Reconstruction of Large Systems Based on Business Data Model

● Shoichi Okugawa

(Manuscript received March 28, 2006)

Some of our customers have large mainframe systems that have become oversized and hard to maintain, and these customers now want to structurally reform these systems. However, it has been difficult for them to learn how to make this reform. This situation prompted Fujitsu to develop the FUTURITY solution, which is based on the experience we gained in the structural reform of KDDI Corp.'s system and involves work processes, development methodologies, and tools. FUTURITY is composed of four solutions: FUTURITY-TB, which is used to abstract a customer's business into a business data model and develop To-Be data models; FUTURITY-AI, which is used to abstract the data structure of a customer's current system and develop As-Is data models; FUTURITY-MP, which is used to compare To-Be data models with As-Is data models and develop reconstruction plans; and FUTURITY-SI, which provides methods and tools for efficiently developing a system designed on the basis of business models. This paper describes the basic system development philosophy and other details of FUTURITY.

1. Introduction

Today, many companies that maintain large-scale mainframe systems are worried about how they can reform the structure of their systems, which are growing larger and more obsolete year-by-year. One solution is to reconstruct the old, outdated systems; however, it is not easy to reconstruct systems consisting of several million steps. Moreover, reconstruction does not always solve the problem of maintainability.[1),2)]

In May 2004, Fujitsu helped KDDI Corporation[3)] (hereafter called KDDI) totally reconstruct its large-scale, mainframe fixed-network billing system. Then, we integrated the approach, ideas, and development methods we used for this reconstruction and the experience we gained to create the FUTURITY solution. The concept of FUTURITY was also applied to SDAS/Service Modeling. (KDDI is the only Japanese information and communications enterprise that provides a comprehensive range of services for fixed-line and mobile communications.)

This paper describes a conceptual approach to constructing flexible, highly maintainable systems and the challenges faced when developing such systems. It then describes the solution system of FUTURITY.

2. Highly maintainable systems

This section explains the essence of highly maintainable systems.

2.1 What companies require from information systems

Maintainability is always a significant factor in constructing strategic information systems that directly affect business. Because these systems are now indispensable for business operations, the rate at which services are provided has become proportional to the rate of system development.

The cellular phone business is a clear example of this proportionality. When KDDI began providing its family discount service, which discounts basic monthly charges and call charges for calls between family members, other companies soon offered similar services. If KDDI had taken six months or more to develop a system for the new service, their business would have suffered significant damage. Competition among companies will become even keener when services such as number portability allow customers to change cellular phone companies without changing their phone numbers.

The primary goal for companies in constructing (or reconstructing) information systems is to survive business competition by using their own highly maintainable systems. System vendors know it is a big challenge to construct highly maintainable systems that will last for many years.

2.2 Constructing a highly maintainable system

How can we develop highly maintainable systems? First, we must visualize the whole system. However, because it is difficult to grasp a system consisting of several million steps as a whole, developers can only understand the parts they are assigned to work on. The old saying, "You can't see the forest for the trees," also applies to system development. It is important to see "the forest" of a large system. That is, we must create a conceptual representation of the whole system and make it a communication base for stakeholders.

Therefore, in our business information systems solutions, we have introduced business data models, which are conceptual structures defined as a model and an architecture, by abstracting business of the real world. For example, we have created a business data model for the KDDI project mentioned above. As F. P. Brooks, Jr. writes in "The Mythical Man-Month,"[4] a business data model abstracted from business of the real world enables the construction of a system that can flexibly adapt to changes in the real world.

Business data models are essential in business software development.

2.3 Challenges when developing flexible systems and their solutions

It is difficult, however, for developers who have been used to function-oriented development (i.e., development to realize required functions) to understand how to develop a system based on a business data model.

There are two obstacles to overcome before a developer can master this approach to system development. One is the abstraction of real-world business into a model. For developers who have been engaged in materializing, which is the opposite of abstracting, it is not easy to extract kaname entities/events from the real world and create a model. The other obstacle is the exact implementation of the model onto computers. It is good to have a model, but a developer might not know how to implement it.

Fujitsu has overcome these two obstacles through its experiences in developing small systems. From these experiences, we have gradually learned the advantages of a development approach based on business data models, and we have now mastered this approach.

First, we brainstorm on the business before we create the business data model. This gives us a close understanding of the business's essence when creating the business data model. Then, by implementing abstract models that reflect the real world using object-oriented technologies, we can easily see which software modules will be affected by real-world changes. The important points in system development are to design a firm architecture, understand the architecture fully, and then develop the system based on the model.

Although it can be difficult for developers to learn this method at the beginning, they soon see that it provides better maintainability. In fact, KDDI's reconstructed system is rated as being 3.5

times more maintainable than its predecessor. Therefore, thanks to this method, an additional development of this system that would have taken, for example, seven months, will now take only two months.

3. Solution system of FUTURITY

FUTURITY is based on a development approach oriented for the above-mentioned business data models. Therefore, it can be also applied to an ordinary development started from scratch. However, we are preparing it as a solution for our main challenge at this time; namely, the reconstruction of large-scale mainframe systems. This section describes the solution system of FUTURITY (**Figure 1**).

1) FUTURITY-TB (To-Be)

Create a To-Be business data model by abstracting the business of the real world. The model can incorporate business factors expected in the near future, but not uncertain factors.

2) FUTURITY-AI (As-Is)

Create a business data model by collecting information from the database, files, and screens of the existing system in a bottom-up manner. This model is used to clarify the problems of the existing data structure by comparing it with the model created in the TB process.

3) FUTURITY-MP (Migration Plan)

Create a migration plan by comparing the outputs from TB and AI and by considering future business factors and development risks.

4) FUTURITY-SI (System Integration)

Construct a system based on the results from TB, AI, and MP. Make the system flexibly adaptable to changes by applying a system development standard that is oriented for business data models and development efficiency improvement tools.

3.1 FUTURITY-TB

FUTURITY-TB corresponds exactly to what Brooks described as “the essential work” in “The Mythical Man-Month.” We create the conceptual structure (business data model) in a top-down manner by finding the “kaname entities” of business in the real world and the “kaname events” that make these entities undergo state transitions.

It is important here to simply abstract the business of the real world without being bound by the mechanisms of existing systems or implementation methods. Therefore, we need to work with people who are familiar with the business being modeled rather than people familiar with the system used to conduct that business. We create three business data models:

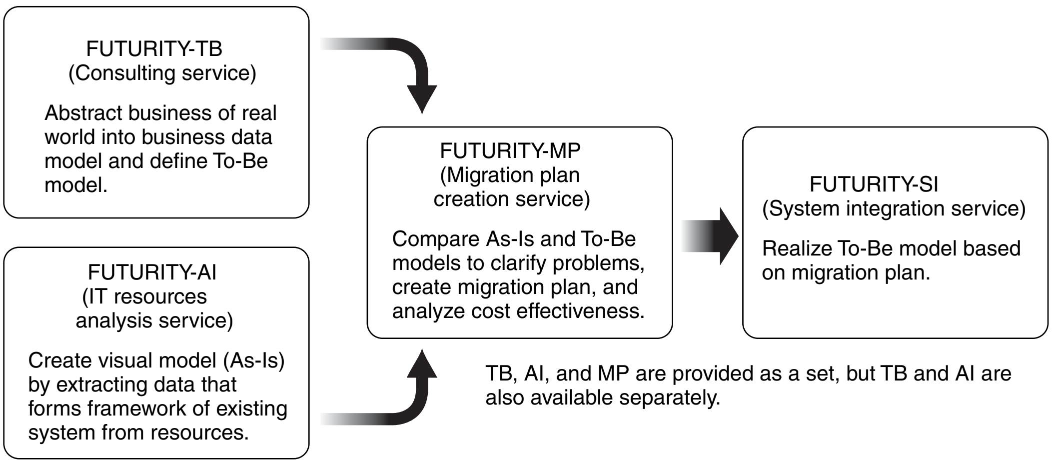

Figure 1
Solution system of FUTURITY.

1) Static model

This model represents the relationships between kaname entities and kaname events using an entity-relationship (ER) diagram.

2) Dynamic model

This model abstracts business events that affect the entities represented by the static model and shows their lifecycles in a state transition diagram.

3) Interaction model

This model represents the relationships among the business events, the business entities, and the sections related to the business events and entities.

It is especially important to recognize that the static model represents the business of the real world, not the implemented database. (While the purpose of the bottom-up approach of Data-Oriented Approach [DOA] is to design databases, the purpose of the business data model is to abstract the business of the real world.) Fujitsu has FUTURITY-trained consultants who can create highly accurate models of the real world.

3.2 FUTURITY-AI

FUTURITY-AI visualizes an existing system. Visualization of existing systems tends to end in failure because analyzing a volume of resources in detail in a bottom-up manner requires a lot of work.

Therefore, FUTURITY-AI focuses on data. It categorizes an existing system's database into entities, relationships, events, rules, information, and mechanisms and extracts only the entities and relationships to create the static model. The dynamic model is created by extracting business events from screens that accept updating and external interfaces. The data structure must be understood from a broad point of view in order to compare the existing system with the output from TB at the same grain size. Comparison at this level often clearly reveals problems in the data structure of the large-scale mainframe systems in the communications carrier, electric power, and gas industries, which have been maintained for nearly 20 years.

3.3 FUTURITY-MP

FUTURITY-MP compares the output from TB with that from AI to create a migration plan. The fundamental concept of FUTURITY-MP is to divide the existing system into loosely coupled subsystems that are not significantly affected by changes in another subsystem and reconstruct the system step-by-step, beginning with the subsystem that benefits the most from the effects of reconstruction.

Consideration must be given to two factors: risk and cost effectiveness. A system consisting of several million steps is usually oversized and entangled like spaghetti, with closely related subsystems and with similar logic scattered here and there. Because it is risky to reconstruct such a system all at once, structural reform of the entire system must be performed by gradually disentangling the "spaghetti." Reconstruction is not necessary for portions that do not require structural reform (i.e., those that will not be greatly modified).

In addition to creating a migration plan, we also offer in cooperation with our consulting section, a service that calculates cost effectiveness.

3.4 FUTURITY-SI

Based on the results from MP, FUTURITY-SI realizes the To-Be system defined in TB. This work implements the business data model as an information system, and in a sense, this is the most difficult of all the processes. In this step, we provide a system development standard, standard architecture, and efficiency improvement tools, including Model-Driven Architecture (MDA), APDC Repository Tool (ART), Service-Oriented Architecture (SOA), and APDC Front Repository Organizer (AFRO). These efficiency improvement tools comply with the system development standard and require the standard architecture.

Although this system integration work

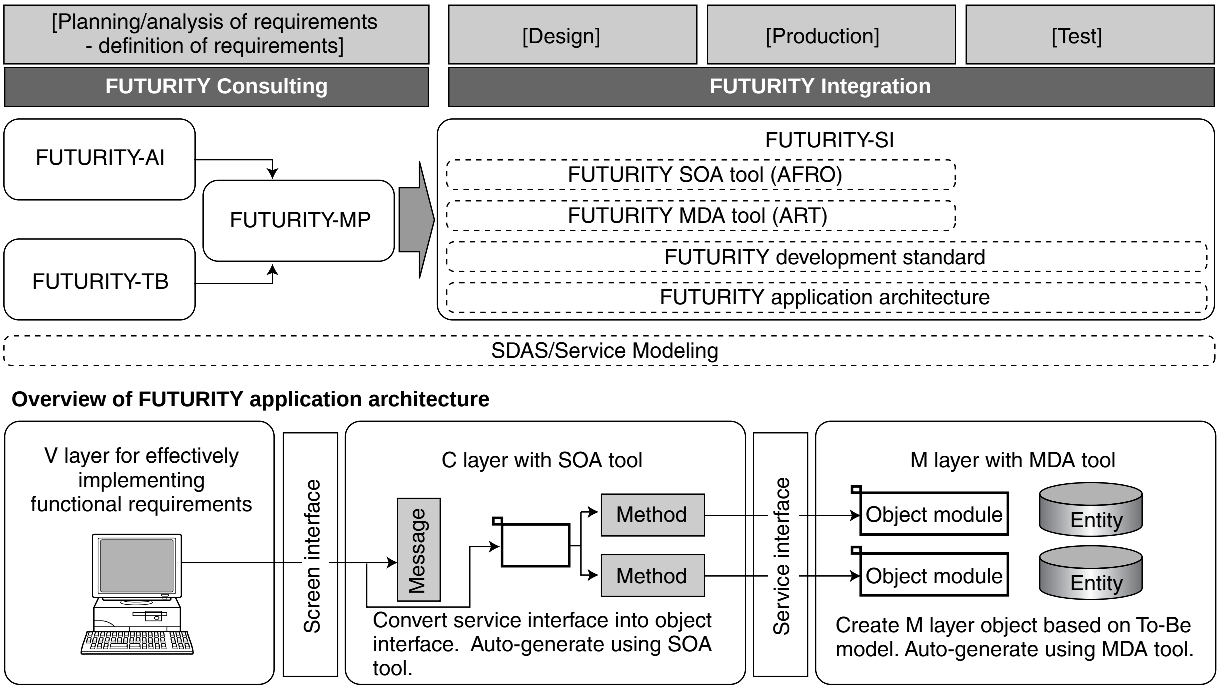

Figure 2
Solution structure of FUTURITY-SI.

basically varies according to the customer's specific needs, the application of the prepared FUTURITY-SI solution helps improve development efficiency (**Figure 2**).

5. Conclusion

The FUTURITY solution was developed based on the experience we gained from reconstructing KDDI's fixed-network billing system. Now, we intend to apply FUTURITY to the electric power industry, which requires a total overhaul of its out-of-date systems.

In the future, we believe that the information system industry will need to adopt the concept of FUTURITY to realize simple, highly maintainable systems and standardized non-person-dependent system development. We also believe this concept will be needed to make a paradigm shift from the traditional business model of mass production to a new business model in which the value-added work of system engineers will be the key to successful competition.

References

1) A. Teshima et al.: Paradigm Shift of Information Systems. (in Japanese), Ohmsha, 1996.
2) T. Shigeno: Make an easy system for additional development and maintenance. (in Japanese), *Nikkei Computer*, 551, p.188-198 (2002).
3) KDDI Corporation. *http://www.kddi.com/english/index.html*
4) F. P. Brooks, Jr.: The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley Professional, 1975.

Shoichi Okugawa, *Fujitsu Ltd.*
Mr. Okugawa received the B.E. degree in Electric Engineering from Doshisha University, Kyoto, Japan in 1985. He joined Fujitsu Ltd. Osaka, Japan in 1985, where he was engaged in development of computer systems dedicated for securities firms and common carriers. He is currently at Telecom, Utility & Media Industries Business Group and in charge of enhancing standard methods of system and human resource development.