Fujitsu's System Development Methodology: SDAS

● Takeshi Oshima ● Masayuki Kashiwagi ● Hiroshi Fukao

(Manuscript received March 23, 2006)

Recent information systems have become critical infrastructures for both industry and society, and there are growing requirements for higher quality and reliability and a shorter development period to meet ongoing changes in the business environment. To meet these needs, Fujitsu has strengthened and enhanced its system development methodology called System Development Architecture & Support facilities (SDAS). Also, Fujitsu has adapted SDAS to new technologies and expanded its scope to cover the complete lifecycle of systems in Japan, from analysis and development to maintenance and rebuilding of existing customer systems. SDAS provides a development methodology, framework, and tools for improving quality and reducing the time needed for development. SDAS features an accurate definition of requirements, a high degree of expandability and maintainability by using a mission-critical application framework, and effective development tools to automate testing and documentation. This paper outlines the features of SDAS methodology, its framework, and development tools.

1. Introduction

To cope with today's rapidly changing business environment, information systems have been used as critical infrastructures for industry and society. These systems are directly related to corporate management and social systems. For example, they are used for the creation of new business using the Internet, cross-enterprise commerce, digitization of public services, and adaptation to the ubiquitous society. Information systems therefore need to meet the high-level requirements for improvement of quality and reliability, short-term development, and application of the latest technologies.

System Development Architecture & Support facilities (SDAS)[1)] is a comprehensive system development methodology that combines the considerable system development know-how that Fujitsu has cultivated over time. The first version of SDAS was released in the 1980s to support application development on mainframes using COBOL, and its functionality was then enhanced in the 1990s to support client-server systems. To meet the above requirements, we have renewed SDAS to further improve quality and reduce the development period corresponding to cutting-edge technologies in the Web system age and extending its scope to cover the entire system lifecycle, from requirement analysis to building, maintenance, and rebuilding.

This paper outlines the features of SDAS, including its methodology, framework, and development tools.

2. Structure of SDAS and goals

This section introduces the latest structure of SDAS, along with its technologies that have been strengthened for quality improvement and short-term development to respond to the Web system age.

2.1 Structure of SDAS

As shown in **Figure 1**, SDAS is composed of technologies for the following: enterprise system planning, support of application development (including requirements, design, implementation, and testing), development processes and project management, and maintenance.

2.2 Goals of SDAS

The goals of SDAS are to improve the quality of system development and shorten the development period by strengthening the following technologies.

1) Accurate definition of requirements and information sharing with customers

Recently, when developing new information systems that will become the infrastructure of a new business area or management innovation, we often encounter ambiguous requirements and specifications because even the customers do not know exactly what they should be. These ambiguities can cause a change in design, which leads to delayed delivery dates. The risk is also great for customers whose systems might not be commissioned by the scheduled delivery date. A key to successful project development is therefore to identify customers' requirements accurately and completely as soon as possible. In these cases, SDAS provides a modeling technique to reduce misunderstanding and misinterpretation of specifications. In addition, by providing tools for describing workflows and screen transitions, SDAS enables document description levels to be standardized and accuracy to be improved when describing business operations.

2) Tools for improving quality and efficiency

Fujitsu has been providing development tools and testing tools that support testing and documentation since the mainframes era. Recently, development tools for Web systems and Java have also been provided. Interstage Apworks (hereafter called Apworks) is an integrated development environment that generates basic source code based on Java 2 Enterprise Edition (J2EE). The SIMPLIA series is a group of testing and maintenance tools. This series, for example, includes a testing tool that automates repeated tests to reduce cost and the time needed for testing. PROSPECS is a documentation support tool that generates maintenance documents from Java source code. These tools have been designed to improve quality and efficiency in development

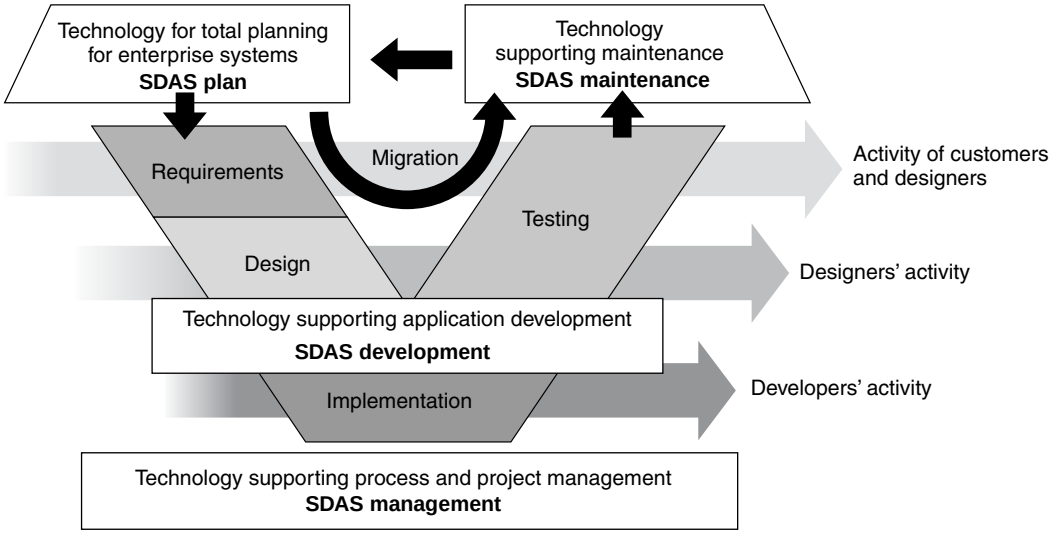

Figure 1
Structure of SDAS.

processes by automating tasks as much as possible.

3) Achievement of efficiency in application development and improvement of reliability and expandability using application frameworks

System expansions and modifications of application structures specific to a particular developer or project require a tremendous number of person hours to investigate, implement, and test because they can have a very wide impact on a large amount of source code. Maintainability can be improved by using application frameworks to standardize application structures and localize the impact areas. Fujitsu provides several application framework products. For example, we provide Interstage Application Server, which is a J2EE-based application server that includes a Web Front Framework; Client J Framework, which is a framework for Java applets; and GKit-Taglib, which is a screen component for Servlet/JSP that enables the development of high-quality screen applications by providing rich functionality for user-friendly interfaces.

3. Fujitsu's approach to EA

To improve efficiency, information systems have been developed on the basis of business and organization lines. As a result, they are inevitably affected by problems posed by the duplication of investment, rise in operation and maintenance costs, and difficulty in making alterations and connections between different systems when new functionality is added. They also face challenges such as misalignments with corporate strategies.

Enterprise Architecture (EA) is a structure for improving business operations and systems from the viewpoint of enterprise-wide optimization rather than conventional local optimization to quickly respond to changes in the social environment and changes in information technologies. EA organizes and systematizes business processes, information system structures, and technologies to be used by visualizing the overall structure of an organization.

Fujitsu provides a method for quickly building EA by applying the best practices obtained through its rich experience in the EA approach. This method is characterized by a seamless relationship between EA analysis and system development using SDAS and TRIOLE[2] (**Figure 2**).

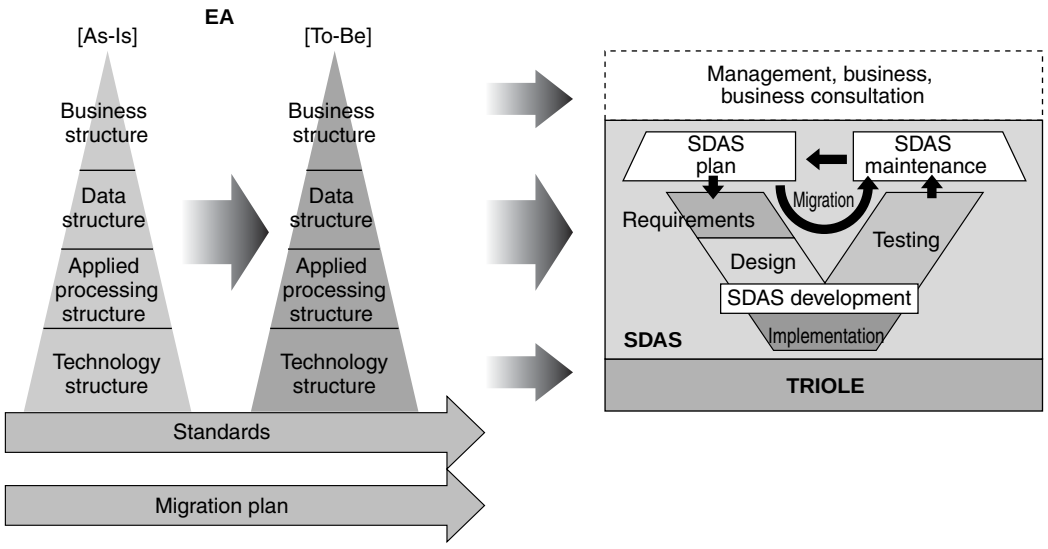

Figure 2
Fujitsu's EA and SDAS.

4. Development standards

This section describes Fujitsu's development standards.

4.1 SDEM21

Fujitsu has established a system development standard called SDEM (Solution-oriented system Development Engineering Methodology) to standardize the content and terminology of system development and thereby promote consistent communications and understanding between customers and development teams. SDEM hierarchically breaks down the development phase and task category, and it can be easily customized to suit individual projects. Fujitsu first provided SDEM and then updated it to SDEM90 and SDEM21 to meet the needs of the latest development techniques and international standards.

4.2 ComponentAA Development Method

ComponentAA Development Method is a practical development standard that provides a design method and documentation to implement applications based on the J2EE architecture. The concepts of object-oriented and component-oriented programming[3)] are incorporated into this standard, so applications can be designed to have high expandability and re-usability.

5. Development technologies

This section introduces the development technologies of SDAS.

5.1 Accurate definition of requirements and information sharing with customers

Fujitsu arranges and provides modeling techniques in the analysis and design phases that help reduce the possibility of misunderstanding customers' requirements.

1) Process modeling technique

This technique makes it possible to eliminate misinterpretations between customers and the development staff by accurately visualizing a total business flow that combines the system and business operations, while eliminating ambiguities in the requirements. (**Figure 3**)

2) Data modeling technique

Fujitsu has played a leading role in providing data-oriented business analysis techniques and design techniques. Using data modeling clarifies the data structure that constitutes the core of business, reduces omissions and duplications, and enables the design of systems that can be easily modified for future expansions (**Figure 4**).

3) SOA development technique (SDAS/Service Modeling technique)

In July 2005, Fujitsu announced a development technology for Service-Oriented Architecture (SOA) that can make IT systems robust against changes.

Fujitsu provides a modeling technique called SDAS/Service Modeling that enables SOA to be realized through the analysis and design of service structures.

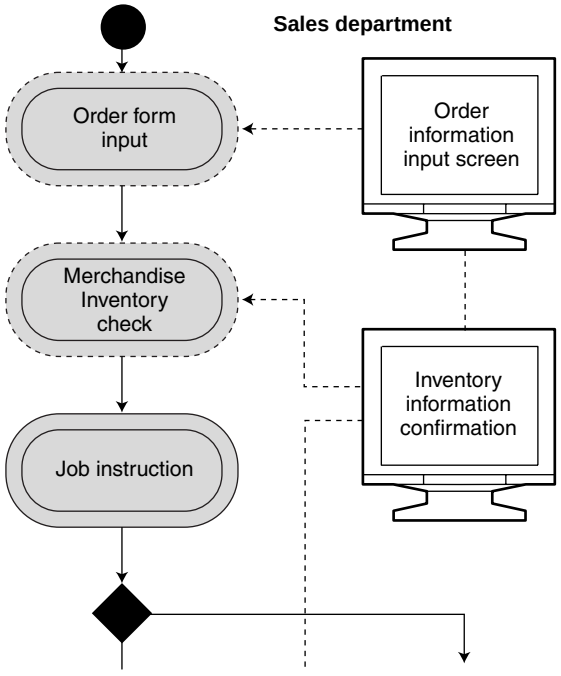

Figure 3
Example of process modeling.

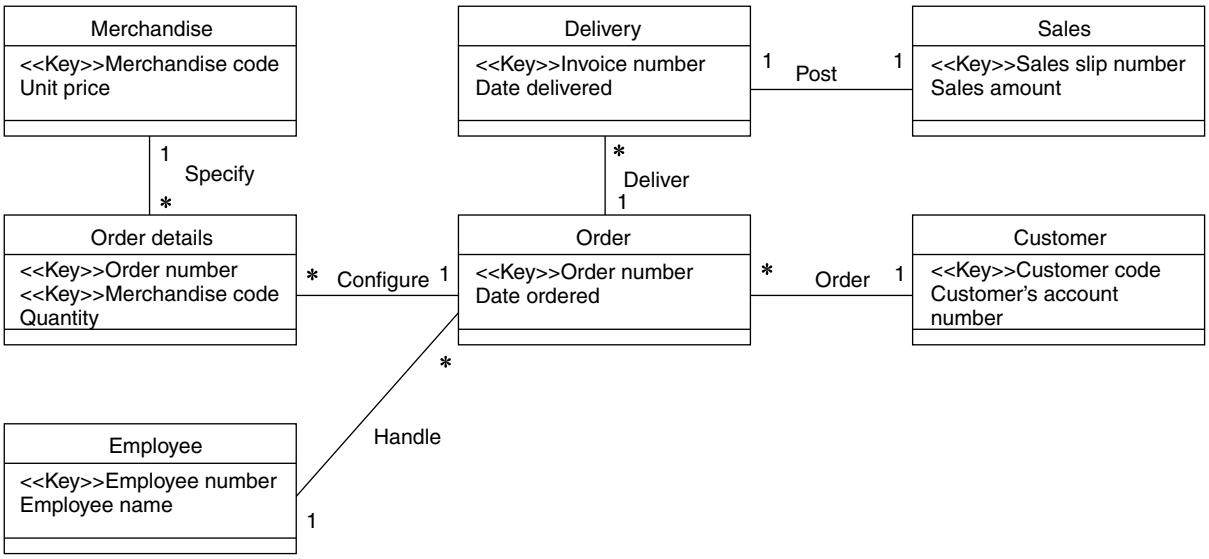

Figure 4
Example of data modeling.

5.2 Improvement of quality and efficiency in development using development tools

Errors easily occur in programming and testing because they are mainly done manually. Therefore, Fujitsu aims to improve quality by preventing errors in development work and increasing the testing coverage ratio through the use of tools that automatically generate the routine source code of a program and automatically repeat tests.

SDAS is intended to ensure quality and realize short-term development by using various tools, from development to testing (**Figure 5**). The following introduces the major tools of SDAS.

1) Integrated development environment: the open standard Apworks

Apworks is an integrated development environment that supports the development cycle as a whole, from analysis to testing and debugging. It provides a wide variety of development functionalities, for example, Unified Modeling Language (UML) modeling, graphical user interface (GUI) screen development, local testing and debugging of server applications, and form development. This environment integrates individual features based on the open-source software, Eclipse,[4] so development can be performed smoothly with familiar operability. Moreover, it enables efficient development through the use of a function that automatically creates prototype source code and by creating and reusing templates for routine coding.

2) Testing support tools

In the test phase, when bugs are fixed or specifications are changed, the already performed tests must be repeated to check that functionality has not been degraded.

In these cases, a regression test can be performed efficiently by using a tool that automatically repeats testing.

Fujitsu provides tools for automating tests, for example, SIMPLIA/JF JudgePruefer, to create and implement test drivers for Java server applications and SIMPLIA/TF-WebTest to test Servlet/JSP screens automatically.

Tools that improve test quality and efficiency are also available, for example, tools for creating test items and test data, comparing data, and measuring test coverage ratios.

3) Document generating tools

In projects that require short-term develop-

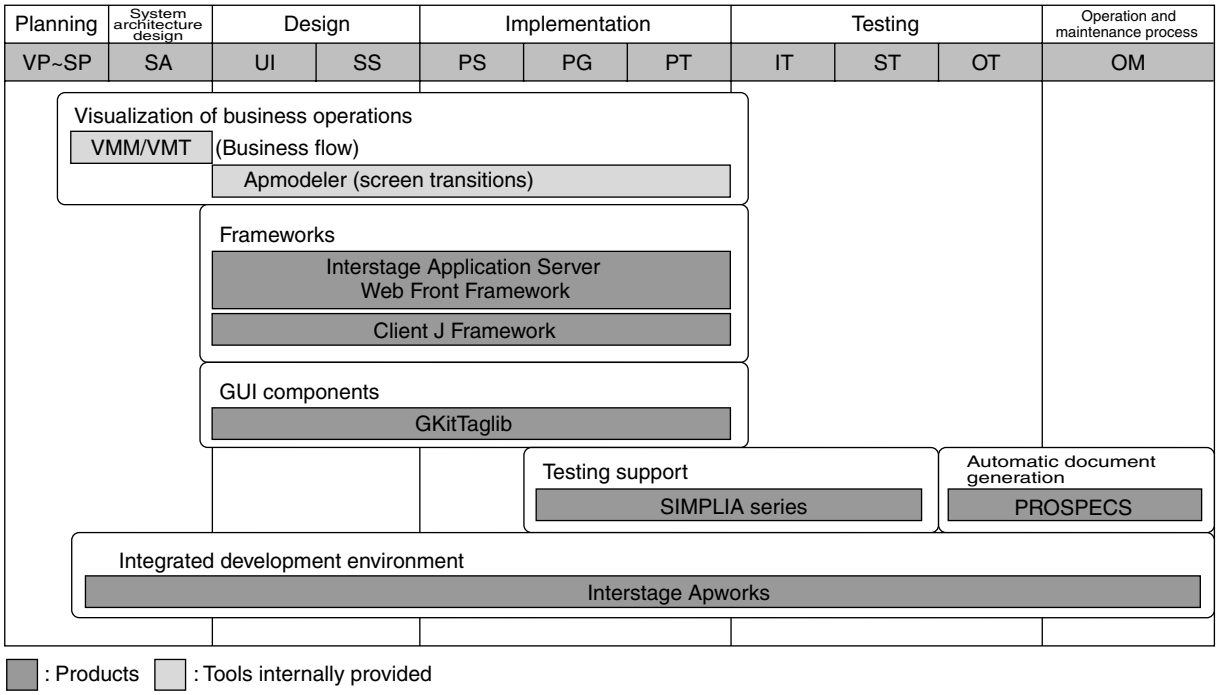

Figure 5
SDAS tools and development phase.

ment, it can be difficult to schedule sufficient time for documentation. Therefore, tools that automatically generate maintenance documents from source code can help shorten a documentation period considerably. Automatically created documents are also effective when transferring a development task or maintenance task to other personnel and developing an understanding of program specifications.

4) Development environment for COBOL compatible with various platforms

Even in Web systems, development using COBOL may be appropriate when using existing COBOL source code and programming knowledge or building processes that handle a large number of batch processes. Fujitsu's NetCOBOL is a COBOL development environment that can run on open platforms; it adopts the latest open technologies, while maintaining high compatibility with COBOL85 on mainframes.

NetCOBOL can also be used effectively for migrations involving the huge volume of existing COBOL assets.

5.3 Achievement of efficiency, high reliability, and expandability using application frameworks

The word "frameworks" in the software world means a group of software components, such as that represented by the Model-View-Controller (MVC) model.[note)] Frameworks are already implemented with functions generally used for business applications and provide an application platform that is based on a particular application structure. In system development using frameworks, applications must be developed according to certain rules. Frameworks standardize application structures and make expansion of functionality and maintenance easy. Moreover,

note) A software design model for implementing systems using a combination of three elements: models, which are central to processing; views, which take care of display and output; and controllers, which receive inputs and control views and models.
This model makes it easy to modify and maintain a program because such function-by-function separation of structures localizes the impact of modifications.

because frameworks provide a basic control logic, developers can reduce their programming work, which improves the efficiency of programming and testing and also improves quality (**Figure 6**).

The following introduces the Java application development frameworks that are applied in ComponentAA Development Method.

1) Web Front Framework

Web Front Framework is a framework for J2EE applications included in Interstage Application Server and Interstage Business Application Server. In this framework, Web-based applications are separated into screen, communication, and business logic. This separation enables developers to concentrate on the programming of business logic because the framework handles data transfer between screens and server applications. In addition, Web Front Framework can reduce the impact of a modification and therefore the framework realizes an application structure that makes it easy to expand and maintain an application.

2) Client J Framework

Client J Framework is a framework for developing screens in Java applet form. To incorporate know-how about developing large-scale business systems, it provides a mechanism for making screen transitions and input/output field attributes configurable from external definitions. This mechanism increases the productivity and quality of applications.

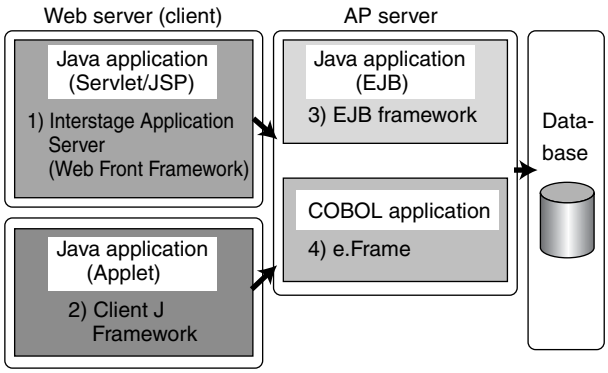

Figure 6
Fujitsu's frameworks and application architecture.

5.4 Software asset configuration management

As the size of system development grows, the type and quantity of software assets that need to be developed become huge. In particular, the type and quantity of Web system development assets, for example, Java source code, JavaServer Pages (JSP), Servlets, EJB-related files, and HTML and XML files, are likely to increase. Furthermore, a development task is often divided and performed by multiple development teams, or in order to shorten a development period, development and testing of multiple subsystems are often implemented in parallel. Under these circumstances, it is important to manage source code histories and establish operation rules for configuration management to ensure the control of common components. It is also important to have configuration management tools that support configuration management. Fujitsu provides support tools for configuration management and techniques to manage configuration according to the size of a project.

5.5 TransMigration Services for more effective use of existing software assets

Existing systems that enterprises have been operating over time are now an integral part of management. An important challenge for many enterprises that are under pressure to renew their entire systems to cope with a new business environment is how to reconstruct their huge volume of existing assets. Fujitsu provides the TransMigration Services based on its know-how about rebuilding systems with flexibility while effectively using existing systems. Following SOA, the TransMigration Services makes the use of existing development assets more effective. This service uses a method of constructing flexible and expandable systems and middleware that enable mainframe applications to operate in an open environment with minimum alteration.

6. Project management

This section introduces some project management technologies for the effective implementation of application development using SDAS.

1) Third-party quality assessment

Fujitsu performs internal third-party quality assessment using dedicated teams.

2) Knowledge management for system development

In system development projects, it is important to share information regarding design and progress across several groups, for example, business application development groups, platform groups, and standardization groups. Fujitsu uses an internal system that supports the work of project managers and information sharing among project members using knowledge management know-how.

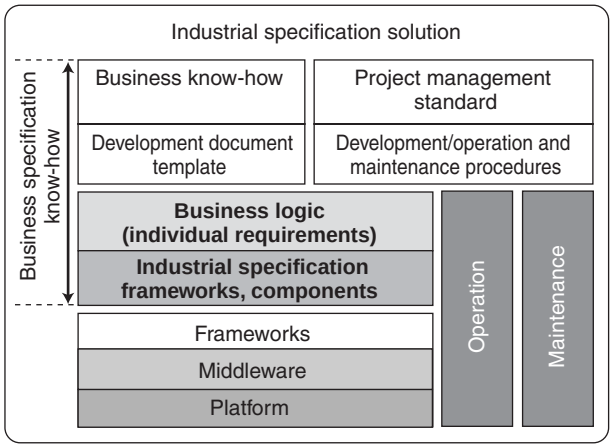

Figure 7
Industrial specification solution.

7. Accumulation and reuse of know-how about each business category

Fujitsu provides various solutions for different business categories, for example, finance, manufacturing, distribution, communications, and public services. In order to construct high-quality systems in a shorter period of time, Fujitsu applies its business know-how to frameworks that are customized to fit the business category. Moreover, Fujitsu is developing methods of reusing business models to enable early definition of customer requirements (**Figure 7**). The purposes of industrial solutions are 1) to customize the standard development method of Fujitsu to suit the characteristics of individual industries and 2) to quickly build high-quality systems suited to customers' requirements. Industry solutions also include procedures for project management, operation, and maintenance.

In addition, Fujitsu is preparing solutions with high expandability and maintainability by adopting SOA.

Fujitsu's specific approaches to accumulation and reuse of know-how are described in case studies in this special issue.

8. Collaboration with industry and industry-affiliated entities

This section describes Fujitsu's activities in the Software Engineering Center (SEC).

The SEC was established on the initiative of the Japanese Ministry of Economy, Trade and Industry to address the goals of improving the international competitiveness of Japan's software industry and promoting technology development, the achievement of international standards, and the development of core human resources.

Since the SEC's establishment, Fujitsu, as a representative of industry, has been playing a leading role by proactively participating in these activities. For example, Fujitsu has provided data regarding system development projects, proposed a method of measuring the scale of functionality, and contributed to the spread of SEC outputs. Through the SEC, we have been working to strengthen the international competitiveness of Japan's software beyond the boundaries of academic, business, and government circles.

9. Conclusion

This paper described the background and goals of the SDAS methodology and outlined its technologies and products. For details of the technologies and products of SDAS and some representative examples, refer to the other papers in this special issue.

In order to provide high-quality solutions to customers by effectively applying cutting-edge technologies, Fujitsu intends to continue improving and enhancing SDAS by adopting the latest technologies and methods and through feedback from various system development projects. Moreover, Fujitsu will globally expand SDAS within the Fujitsu Group.

References

1) Fujitsu's comprehensive system development architecture "SDAS." (in Japanese). *http://segroup.fujitsu.com/sdas/*
2) Fujitsu's IT strategy "TRIOLE." *http://www.fujitsu.com/global/services/solutions/triole/*
3) K. Hashimoto and T. Oshima: Fujitsu's Approach to Component-base system development. (in Japanese). *FUJITSU*, **50**, 2, p.72-77 (1999). *http://magazine.fujitsu.com/vol50-2/paper03.pdf*
4) Eclipse. *http://www.eclipse.org*

Takeshi Oshima, *Fujitsu Ltd.*
Mr. Oshima majored in Applied Physics and graduated from National Defense Academy in 1982 and majored in Control Engineering and graduated from National Defense Academy Graduate School of Science and Engineering, Yokosuka, Japan in 1987. He joined Fujitsu Ltd., Tokyo, Japan in 1991, where he has been engaged in software development technology.

Masayuki Kashiwagi, *Fujitsu Ltd.*
Mr. Kashiwagi received the B.E. and M.E. degrees in Computer and Information Sciences from Ibaraki University, Hitachi, Japan in 1981 and 1983, respectively. He joined Fujitsu Ltd., Numazu, Japan in 1983, where he had been engaged in software development. Since 2003, he has been engaged in support of application software development.

Hiroshi Fukao, *Fujitsu Ltd.*
Mr. Fukao received the B.E. degree in Electronic Engineering from Tohoku University, Sendai, Japan in 1989. He joined Fujitsu Ltd., Kawasaki, Japan in 1989, where he has been engaged in software development technology.