoring and operation management.

At first the company designed a layered structure for the information systems, as shown in Figure 4.4. To design this structure, the company decided on a direction for an operation design and a systems operation. For the operation design, the direction involved:

- Fundamental reform of the operations design with three *R*s (rule, route, and role);
- Enhancement of the quality and effective use of information; and
- Integration and optimization of the information systems.

The changes for the systems operations involved:

- Proper design and functional layout throughout the whole system,
- Normalization and appropriate layout of the database, and
- Application of a unified and highly reliable infrastructure technology.

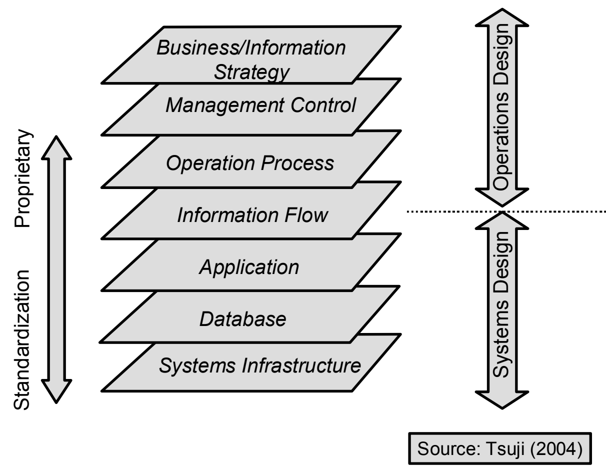


Figure 4.4 Structure for Building Information Systems

Ebara's information infrastructure includes a network, authentication system, portal system, common interface, integrated servers, and systems monitoring and operation management systems. The company designed and implemented an integrated security system over this infrastructure. A common interface in Figure 4.5 is illustrated as an example of the information infrastructure.

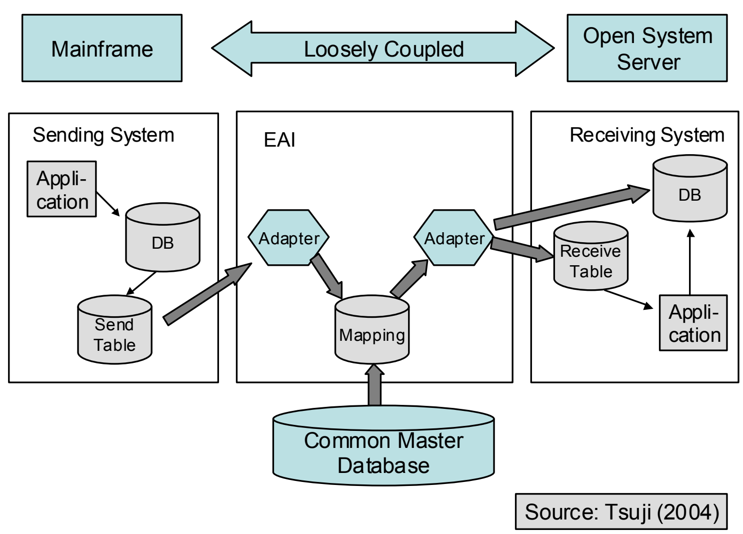


Figure 4.5 Common Interface

This interface aims at:

- Improving and visualizing a maintainability and an operations ability by unifying the interface mechanism,
- Seamlessly integrating the legacy system and open system with loose coupling technology,
- Shortening the development period and reducing the cost for systems development,
- Improving data quality through consolidation management of common master data, and
- Defining the person in charge of the master data.

First, the company envisioned the systems operation service as providing "the division and the department of Ebara and their affiliated companies for a systems operation service with a high quality and a strong competitiveness at the optimal cost as a finally responsible division of the service" (Tsuji and Oda, 2004). In order to realize a defined administrative process as a future model of systems operation service, the company decided to implement a workflow system and operation monitoring system, as well as rebuild an asset management system.

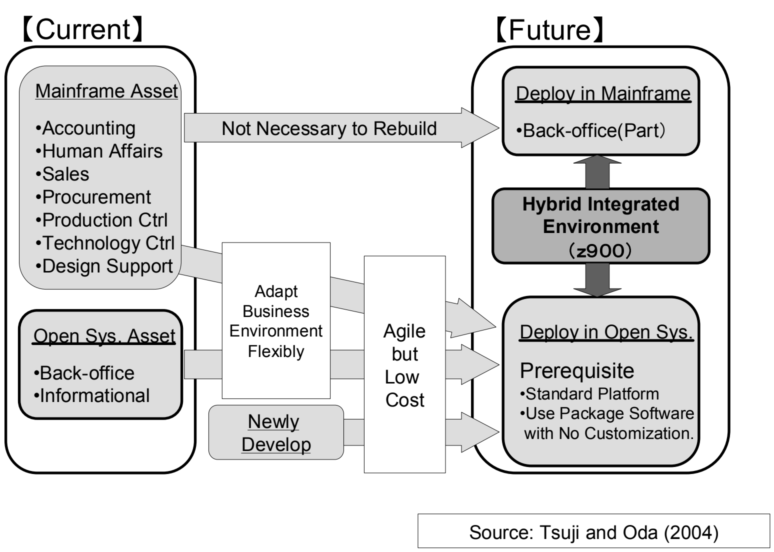


Figure 4.6 Server Integration

In the process, the company decided that:

- Assets in the mainframes that did not need to be reformed should still run under the same environment, because operations process and need does not require change.
- Assets included in both mainframe and open systems that dealt with business environment flexibly, as well as new developments, should be developed in the environment of the open system.

Figure 4.6 illustrates these situations. The company introduced a hybrid type integrated environment that can run both environments together, thus operating efficiently and at a reduced cost. The hybrid type integrated environment means that systems with different operation systems (OSs) can run together under the same platform. The company introduced a z900 series server machine provided by IBM, which was partitioned logically and allowed several numbers of OS390s and Linuxs on the same platform.

Under this integration environment, the company made a systems operation schedule, built shared monitoring systems, and operated these systems. The company explained, "Supposing we took an initial cost and compared for each application, the result would be different. Since we took a five years lifetime cost, including maintenance, systems operation, and training of human resources, the integration environment showed an advantage" （Tsuji and Oda, 2004）. Thus, the lifetime assessment explains why the company deployed a unified platform with multi-OSs environment (z900 series) instead of deploying the systems environment per each application. Some of the initial deployment costs, including hardware costs, for this solution were rather expensive than the costs for deployment of each application separately, though.

Ebara's environment realizes a physically integrated infrastructure in spite of completely independent applications in the layer of information systems service. This relation is relevant to the "trade-off" between layers in the structural viewpoint. The company decided upon a cost structure based on not only an initial acquiring cost and implementation cost, but with TCO over a lifetime (including an additional maintenance cost and a systems operation cost).

This case illustrates how to make an overall framework as a whole first and then build respective items as a part when rebuilding information systems as an enterprise information system. In addition, it includes grasping the integrated infrastructure not as individual applications but as a common infrastructure, and evaluating the cost for maintenance and systems operation cost with a lifetime base. These concepts precisely correspond to the proposed methodologies in terms of exploiting the EIS scenario when a firm builds the integrated information infrastructure by applying the part and the whole viewpoint. Recently, in many cities, various public facilities that have been built through dependence on temporary incentives are burdened with rising maintenance costs and growing budget deficits. Looking at the information systems, firms should evaluate with TCO of lifetime bases, including not only an initial building and implementing cost but the cost of maintenance, systems operation, and disposal.

4.4 The Case of Monex Inc.

Triggered by the deregulation of stock commissions beginning in October 1999, online securities businesses that provide trading environments to their customers through the Internet are in full swing in Japan. This industry is characterized by its bilateral character; a business is both a financial company and an e-business. The features of a financial company require precise and secure execution and compliance with laws and regulations. This is necessitated by the large amount of money that flows through a securities firm. Distinct from a financial company, an e-business requires that business processes migrate from the traditional branch office and face-to-face oriented sales systems to virtual sales systems that utilize information technologies. Thus, this requires fulfilling non-functional requirements, such as availability, scalability, performance, and security, as well as functional requirements that correspond to the needs and services for real business.

A few months after Monex started its business, a performance problem emerged at the peak time of system load. This resulted from an underestimation of capacity planning, because, at the first stage of its business, nobody could forecast the actual influence and result of its drastic drop in commission, which was down about one tenth compared to the commission before deregulation. Trying to tune its systems, the company enhanced the system’s capacity. While the company could have scaled up and/or out a new systems infrastructure to meet the peak load, it had to consider cost. “Scale up” means to replace the hardware with a higher specification one, and “scale out” means to add hardware(s) to the existing environment.

Since cost structure matters, especially for online businesses for which one competitive edge is low cost, many companies cannot invest enough money in their systems. Monex decided to go for both system capacity and cost. For its mission critical system, the company built by in-sourcing. For its information quote service that had been provided the same systems infrastructure as transaction processing system, the company utilized the ASP service provided by external vendors. The company did this because information quote service does not require a mission critical characteristic. Thus, ASP as a professional service that uses cheap machines can be considered to provide high quality and inexpensive service to the customers.

From the perspective of functional requirements for EIS, an information system that supports doing an e-business (e-business system), consists of two functions: (a) a front system that covers a direct interface to the customers through the Internet, and (b) a back-office system that sustains business activities as a securities company. These two functions collaborate with each others to form an online trading system. In addition, in terms of the systems integration in relationship to the external systems that lie beyond the boundary of the firm as a B2B connection, an e-business system is considered a virtual enterprise that has functionally integrated the conjugated aggregation of information systems beyond the extent of a single enterprise. When considering the architecture of an e-business system from the EIS perspective, it follows that the firm should think about not only each application to handle various functions, but also the architecture to describe entire systems as an integration of applications.

Trading or brokering business on securities is better adapted to online business, because the business, by its nature, transfers the right of ownership. Thus, the shared operation systems have been established before the online brokerage business has flourished to some degree. For instance, the back-office system that takes charge of booking and so forth has been prepared as a shared service by, for instance, the Nomura Research Institute (NRI) or the Daiwa Institute Research (DIR) for securities firms that could not deploy the systems by themselves. The securities firms can transfer the data to the Japan Securities Depository Center, Inc. (JASDEC), which performs the beneficial shareholder notification without processing the name transfer operation in each case through the back-office system. Firms also connect to the settlement or the clearing organization through the same path. Moreover, they exploit the various information systems services for securities business, such as the stock quote information from information vendors. Since the deregulation of the stock trading commission, these tendencies have been accelerated and the integration infrastructure among related firms as shared services have been established with the emergence and the rapid diffusion of online securities business.

Figure 4.7 shows the actual collaboration and/or integration of information infrastructure between the Monex system and the systems outside of the firm.

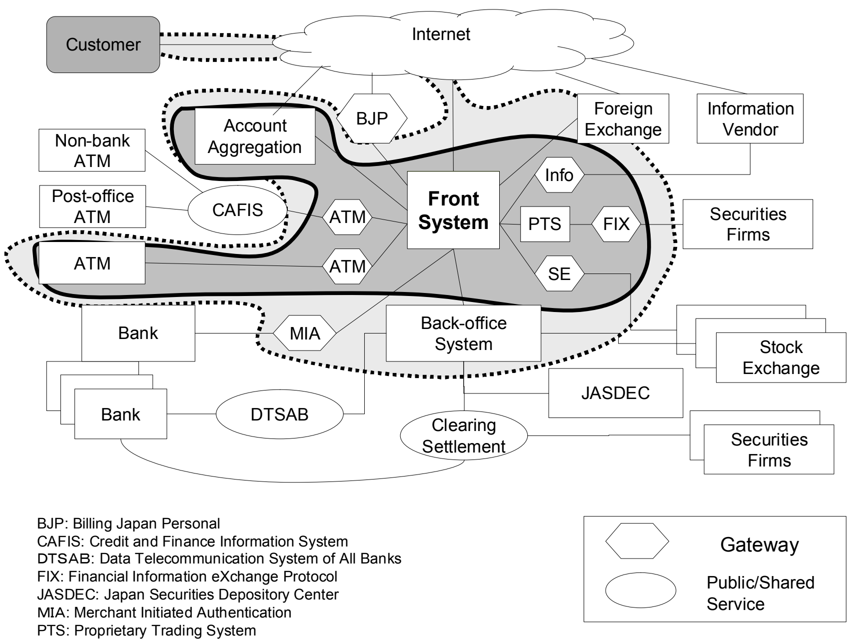


Figure 4.7 Ins and Outs of Monex System

The Monex system, as an asset, includes the area surrounded by a solid line in Figure 4.7. It includes a dotted line from the perspective of systems operation, as well as all the functions in the figure from the perspective of the customer of Monex. Thus, it can be said that the Monex system for the customer includes virtually all of the area in Figure 4.7, because it is integrated with various functions shown in Table 4.1. Table 4.1 shows the situation of systems integration for an online securities company along with the classification of Table 3.1. Monex deploys two kinds of protocols to connect the back-office system: one is an MQ series (IBM product) for online connection and the other is HULFT (Host UNIX Linkage File Transfer) format for batch transfer. For the connection to banks or securities firms, they exploit common protocols such as the Credit and Finance Information System (CAFIS), which is a shared service type credit online system, and the Financial Information eXchange Protocol (FIX), which is a standard international protocol for securities business. For the account aggregation system or the information quote service, the company uses a front-end integration technique that can integrate in the presentation layer. A Monex customer can utilize stock information in the frame of the same browser by clicking.

Table 4.1 Systems Integration for On-line Securities Company

	Category	Topology	Tier	Protocol	Others
Account Aggregation	request/response	hub and spoke	presentation	http/shttp	async
Collaboration to Quick Site	request/response	hub and spoke	presentation	http/shttp	async/ loose coupled
ATM Connection (CAFIS)	messaging	hub and spoke	presentation	CAFIS	sync
Back-office System Connection(1)	messaging	point to point	data	MQueue	async
PTS Connection	messaging	point to point	business logic	FIX	async/ loose coupled
MIA Connection	messaging	point to point	business logic	SET	sync
ATM Connection	messaging	point to point	business logic	not open	sync
Stock Exchange Connection	messaging	point to point	business logic	proprietary	async
Back-office System Connection(2)	reconcile	point to point	data	HULFT	batch

CAFIS is a registered mark of NTT Data, MQ Series is IBM, and HULFT is Saison Information Systems.

Thus, online securities systems exploit the shared service or public infrastructure, which enables them to: (a) form virtually integrated securities systems centering on an in-house developed front system and an outsourced back-office system, and (b) provide service to the customer. Various integration techniques are deployed as a fundamental technology to connect each infrastructure. The integration technologies in the online securities business have been fundamental technologies as a shared or a public infrastructure. For the back-office system that is a logical intra-enterprise system, the business exploits its proprietary file format for data transmission. ATM, MIA, or the connection to the securities exchange as a public infrastructure, however, exploit a designated file format. Thus, actual implementation differs inside and outside of the firms, as the ins and outs viewpoint illustrates.

4.5 Discussion of Applicability and Effectiveness

4.5.1 Structure Viewpoint

Brooks (1995, p.183) mentions that "the complexity of software is an essential property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence." The term "accident(al)" does not mean "occurring by chance," nor "misfortunate," but more nearly "incidental" or "appurtenant." In the context of EIS, Brook's essential meaning might be how to describe or realize business architecture of the firm in EIS, since EIS is abstracted from individual information systems as an organic aggregate at the level of enterprise. This means to map business architecture into the business layer of the structure viewpoint. Then the information systems service layer, which is composed of an application portfolio and information systems services, is constructed under the business layer. Moreover, the integrated information infrastructure layer is built. At this juncture, trade-offs in how to realize the given conditions, whether in the information systems service layer or the integrated information infrastructure layer as a common function, should be considered.

KDDI creates the conceptual data model as a core of its information systems from the described business model under the recognition that information systems simulate the real world. Then the application portfolio, the information systems architecture, and the migration plan are created based on this conceptual data model, and the sub-system partitioning is performed. KDDI's process may correspond to the structure viewpoint, as below:

- The conceptual data model corresponds to the business layer,
- The application portfolio corresponds to the information systems service layer, and
- The infrastructure in the IS architecture corresponds to the integrated infrastructure layer.

Similarly, in Ebara, the company describes its structure of information systems in seven layers, as shown in Figure 4.4. In this case, each layer may correspond to the layer of the structure viewpoint as follows:

- The layers from the business/information strategy to the information flow correspond to the business layer,

- The application and database correspond to the information systems service layer, and
- The systems infrastructure corresponds to the integrated information infrastructure layer.

In addition, Ebara creates a hybrid environment in the infrastructure domain. From the perspective of a logical model, the company deploys each function on each environment. From that of physical model, the company unifies an environment and takes the infrastructure service as shared service. In this case, though each application is independent from others in logical meaning, the servers in which each application runs are virtually integrated as the physical deployment and are maintained and operated under the unified procedures. Thus, this case shows that logical design and physical deployment can be composed as completely different forms.

In the case of Monex, the company has not built any systems by itself. Instead, it has outsourced not only systems development but also systems maintenance and operation, including infrastructure. The only thing that Monex does is design the application and the service portfolio based on its business model. As an asset, Monex owns the front system in Figure 4.7, but outsources the systems operation to a managed service provider (MSP). Figure 4.7 shows the difference between logical view (application and service portfolio) and physical view (infrastructure).

4.5.2 Part and Whole Viewpoint

KDDI partitions the information system for the implementation using the following steps (shown in Figure 4.1). First, the company designs the architecture from the whole viewpoint. Second, it constructs the migration plan. Finally, it partitions the plan logically, drawing upon the characteristics of divisions, and doing it physically, using the applicable information technologies. The basic idea of this procedure is to determine fundamental elements of the whole, to assure interoperability, and then to optimize from the enterprise perspective by considering individual conditions. Moreover, from the perspective of the IT governance to maintain systems, KDDI has established a corporate organization for data administration. The company has the authority to examine every activity for building new applications or maintaining or enhancing existing applications from the perspective of data management. This system assures the consistency and interoperability of each application in the whole system. The reason the mechanism to collaborate between applications is used not on a peer-to-peer basis but via messaging hub as a common infrastructure of the firm might be seen as the same philosophy as the part and whole viewpoint.

Ebara's server integration would not have succeeded if the company justified the cost with each application basis. It, however, evaluated it with a consolidated and lifetime cost basis. This case does not show a conflict of interest, but rather it shows that even the aggregate of the systems that have less relation in the information systems service layer can receive the benefit of consolidation if it is considered in the integrated information infrastructure layer. One more example of part and whole viewpoint is the company deploys the common interface as a share service infrastructure. Although KDDI's messaging hub uses the same idea, this interface aims to interchange the data between systems, including mainframe systems and open systems server. This common interface protects each application from forming complex and complicated structure if the company can manage the development discipline properly.

4.5.3 Ins and Outs Viewpoint

KDDI uses two kinds of message hubs: an internal hub and an external hub. The internal hub connects between the front systems for intra-enterprise and the back-end systems. The external hub goes between the front systems for outside of the enterprise such as customer portal, net-banking system, etc., and the backend mission-critical systems. The reasons to separate hubs are not functional issues, but rather administrative and interface ones. Administrative issues include systems monitoring and operation, systems maintenance, service level management for performance, availability, security, and so on. Having an interface issue means that the company cannot use its own specification, but should adjust to an agreed or common/standard one. On this occasion, it should fit to the semantics level of data as well as the syntax level. For the information infrastructure, it should adapt from the physical layer, like network line and devices, to the systems monitoring and operation. The operation affairs should extend beyond the legal or business boundaries and also consider the escalation system collaborating with the connection party to prevent incurring a failure or trouble. The customer portal that must cover several million customers, for instance, requires intense availability and performance that are at completely different levels from those for intra-enterprise systems. Thus, different levels of service and organization for external service are required, because of service time or in order to maintain service level. Thus, an ins and outs viewpoint is related to the controllability and interface to the connection party, on the assumption of setting proper boundaries.

The case of Monex shows an actual example wherein various applications and services provided outside of the firm form a logical virtual enterprise system. The view from a logical layer and that from a physical layer highlight different aspects. If it is looked at from the perspective of ownership, it appears completely different. In the integrated information infrastructure, ins and outs show completely different features. For the external connection (for example, to the stock exchange or banks), the company has to use designated or common protocols, while it can use some proprietary protocols for the logical intra-enterprise systems such as back-office system connection. This contributes to the controllability of the firm.

4.5.4 EIS Scenario

In order to realize the architecture, KDDI seems to execute its plans using the same idea as an EIS scenario. For example, when it analyzes the gap between the present and future information systems and makes a feasible migration plan, this is the same thinking as in an EIS scenario that sets up a live-to-be architecture and makes a migration plan in the city planning approach. Avoiding a big-bang approach and proceeding with iterative reformation in the ordinary operation corresponds to the combination of small- and mid-size development plan and zoning, and this might mean steadily putting forward redevelopment in the context of the EIS scenario.

When Ebara decided to integrate the servers among mainframe systems and open systems, the company wrote the migration scenario in which it set the basic policy to reform its systems environment. In order to justify the IT investment, it took a TCO over lifetime base analysis from the whole viewpoint. This process is in accordance with the idea of the EIS scenario.

4.5.5 Evaluation of Applicability and Effectiveness

In order to evaluate the applicability and effectiveness of the EIS city planning framework, the structure of the framework will be now compared to that of actual cases. In this section, the following criteria are extended:

- Clarify as-is, to-be, and live-to-be architecture of the cases from each viewpoint. For the structure viewpoint, the three layers are independent from each other, with the information services layer reflecting the business layer. For the part and whole viewpoint, the whole and the part are separated, or the role of whole is defined. For the ins and outs viewpoint, the boundary of the enterprise is defined and the applied technologies are distinguished for use.

- Clarify the role of the EIS scenario.
- Evaluate applicability and effectiveness of EIS according to whether it can reveal the difference and correspondence between the framework and the actual case.

In this section, the cases are discussed from the three viewpoints of EIS architecture. Tables 4.2 – 4.4 show how each of the viewpoints are realized or implemented according to the criteria in each of the cases. These tables reveal that the differences between the as-is, the to-be, and the live-to-be architectures of each case and illustrate the role of the EIS scenario. In these tables, “as-is” means the state of the before activities, “to-be” means the ideal state at that time, and “live-to-be” means pro tempore goal considering constraint. As a result, it can be said that the framework is applicable to the actual cases and effective in explaining the success factor.

Table 4.2 KDDI's Case

	As-Is	To-Be	Live-To-Be
Structure Viewpoint	- Consists of aggregate of disparate, "silo" type, unlayered structures of partially formed spaghetti.	- Systems with layered structure. - IS Service layer should reflect business layer and build based on common infrastructure.	- Conceptual data model-centric systems design, development, and implementation.
Part and Whole Viewpoint	- Consists of systems-only parts that do not reflect a clear conception of whole.	- Systems have a structure that includes distributed autonomous parts forming the whole in harmony.	- Corporate data administration department controls data schema. - Implementation of tem-plate as a common message for interchange. - Implementation of hub as a shared infrastructure.
Ins and outs Viewpoint	- Each application system has its interface with outer systems, if required.	- Properly role-shared deployment of infrastructure between ins and outs.	- Separate implementation of internal and external hub.
EIS Scenario	- Whole direction is decided upon and then the local phone business system is started. - Other systems follow business reformation for that domain. - Migration plan targeted to live-to-be architecture. - Not taking a "big-bang approach," but an iterative one (evolutional prototyping method).		
Background	- Repeated M&A made information systems complex, rigid, and hard to maintain. - Ageless, adaptive information systems integrated at the whole firm level were required in order to be an enabler of business.		

Table 4.3 Ebara's Case

	As-Is	To-Be	Live-To-Be
Structure Viewpoint	- Aggregate of "silo" type and mainframe-based systems partially integrated for sales.	- Seven-layered structure of Information systems.	- Infrastructure rebuilding plan from the perspective of whole enterprise. - Deployment of common interface.
Part and whole Viewpoint	- Almost no whole function; "distributed part" systems.	- Distributed applications with unified data based on shared infrastructure.	- Integrated server environment established across the firm. - Hub deployed as a shared service.
Ins & outs Viewpoint	-	-	-
EIS Scenario	- Integrated server environment deployed as a shared service based on cost justification over lifetime. - Integrated infrastructure deployed first, then applications on the shared infrastructure are implemented.		
Back-ground	- Firm's system consists of disparate and legacy-based information systems with different business models. - Each application is distributed at each plant.		

Table 4.4 Monex' Case

	As-Is	To-Be	Live-To-Be
Structure Viewpoint	- No system.	- Establish systems of shorter release time to meet business requirements. - Data model based on conceptual data model that reflects business model.	- System has independent layered structure. - Data structure is ignored to meet speed of business.
Part & whole Viewpoint	-	-	-
Ins and Outs Viewpoint	- No system.	-Best mixed module system among internal and external functions.	- Utilization of common or shared infrastructure services outside of the firm. - System consists of mixture of internal and external modules.
EIS Scenario	- First selection is timed. - Mission critical system is developed as in-sourcing. - Non-mission critical functions are developed or outsourced by comparing quality of service, cost, and time.		
Back-ground	- Online trading system cannot exist without connecting external systems such as stock exchange system, clearing system, etc. - Online business requires a robust and high availability information infrastructure. - Data structure has become complex because of "speed first" development.		

4.6 Summary of the Chapter

In this chapter, the author has studied and discussed the applicability and effectiveness of the EIS city planning approach framework with the cases of KDDI, Ebara, and Monex. The structure of the framework and that of the cases are compared from the perspective of three viewpoints, and migration path is analyzed by comparing the concept and the methodology of an EIS scenario.

KDDI has adopted the concept of object orientation to build an "ageless (rust-free)" information system, and has started with the construction of a conceptual data model that can simulate the real world. This approach is based on a similar idea to the framework of the EIS city planning approach, which can adapt to a change in business environment with agility and can build an EIS as an enabler of business. Thus, designing EIS architecture centering on three viewpoints—structure, part and whole, and ins and outs—is effective in pointing out what needs to be considered before implementation, in positioning in the whole, and for designing interoperability.

Ebara has accomplished a hybrid type integrated environment on which it physically implemented both mainframe systems and open systems, while the applications are less related to each other. In order to deploy this infrastructure, the company evaluated an IT investment justification with consolidated TCO throughout the lifetime, including the cost of maintenance and systems operation as well as the initial cost. If the company had evaluated on the basis of each application, the result would have been completely different. The concept of an EIS scenario is applicable for this purpose.

Monex has implemented its online trading system with a select and assembly basis. This system has a completely different view from that of the logical systems viewpoint or the physical systems viewpoint. This system illustrates different features of the interface, whether it is connected to the inside of the firm or the outside. This contributes to the controllability issues of the firm.

The applicability and effectiveness of EIS have been discussed by comparing the framework to the structure in each case. According to these cases, the characteristics of each show similarities to the framework and the concept of the proposed EIS city planning approach. Consequently, the author thinks the applicability and effectiveness of the framework has been clearly demonstrated.

References

Nikkei Computer Editorial Desk, "Never Rust Systems Anymore," *Nikkei Computer*, 2004.7.12, 2004, pp.61-63, 2004. (in Japanese)

Namba, Y. & Iijima, J., "A Study on Integration of Enterprise Information Systems based on EIS City Planning Approach," *Journal of Information Processing Society, Japan*, Vol.46, No.3, in press, 2005. (in Japanese)

Shigeno, T. *System Integration SIG, Japan Society for Management Information*, 2004a. (in Japanese).

Shigeno, T. "System Integration in Era of Change", *Proceedings of Systems Integration SIG, Japan Society for Management Information*, 2004b. (in Japanese).

Tsuji, S. and Oda K. *System Integration SIG, Japan Society for Management Information*, 2004. (in Japanese).

Umeda, A., "KDDI Case," Modeling SIG, IS, IPSJ, 2003. (in Japanese)

Chapter 5

The Move toward a More Effective EIS City Planning Approach in Practice

5.1 Introduction

This dissertation has focused on the EIS city planning approach as it compares to physical city planning. Many people, however, point out that city planning itself often does not go well. Moreover, they question the applicability and effectiveness of the EIS city planning approach as a basis of these discussions. Actually, cases that are the result of poor city planning are often seen or heard. Considering these things, in this chapter, the author discusses why city planning itself seems ineffective. On that basis, the scope of applicability for the city planning metaphor is discussed. Then a new approach to city planning possibly that could break through current issues is illustrated, and whether it can apply to upcoming problems in EIS is discussed. Finally, the structure required of information systems is discussed to enable adaptive EIS by introducing KDDI's philosophy for building information systems.

5.2 Issues in Physical City Planning

5.2.1 Why Physical City Planning Seems Unsuccessful

The reasons city planning seems unsuccessful are as follows:

- It cannot follow the speed of change of the real world;
- It takes large amounts of time to reach consensus between owners of the project, local residents, and a city authority; and
- It is necessary to be consistent with plans of upper-level or neighboring areas.

Much of this failure may be attributed to rigid plans that cannot be changed once planning is underway. Even if it is possible to change, it either costs too much or takes too much time because of its inflexible nature. In addition, diversification of the citizens' requirements to be sufficient for city functions accelerates this tendency. This leads to an objective function for city planning that is more complex, making it more difficult to meet each citizen's requirements.

These issues are also applied to the area of the information system in a firm:

- The information system cannot follow the speed of change of the business environment;
- It takes a time to meet users' requirements;
- Dependency and influence from outside of the firm has become bigger; and
- Degree of collaboration among existing systems has become bigger, leading to a

spaghetti structure. This also decreases efficiency of maintenance and becomes a barrier to interfacing existing systems, even when newly developed.

Moreover, the differences between the real objects (physical city planning) and the virtual objects (EIS city planning) are attributed to visibility and substantiality. In other words, real things have various physical constraints and virtual things have the issue of invisibility, as discussed in Chapter 3.

Eventually, the main issues lie in the process used to accomplish the target architecture. In that context, this issue does not involve the applicability and effectiveness of the EIS city planning itself, but involves the issues of implementation and realization of the goal. The EIS scenario includes important suggestions because this scenario should cover the process needed to migrate from as-is architecture to to-be—actually, live-to-be—architecture. Moreover to-be architecture itself has changed as time passed. In addition, the issues can be attributed to the architectural issues of each information system that composes the EIS. This is because the architecture of information systems deeply affects the flexibility of EIS. In this sense, it may affect the type of reference model that should be selected for the information system that composes the EIS.

According to Putman (2001), "A reference model provides a general-purpose, formal definition of architecture and design practice. It provides a durable framework for structuring decision-making about an architecture. It guides the choices made in an architecture, which, in turn, guide the choices made in an implementation." This dissertation assumes this term to have the same meaning in the work.

Consequently, the EIS city planning approach should realize the following in order to solve these issues:

- Set an appropriate to-be architecture that accurately reflects the business model of the firm,
- Design architecture that enables iterative development in order to shorten the time necessary for each project, and
- Have a structure and adapts to changes in the business environment flexibly and agilely.

5.2.2 New Approach for City Planning

City planning, as has been discussed, relies on the foresight and leadership of public administration. City planning that is prerequisite to interactive, high-level discussions with participating coalitions is called "post-modern city planning," and can be compared to the more traditional "modern city planning" (Yoshikawa, 2000, 121-127). Table 5.1 shows the characteristics and functions of these two types of city planning.

Table 5.1 Modern City Planning and Post-Modern City Planning

	Modern City Planning (Prior Fixed Planning)	Post-Modern City Planning (Consensus Type Planning)
Characteristics and Prerequisites	- Supply-side perspective - Prerequisite of foresight and leadership of public administration - Tree type and top down planning structures - Budget control and compliance	- Demand-side (city) perspective - Interactive participation and coalition - Leverages mechanism of market economy
Function of City Planning	- Structural cognition as abstracting whole city - Uniform control by laws or regulations - Uniform area designation system for major infrastructure - Urban policies pushed by societal growth and change	- Structural cognition as "rhizome" supported by a network - Plan-do-check cycle (iterative city construction) - Resident participation in planning for infrastructure - Emphasizes assessment in planning of public project and environment
Function of Market Economy	- Regulation and guidance against market failure such as overpopulation and deterioration of city environment - Invoked as part of financial policy to cope with economic fluctuation	- Escalation in competition for market economy - Competition and interpenetration between administration and market economy

Source: Yoshikawa, 2000

Modern city planning functions by thinking of the city as a separate entity that is managed by laws or regulations. On the contrary, post-modern city planning functions by considering the city as a rhizome supported by a network. A rhizome is a horizontal, root-like stem that extends underground and sends out shoots to the surface. "Rhizome" is also a figurative term to describe non-hierarchical networks of all kinds (Deleuze & Guattari, 1984). It is not clear at this moment that this rhizome structure can directly apply to information systems, but this contrast between modern and post-modern city planning suggests the future direction of information systems, specifically EIS.

To realize the similar concept of post-modern city planning, EIS is required to have a structure of distributed autonomous parts harmoniously forming the whole. Thus, an

EIS and each information system that comprises the EIS should have a structure that enables this approach, although this discussion focuses on a finer granularity of EIS.

5.3 Discussion on Enabling New Concept

5.3.1 Essence and Accident

What Brooks (1995) described as "essence" and "accident" in Chapter 4 provides insight into the development process of the OS for System 360, as is applicable for larger granularity systems like EIS. If only essences from the real world can be selected, the core system is built with them. Then an independent front-end system as an accident is built on the core system, which may prove flexible enough to handle environmental change and stable systems. In this sense, it may be the "silver bullet" that Brooks mentioned, although he seemed dubious on this topic. Whether it would be the silver bullet or not, seeking things that confer essence from the real world and trying to build EIS is a worthy endeavor.

5.3.2 Maturity Model for Information Systems

If someone intended to adopt post-modern city planning for an undeveloped area from scratch, it would prove unsuccessful since post-modern city planning requires full use of the infrastructure deployed in modern city planning. For instance, a proper common interface will be required when an independent part tries to connect to another part. Furthermore, incorporation of a common protocol and coherence of data semantics are also needed. Moreover, post-modern city planning requires structural development to equip infrastructure, and this requires moving beyond modern city planning. In other words, the stage of modern city planning is required in order to attain post-modern city planning. In this sense, the relationship between modern city planning and post-modern city planning can be considered as stages of architectural development in a maturing process.

The Capability Maturity Model (CMM) is a famous model. CMM was developed by the Software Engineering Institute (SEI) at Carnegie-Mellon University to describe a framework of five levels of capability for the software development process (Software Engineering Institute, 1994). Each level identifies the pattern that a firm demonstrates in building and implementing a software project. The five product development process levels are:

- Maturity level one: Initial level (ad-hoc, immature). At this level, the organization typically does not provide a stable environment for developing new products.
- Maturity level two: Repeatable level. At this level, policies for managing a development project, and procedures to implement those policies, are established.
- Maturity level three: Defined level. At this level, the standard process for developing new products is documented. These processes are based on integrated product development practices and are integrated into a coherent whole.
- Maturity level four: Managed level. At the managed level, the organization establishes metrics for products and processes and measures results. Projects achieve control over their products and processes by narrowing the variation in their process performance to fall within acceptable boundaries.
- Maturity level five: Optimized level. At this level, the entire organization is focused on continuous process improvement.

As CMM focus on firms' capability to produce software, it does not directly address concerns about the maturity of a firm's architecture.

Ross (2003) reasoned that the maturity of an architecture exists in four stages. Ross's meaning of architecture is considered similar to the EIS architecture that the author has advocated, since she stated the relationship among information systems in the enterprise. She stated, "Firms' common experiences in evolving their IT architectures suggest four distinct stages of increasing enterprise IT architecture competency." According to Ross, the four stages are as follows:

- Stage 1: Application silo architecture. This architecture consists of an architecture of individual applications rather than an architecture for the entire enterprise;
- Stage 2: Standardized technology architecture. The IT architecture becomes enterprise-wide and provides efficiency through technology standardization and, in most cases, centralization;
- Stage 3: Rationalized data architecture. The enterprise-wide IT architecture expands to include standardization of data and processes.
- Stage 4: Modular architecture. The architecture builds on enterprise-wide global standards with loosely coupled applications, data, and technology components to preserve global standards while enabling local variation.

These stages show a stage of development as each stage builds on a preceding stage.

When a Stage 1 firm would like to become a Stage 4 firm, it needs to standardize technology and deploy common information infrastructure as in Stage 2, as well as rationalize data architecture enterprise-wide as in Stage 3. Only after completing these processes can it take on the challenge of building a modular architecture as in Stage 4.

Comparing the two types of city planning with Stage 3 and Stage 4 in Ross's model, a relationship is evident. The target of Ross's Stage 4 is a systems structure where distributed autonomous parts harmoniously forming the whole (EIS), while the post-modern city planning is a rhizome structure based on a flat network. For city planning, it is difficult to reach the stage of post-modern city planning without going through the stage of modern city planning. Post-modern city planning assumes that infrastructure has improved. For the infrastructure improvement, the structured approach is effective because it requires a wide range of scope and planning.

5.3.3 Structure Enabling New Approach

As noted, the essential issue is illustrated in the stage of maturity that each stage requires compared with the preceding stage. Further, an EIS and its components must have a structure for an EIS city planning approach to be carried out. In addition, a structure where distributed autonomous parts harmoniously form the whole (EIS) is necessary for a system to adapt to the pace of change in the business environment. An EIS city planning approach is an effective framework for realizing such a model by synthesizing and modeling each of the relevant viewpoints.

A service-oriented architecture (SOA) has recently become popular. SOA is a concept for building systems composed of applications consisting of combined services. The term service used in the context of SOA describes the functional unit of a system logically. Service is configured by a unit of actual operation process and the system can invoke the service through defined open and standard interfaces. For implementation, SOA is a mechanism to define a service layer that is invoked from outside of the system through an open interface. Each service links the application functions as service components. In other words, SOA is a mechanism that abstracts the application interface by wrapping service layers on the applications (Figure 5.1).

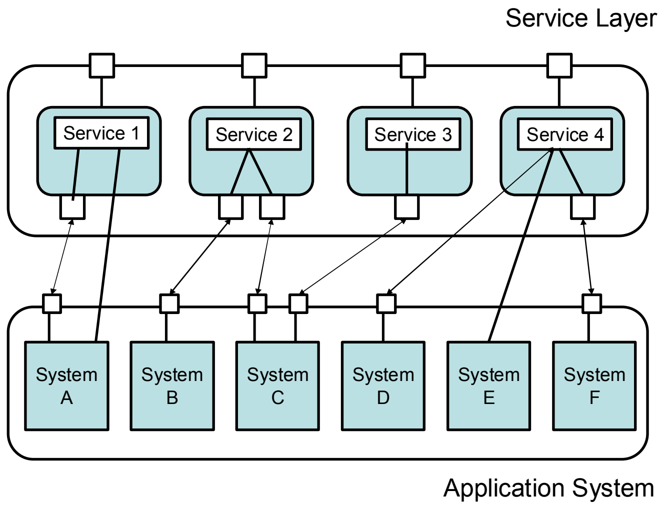


Figure 5.1 Concept of Service

On that basis, application and/or service functions can be configured to select the required functions through the service layer. Moreover, systems maintenance can be done per service unit, as the service itself is the existence of abstraction. In order to leverage SOA for the level of EIS, services should be managed cohesively, or a spaghetti structure of services will be formed. Implementing a repository for service and maintaining could potentially avoid the conflict of semantics. This method, however, is less efficient as the number of services to manage increases. The criteria of service interchange will be required, but with increasing the number of services, this task can be expected to increase exponentially. On that account, building systems based on the conceptual data model—similar to KDDI's—will be effective.

5.4 Insight from KDDI's Case

In KDDI, it is possible to build information systems with a concept almost identical to the EIS city planning approach due to the ability to change or the rust-free information systems based on the concept of object orientation. In order to deploy these structures, subsystems should be defined along with the conceptual data model (ensuring independency of subsystems), and then the rules for implementation (such as the implementation of hubs) and the managing organization should be introduced. With respect to building adaptive information systems in a changing business environment, this case presents the conceptual data model as an invariant element that is designed adequately, and the subsystems are divided and deployed in a manageable size. Furthermore, it is an organized information system that effectively maintains its independence.

KDDI creates the conceptual data model by modeling the real world at the core of its information systems from the described business model, recognizing that information systems simulate the real world. Then the application portfolio, the information systems architecture, and the migration plan are created based on this conceptual data model, and the component partitioning is performed, as mentioned in Chapter 4. Figure 5.2 illustrates the three schema architecture based on ANSI (Shigeno, 2004b). According to Hay (1997), the conceptual schema forms the external view into a single, coherent definition of the enterprise's data. An external schema may consist of a variety of elements. An internal schema is an organization of data, according to the technology being used to record it.

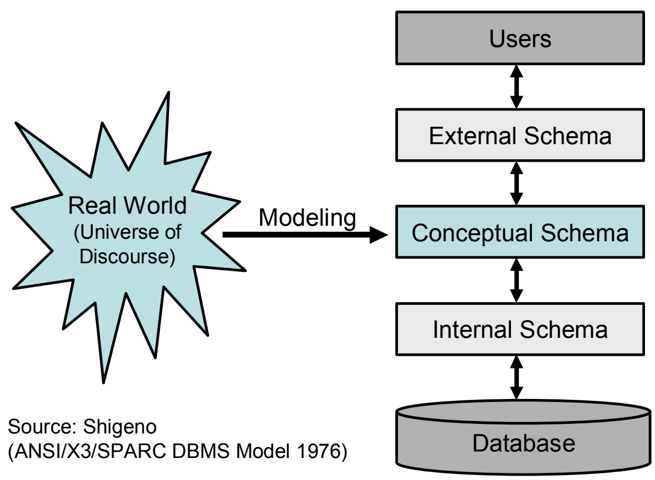
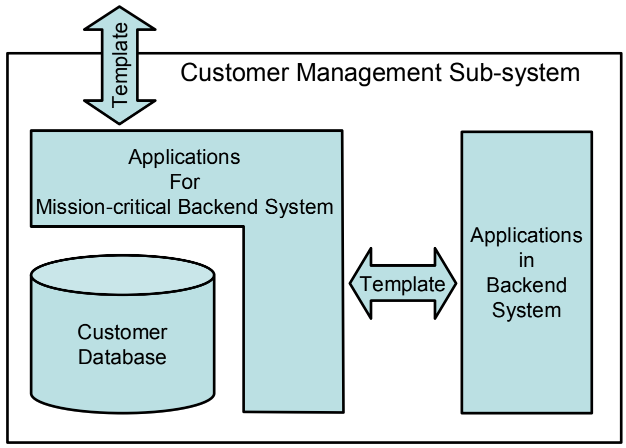


Figure 5.2 The Three Schema Architecture

As mentioned in Chapter 4, KDDI developed the conceptual schema (data model) that models the real world under the concept of object orientation. Then the company built the logical data model and, finally, the physical model. Through these processes, it partitioned the system into sub-systems. Each sub-system, as shown in Figure 5.3, is separate from: (a) the other sub-systems, (b) the applications in backend systems, and (c) the front-end systems by template. This architecture assures that KDDI's sub-systems are independent of each other. The template attempts to prevent a sub-system from developing complex and complicated structure, and that maintains systems flexibility.



Source: Shigeno, 2004

Figure 5.3 Structure of Mission-Critical Sub-System

According to the interview with Shigeno—CIO of KDDI—KDDI does not feel the necessity to change this model, though it has been more than ten years since the first application system based on this idea was implemented. This shows that the “essence” of information systems has not changed with time, if it designed properly.

In this sense, Brooks’ meaning of essence for KDDI systems might be the conceptual data model, where the company maps its business model under the philosophy that the information system should simulate the real world. In the context of Ross’ maturity model, the company built the forth stage model, which goes beyond the third stage model of data and process standardization, which is a prerequisite of the conceptual data model..

5.5 Summary of the Chapter

There are many reasons physical city planning seems unsuccessful. The major issues are that the speed of change and the diversification of residents impede the effectiveness of city planning. Though it has been effective in building urban area and its infrastructure until recently, city planning is criticized because it keeps the stage of modern city planning based on the foresight and leadership of the public administration. Recent business climates that require a speed of change reveal a deficit in city planning rather than strength.

Applying this discussion to EIS city planning, a similar criticism can be seen. This does not originate from the essential issues of the EIS city planning approach, but from the issue of how to realize the live-to-be architecture the EIS scenario targets. And it originates from the issue of architecture for the level of information system that composes EIS. In other words, a finer granularity than the EIS level will be needed to realize this.

The post-modern city planning has proposed to improve these issues. The post-modern city planning presumes modern city planning. Thus, the level of information systems at a lower abstraction level of EIS becomes the issue. From that perspective, the difference between both city planning models may be attributed to the difference between levels in the maturity model. If so, it requires the step (stage) that Ross mentioned for the EIS to migrate to the new concept of city planning. In other words, Ross's Stage 4 would be accomplished through realization of the standardization of technologies (Stage 2) and the data rationalization (Stage 3).

It is predicted SOA, including Web service technology, will be popular in the near future. That is to say that Stage 4 is a completed feature of a modular architecture such as SOA. These situations will force the EIS to select and utilize various information systems services from both inside and outside of the firms. In this situation, firms have to consider: (a) what the core competence of the firm's business is, (b) how to realize the competence in EIS (in other words, in the area of the three viewpoints), and (c) how to select and implement applicable technologies. In particular, it is important that firms define and design boundaries of both business operations and information systems beyond that of the firms as legal entities in order to integrate each information systems service or application, and also to implement proper systems operation management.

From the EIS point of view, the service that is built based on respective requirements from "part" will cause problems. This is because each service based on local requirements is not always consolidated under optimized policy. KDDI built the conceptual data model first as an invalid thing and then built derivative systems under the existence of the conceptual data model. This approach might be a direction to solve the complexity of the systems. This gives EIS a robust foundation to migrate to the stage where the concept, corresponding to the post-modern city planning, will be workable.

References

Brooks, F.P., *The mythical man-month: Essays on software engineering, Anniversary edition*, Addison-Wesley, 1995.

Deleuze, G. and Guattari, Massumi, B. (Translation). *F. A thousand Plateaus*, University of Minnesota Press, 1987.

Hay, D. "From Relational to a Multi-Dimensional Data Base," 1997. www.essentialstrategies.com/documents/relmulti95.pdf

Putman, J., *Architecting with RM-ODP*, Prentice Hall, 2001.

Ross, J. W., "Creating a Strategic IT Architecture Competency: Learning in Stage," *CISR Working Paper* No.334, April 2003.

Software Engineering Institute (SEI), *The Capability Maturity Model: Guidelines for Improving the Software Process,* Addison Wesley, Massachusetts, 1994.

Yoshikawa, T., *Evaluation of City Planning,* in Minohara, K. (Edit), *Challenge of City Planning,* Gakugei-Shuppannsha, Tokyo, 2000. (in Japanese)

Chapter 6

Conclusion

6.1 Summary of the Study

The purpose of this study is to present a comprehensive IT architecture framework for firms that intend to rebuild their EIS to adapt to the speed of change in the business environment. Most firms have an EIS consisting of heterogeneous and disparate information systems, and may be complex and complicated structurally. Despite these situations, business requires collaborative work with partner companies or direct sales to customers. These accelerate inflexibility, difficulty in maintenance, arduous operation, and high cost of maintenance and enhancement of EIS. Thus, firms need a workable guide to rebuild their EIS in practice.

In Chapter 2, earlier studies concerning this subject were introduced. Zachman's was the first study to grasp large-scale systems in a logical structure (architecture). After Zachman, the RM-ODP model, the enterprise architecture, and the city planning approach followed. The city planning concept uses a metaphor that likens an enterprise information system to a city. In Chapter 2, some criticisms of this approach were explored. One such criticism regards the effectiveness of the metaphor; another questions the scope of the framework. Regarding the former, the limits of the metaphor have to be conceded, although using it to explain an abstract object to a layperson has been effective. In response to the latter criticism, the framework of IT architecture needs to be presented. In this discussion, the requirements for a comprehensive framework for EIS were suggested.

In Chapter 3, the EIS city planning approach, which consists of EIS architecture and EIS scenario, was proposed. In order to describe EIS architecture, IEEE's model for describing architecture is used. As EIS architecture selects viewpoint(s) according to the model, selecting a viewpoint is essential in describing architecture. The most notable aspect of this framework is using the methodology and concept of physical city planning. This applies to selecting viewpoint to describe EIS architecture. When planning an EIS scenario, the methodology and the know-how from physical city planning are also exploited.

In Chapter 4, the cases of KDDI, Ebara, and Monex were discussed to evaluate the EIS city planning approach framework. The framework and structure of the case were compared, and similarities were identified. Each case showed some common concepts with the framework. Specifically, analysis with three architectural viewpoints is effective in revealing architectural characteristics. In addition, the discussion from the perspective of the EIS scenario is effective to illustrate an approach for each company.

In Chapter 5, further discussion was introduced for a more effective EIS city planning approach in practice. The issues were discussed based on the conclusion that the current city planning approach seems to be less effective than desired. These issues are mainly due to the problem of level of architectural maturity. The post-modern city planning approach was proposed to resolve the current issue. Since the idea of the approach for the information systems world seems to be a system of distributed objects, the executable structure of information systems that compose EIS is required to realize this architecture. KDDI's case uses much of this structure; its approach to rebuilding EIS will perhaps represent one direction for resolving this issue.

6.2 Significance and Implication of EIS City Planning Approach

The author proposed three viewpoints drawing upon a physical city planning approach to describe the EIS architecture that composes the EIS city planning approach. When selecting viewpoints, the analogy from physical city planning was used. The structure viewpoint can be applied to design structured (layered) architecture. This viewpoint corresponds to the layered structure of city, infrastructure, urban design, and social environment. The part and whole viewpoint can be applied to accommodate a conflict of benefit and/or deficit between the part (e.g., strategic business unit, division) and the whole (i.e. headquarters) in the firm. This viewpoint is derived from a function of physical city planning, a conflict of benefit as in a NIMBY facility, or development in a specific area. The ins and outs viewpoint can be applied to mitigate the issue of boundaries over which the business of enterprise expands. This viewpoint corresponds to the accommodation between the city planning and the plan for upper-level or neighboring areas.

The author also proposed the EIS scenario to accomplish the migration plan from the program management point of view in order to accomplish a live-to-be architecture described by the viewpoint. The term "program management" comes from the program and project management (P2M). It manages an aggregate of projects concurrently at the level of the whole firm, rather than a project management that targets each independent project. This EIS scenario exploits the methodology or the experience of physical city planning, such as a zoning, a development plan.

Information systems are conceptual objects and have an invisible existence. An architecture for information systems which is abstracted from information systems as a design concept is even more invisible for the lay person. Furthermore, it is far more difficult for management or an involved party to understand EIS architecture, which is abstracted from the information systems at the level of enterprise. Needless to say, exploiting such a metaphor at a time like this is quite effective in helping them understand viscerally, because a metaphor can use visible or well-known objects to substitute for invisible objects. In this sense, metaphor has its place. Since metaphor has a limit in its applicability, it sometimes may provide an ineffective image or cause misunderstanding if it is used beyond its limits.

Besides, city planning that includes only the technologist's perspective of the relationship between IT and business processes would be trivialized in the EIS architecture. To do so misses the key function of the EIS as an enabler for business, which essentially should be potentially incorporated in the architecture. Designing architecture, planning, and executing a workable migration plan to accomplish the architecture, along with the framework of EIS city planning approach that is proposed in this dissertation, are important in this respect.

The author applied the case method to evaluate the framework. Although the selection of cases may seem a controversial issue, a reasonable rule to follow in the selection of cases was decided upon. This rule includes representation and generality, concreteness and openness, and relevance. KDDI's methodology is a case that demonstrates how to build information systems, and includes a lot of helpful know-how. Ebara's shared service for server consolidation offers an example of virtualization of information infrastructure. Monex's case suggests and illustrates systems collaboration in the future, though it is far from a sophisticated form. For further discussion of the applicability and effectiveness of the framework, it should be used as the methodology to accumulate various cases and analyze them comparatively.

Since the further issue of applying an EIS city planning approach was discussed in Chapter 5, the approach of the post-modern city planning suggests a direction for building information systems in the future. As Ross pointed out in the maturity model of IT architecture, the degree of maturity of the information systems of the firm itself becomes a subject of discussion when a firm executes an approach.

Post-modern city planning suggests a direction to resolve the issues currently facing city planning. When this concept is applied to EIS city planning, each information system that composes an EIS must have a structure, though this discussion is focused on a finer grained object than EIS. This indicates that the information systems themselves become the fourth step of Ross' maturity model, the structure of which includes distributed autonomous parts harmoniously forming the whole. As EIS has matured, a traditional planning-oriented method is becoming harder to apply. Instead of the traditional method, the similar approach of the post-modern city planning would be one alternative. Before applying this approach, however, there are several problems to be solved. KDDI's philosophy of object orientation to assume information systems as the simulator of the real world would be one possibility. On that basis, the structure of information systems in which the object corresponding to Brooks' essence is structurized, and the object corresponding to the accident is implemented with an adaptable environment under the existence of essences, would be effective.

6.3 Further Studies

Future work on this topic could include applying this framework to assorted cases that have a major influence on entire systems in order to examine further applicability and effectiveness. These cases include EIS integration caused by M&A, ERP implementation as a tool to rebuild EIS, and EIS integration with e-business systems. Through such applications, the author could evaluate the benefit and significance of the framework and enhance the applicability and effectiveness of the framework. The author also hopes to provide reference models for the framework that will further help to visualize and substantiate the EIS city planning approach. Moreover, the author hopes to propose a structure and an organization for EIS governance that can cover the lifecycle as zoning and that can execute an EIS scenario.

It might be possible to combine a maturity model of architecture into the EIS scenario if the reference models of the modern and post-modern city plan could be compared with those of the maturity model of architecture for EIS. On that basis, the author would like to reorganize the EIS city planning approach on the model of information systems that have an adaptive structure to change.

Appendix:

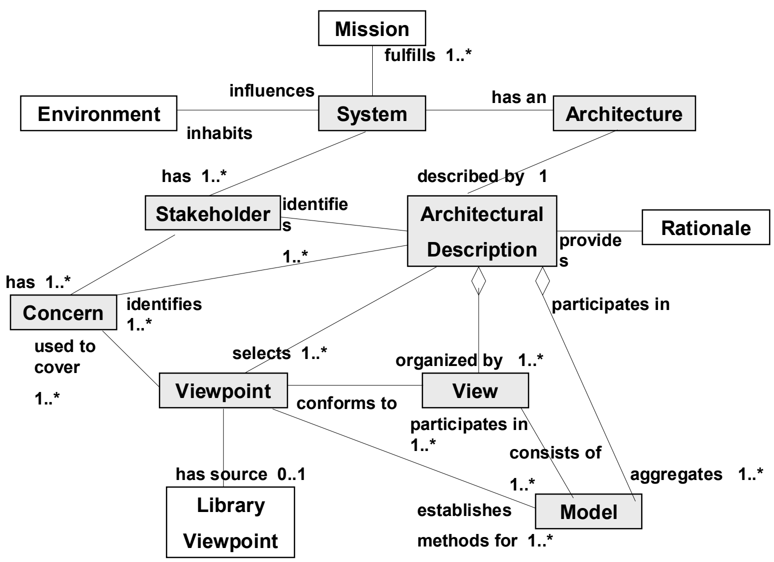
Glossary and Acronyms

architecture

- The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. (IEEE Std., 2000)
- the structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. (CIO Council, 2001)

architectural description

- A collection of products to document an architecture. (IEEE Std., 2000)



Source: IEEE Std. 1471-2000

Figure A1.1 Architectural Description

as-is architecture

- **current architecture** Defines the "as is" enterprise architecture and consists of two parts: current business and design architectures (i.e., data, applications, and technology). This is a representation of current capabilities and technologies and is expanded as additional segments are defined. (CIO Council, 1999)
- **baseline architecture** The set of products that portray the existing enterprise, the current business practices, and technical infrastructure. Commonly referred to as the as-is architecture. (CIO Council, 2001)

Capability Maturity Model (CMM)

http://www.npd-solutions.com/cmm.html

- The Capability Maturity Model was developed by the Software Engineering Institute at Carnegie-Mellon University to describe a framework of five stages of evolution or levels of capability or process maturity. The CMM describes an evolutionary improvement path from an ad-hoc, immature process to a mature, disciplined process. This model applies to new product development as well as software development.
- The five product development process CMM levels are:

- **initial level** (ad-hoc, immature): At the initial level, the organization typically does not provide a stable environment for developing new products.

- repeatable level: At the repeatable level, policies for managing a development project and procedures to implement those policies are established.

- defined level: At the defined level, the standard process for developing new products is documented, these processes are based on integrated product development practices, and these processes are integrated into a coherent whole.

- **managed level:** At the managed level, the organization establishes metrics for products and processes and measures results.

- **optimized level:** At the optimized level, the entire organization is focused on continuous process improvement.

CMM ➔ Capability Maturity Model

concern

- Each stakeholder typically has interests in, or concerns relative to, the system. Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders. Concerns include system considerations such as performance, reliability, security, distribution, and evolvability. (IEEE Std., 2000)

CRM ➔ customer relationship management

customer relationship management

- Technique to establish long-range relationships with a customer by utilizing information systems. Usually based on a detailed customer database, all relationships including transaction to each customer are controlled.

distributed processing system

- A distributed processing system is a system composed of separate processing systems that inter-work for the purpose of sharing information and processing. (Putman, 2001)

enterprise

- The highest level of business domain governed by unified policy. When the lower business domains are highly independent and the business relationship to upper level is weak, the independent business unit can be also defined as, substantively, an enterprise. Enterprise includes not only business enterprise but also non-business enterprise, such as governmental ones.
- An organization (or cross-organizational entity) supporting a defined business scope and mission. An enterprise includes interdependent resources (people, organizations, and technology) that must coordinate functions and share information in support of a common mission (or set of related missions). (CIO Council, 1999)

enterprise architecture

- A strategic information asset base that defines the mission, the information necessary to perform the mission, the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs. An enterprise architecture includes a baseline architecture, target architecture, and a sequencing plan. (CIO Council, 2001)

enterprise resource planning (ERP)

- A technique or concept with which to manage a whole enterprise that is integrated from the perspective of effective utilization of management resources..
- An ERP is an integrated cross-business software package to realize the concept.

ERP ➔ enterprise resource planning

framework (of an architecture)

- A structure for investigating an architectural problem or building a specific approach to architectural concerns that are related to each other through the rules of the structure. As per [RM-ODP-2], support to create an architecture specification addressing integration, portability, inter-working (interoperability), and of course distribution. (Putman、2001)

- A logical structure for classifying and organizing complex information. (A practical Guide)
- The framework provides a sustainable mechanism for identifying, developing, and documenting an architecture description of high priority areas built on common business areas and designs that cross organizational boundaries. (FEAF)

model

- Represents the system from a set of concerns or foci on the system. A model can be a document, a representation in a visual tool, or some other artifact. A system, then, is typically represented by a set of models, each addressing some particular area of concern. (Putman、2001)

open system

- An open system is a system that provides portability and interoperability, whether or not the software entities are on the same computer or reside on different computers. Portability allows the software entity to operate on different computer systems. Interoperability allows the software entities to exchange information in a way that can be "understand." (Putman, 2001)

reference model

- A reference model provides a general-purpose, formal definition of architecture and design practice. It provides a durable framework to structure decision making about an architecture. It guides the choices made in an architecture, which, in turn, guide the choices made in an implementation, (Putman, 2001)

Reference Model of Open Distributed Processing (RM-ODP)

- Standard for modeling object-based distributed processing architectures that separates concerns and simplifies the specification of heterogeneous, open distributed processing system. (Putman, 2001)

RM-ODP ➔ Reference Model of Open Distributed Processing

spaghetti

- Since business requirements have always been urgent and mandatory, some firms' systems have been maintained, enhanced, and integrated with ad hoc and makeshift solutions without a long-range view. Accordingly, they are burdened with increasing systems complexity, resulting in high maintenance costs and untenable structures. This tangled aspect of information systems is described as "spaghetti," a term initially used for a program with a confusing control structure and, in turn, for a state of uncontrollable applications. To make things worse, now the term is used to describe systems beyond the single-application level. IBM (1999) stated, "The problem gets worse as you try to connect more and more applications . . . This type of solution has built-in inflexibility and a high cost of maintenance and we characterize it as inter-application spaghetti."

stakeholder

- An individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system. (IEEE Std., 2000)

system

- A collection of components organized to accomplish a specific function or set of functions. (IEEE Std., 2000)

target architecture

- The set of products that portray the future or end-state enterprise, generally captured in the organization's strategic thinking and plans. Commonly referred to as "to-be architecture." (CIO Council, 2001)

to-be architecture

- Defines the to-be-built enterprise architecture and consists of two parts: target business and design architectures (i.e., data, applications, and technology). This represents the future capabilities and technologies resulting from design enhancements to support changing business needs. (CIO Council, 2001)

transitional processes

- Support the migration from the current to the target architecture. Critical transition processes for the federal enterprise include capital IT investment planning, migration planning, configuration management, and engineering change control.

view

- A representation of a whole system from the perspective of a related set of concerns. (IEEE Std., 2000)

viewpoint

- A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.