# Expansion of Application Configuration Management

● Junichi Kamakura

*(Manuscript received November 11, 2009)*

**Application maintenance requires total success when changing existing applications. Even when changing a few lines of programming, it is necessary to investigate the impact of those changes and conduct tests to make sure that they do not affect other configuration items. These investigations and tests make up 80 to 90% of the application maintenance cost. The accuracy and efficiency of investigating the effects of any change need to be improved. Usual investigation depends on experienced and skilled engineers who are in charge of the system and application. They not only investigate applications but also the framework of applications and infrastructure. They also consider operations. We arranged configuration items that are needed for investigations and constructed a model. That model was applied to CentraSite's registry, and we made an investigation method. By using that method, all the engineers working in the operation and maintenance process can conduct investigations with the same accuracy and efficiency.**

## 1. Introduction

When maintaining applications, changing even only a few lines of a program will necessitate a thorough investigation of that change's effects. In addition, testing needs to be conducted not only to check the changes made but also to make sure that those changes do not affect other items (regression test). The person-hours required for these investigations and tests to identify the scope of maintenance account for 40% of all the person-hours needed for program changes.[1] Regression tests account for 40 to 50% of all processes (**Figure 1**). Kazuya Masui refers to this situation as "a two-humped camel" at the Software Evolution Research Consortium (SERC).[2]

To reliably identify the extent of impact of changes to a program, it is necessary to understand programs constituting previously released versions and their levels of change as well as correlations between programs. In addition, the configuration information (combined information on OS and middleware installations) of the server on which the application in question runs is becoming more complicated and it is important to have an understanding of it. In many cases, however, these types of information are not controlled while taking into consideration the ease of data retrieval, and this makes identifying the extent of impact less efficient. A system that allows engineers in charge of application maintenance to investigate these issues is required.

Accordingly, we have expanded the application configuration management that has conventionally been implemented, and built a system to give us an understanding of configuration information including the application operation environment.

This paper clarifies the difficulties of

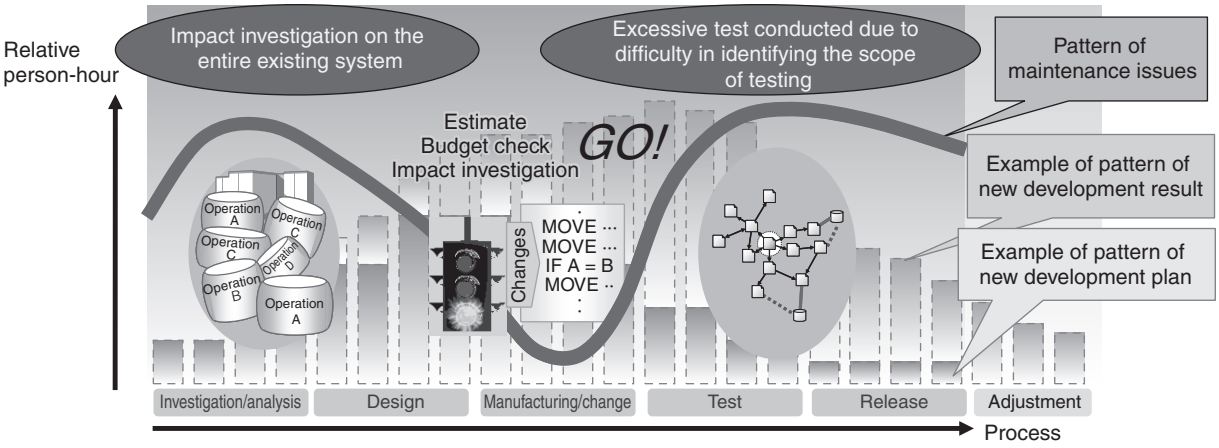


Figure 1
Many person-hours of investigations and tests are required for changing only a few lines.

conducting application maintenance and the actual situations in which impact investigations require many person-hours in application maintenance, and it presents correlations between configuration items and a management model to describe their implementation.

## 2. Difficulty of application maintenance

The object of application maintenance stated in Chapter 6 of JIS X0161: 2008 is "to modify the existing software product while preserving its integrity."[3] The difficulty with application maintenance is making sure that this integrity is preserved. Specifically, it is hard to investigate any impact on the existing system and also difficult to completely ensure that such maintenance does not affect other items. Unlike new development, application maintenance requires many person-hours and a lot of time to conduct the processes of impact investigation in order to preserve the integrity and reach the stage of a test release. The reasons for this include (i) even when just a few lines of changes are to be made to the program, the impact on the entire existing system needs to be investigated before they can be made, so as to judge their appropriateness, and (ii) regression tests on the entire system need to be conducted due to the difficulty of identifying the scope of testing after the changes are made. Because of this, a challenge in application maintenance is how to conduct these two tasks (impact investigation and test release) in a thorough and efficient way. The many person-hours required for impact investigation are due to the complexity of OS and middleware combinations, much more complex than in the past, and the enormous scope of maintenance.

The system of the industrial corporation A shown in **Table 1** includes 960 screens and 220 forms and is equivalent to approximately 2000 programs (5000 programs in terms of managed assets) if developed in COBOL using a host. If developed with open systems and Java, the scope of management is large enough to include approximately 150 000 files. With financial corporation B, the scope of management is as large as approximately 40 000 files. A large proportion of the maintenance costs is spent on accurately identifying the scope of impact of maintenance on these many application assets.

## 3. Model of impact investigation in application maintenance

We have clarified how the impacts of application specification changes are investigated and why that investigation requires many person-

Table1
Scope of configuration management growing enormously.

| Category of business | Size of assets to be target of configuration management | Major development language | Notes |
|---|---|---|---|
| Industrial corporation A | 150 000 files<br>(960 screens, 220 forms) | Java | Number of programs converted in terms of host asset<br>⇒ COBOL: equivalent to 2000 programs<br>⇒ Related assets: equivalent to 5500 programs |
| Financial corporation B | 40 000 files<br>(online transaction, batch) | C, Java | Average number of steps in source<br>⇒ 348 steps/program on average<br>(5.1 Msteps/14 518 programs) |
| Credit corporation C | 25 000 files<br>(500 screens, 3000 forms) | C, COBOL, Java, YPS | Type of extension<br>⇒ 76 different extensions<br>(e.g. java, class, so, cbl, cpy, pco, smd, yac, bat, dll, frm) |
| Distribution corporation D | 3745 sources<br>(including manually written 510) | Java<br>(EZDeveloper) | Average number of steps (manually written source)<br>⇒ 101 steps/program on average<br>⇒ 80% of programs contain 100 steps or less |

hours and time for application maintenance (**Figure 2**).

If a new requirement is given by the management layer or a user department, its impact on the system requirements needs to be verified. To find out the impact on the system requirements, it is necessary to conduct investigations while taking into account application limitations and implementation methods. For this purpose, investigation is conducted from the following three perspectives:

1) Verification of impacts arising from limiting conditions of the application: quantity limitations due to the sizes of tables, etc (①).
2) Investigation from the viewpoint of how much impact occurs on which application within the application: relations with sub-modules, checking for any impact on other modules through a database, etc (②).
3) Impact investigation from the perspective of operation: impact on operational requirements including performance, capacity and time (③).

To investigate impacts between applications, the impact on the application platform must be taken into consideration. The impact on the application platform is also investigated from three perspectives:

1) Verification of the impact arising from limiting conditions of the application platform (④).
2) Investigation of the extent of impact within the components of the application platform (⑤).
3) Impact investigation from the perspective of operation: investigation of the impact caused by the change of application platform on operational conditions including the performance, capacity and time (⑥).

In a similar fashion, the impact on the respective layers of the system platform and operation & maintenance is investigated (⑦ to ⑨). Based on the results of the impact investigation, proposals are made regarding the methods of requirement implementation and change, and this is then reported to the operations layer. The operations layer consults with the management layer, which generates yet another requirement. Then, another impact investigation is conducted in relation to this other requirement and proposals are prepared again regarding the methods of implementation and change. This process is repeated before the changes are actually made. For this reason, when maintaining applications the number of person-hours required for impact investigation before making changes to a program grows to be

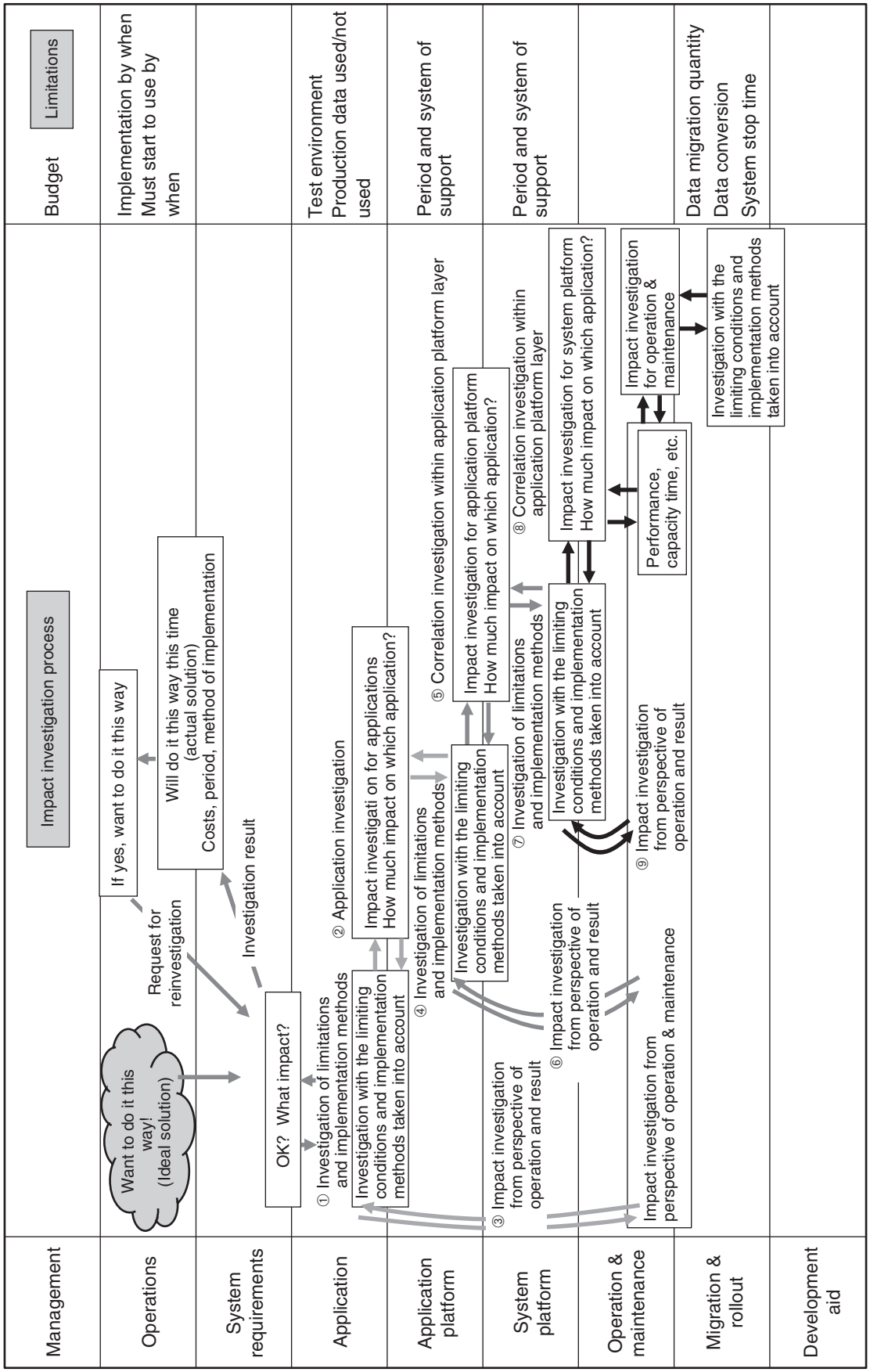


Figure 2
Process of investigating impact.

huge even if the program changes end up to be only a few lines.

As mentioned earlier, impact investigation is not confined to investigations on the application but must also cover investigations on items such as the application platform and system platform, which requires correct management of the configuration information for the existing system resources in order to conduct impact investigation. A system that allows correct management and retrieval of the application configuration, application platform configuration, and operational environment configuration is required.

## 4. Actual situations of configuration management

We have investigated how configuration information is managed in application maintenance. A diagnosis has been made of the operation & maintenance process in more than 1700 projects, and the findings show that the level of implementation of configuration management was low (Level 0 to 1) in approximately 40% of the projects and that management depended on the know-how of individuals (**Figure 3**). Because of this, engineers in charge of application maintenance could not make effective use of configuration information. The managed configuration information is not used when impact investigation is conducted and the result of investigation varies depending on the experience and know-how of the engineer in charge.

## 5. Correlations between components and management model

Verifying the site of application maintenance has shown that, if any problem occurs when conducting impact investigation there is no effective means of identifying the scope in question because of the low level of implementation of configuration management. In addition, this narrowing down of the scope may result in division of the scope of investigation because of division of maintenance among different engineers such as an engineer in charge of operations and another in charge of infrastructure. For this reason, the

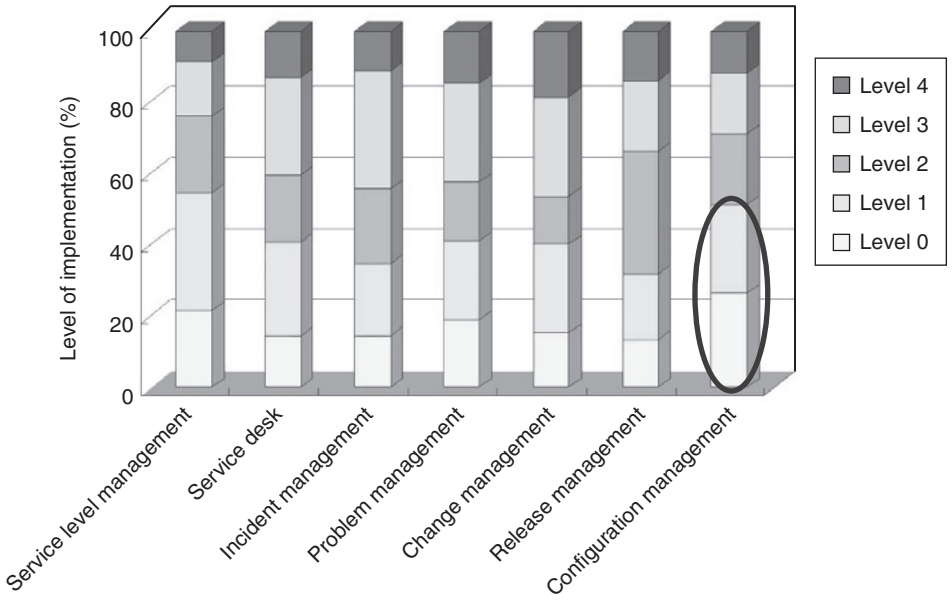


Figure 3
Result of diagnosing operation and maintenance process.

configuration information obtained as a result of the investigations conducted by the respective engineers is not shared, which makes the impact investigation less useful than it could be. To address this issue, we have studied how to manage the correlations between the infrastructure and applications to acquire effective information.

As a perspective for organizing configuration items in an IT system, we have attached the greatest importance to understanding the extent of impact in maintenance. As basic information for extracting this information, we have used the IT infrastructure configuration information RCXML (approximately 1000 entities), for which Fujitsu Laboratories has been formulating specifications since 2004.

Regarding the correlations with applications, we have sampled past problems and narrowed down items from the viewpoint of whether the knowledge of the correlation could have prevented the problem. This process included repeated consolidation and review of items for each case for narrowing down, and in this way we have consolidated the items into 26 entities including program sources, makefiles and execution modules (**Figure 4**).

Furthermore, based on the examples of investigating problems that occurred, we have taken some measures to improve the efficiency of impact investigation including the provision of relations between the execution modules and program sources (dotted lines in Figure 4), which is intended to improve retrieval performance, in addition to the actual relations between the execution modules, makefiles and program sources (solid lines in Figure 4). This allows us to track the relevant assets in our investigation regardless of the asset used as the starting point

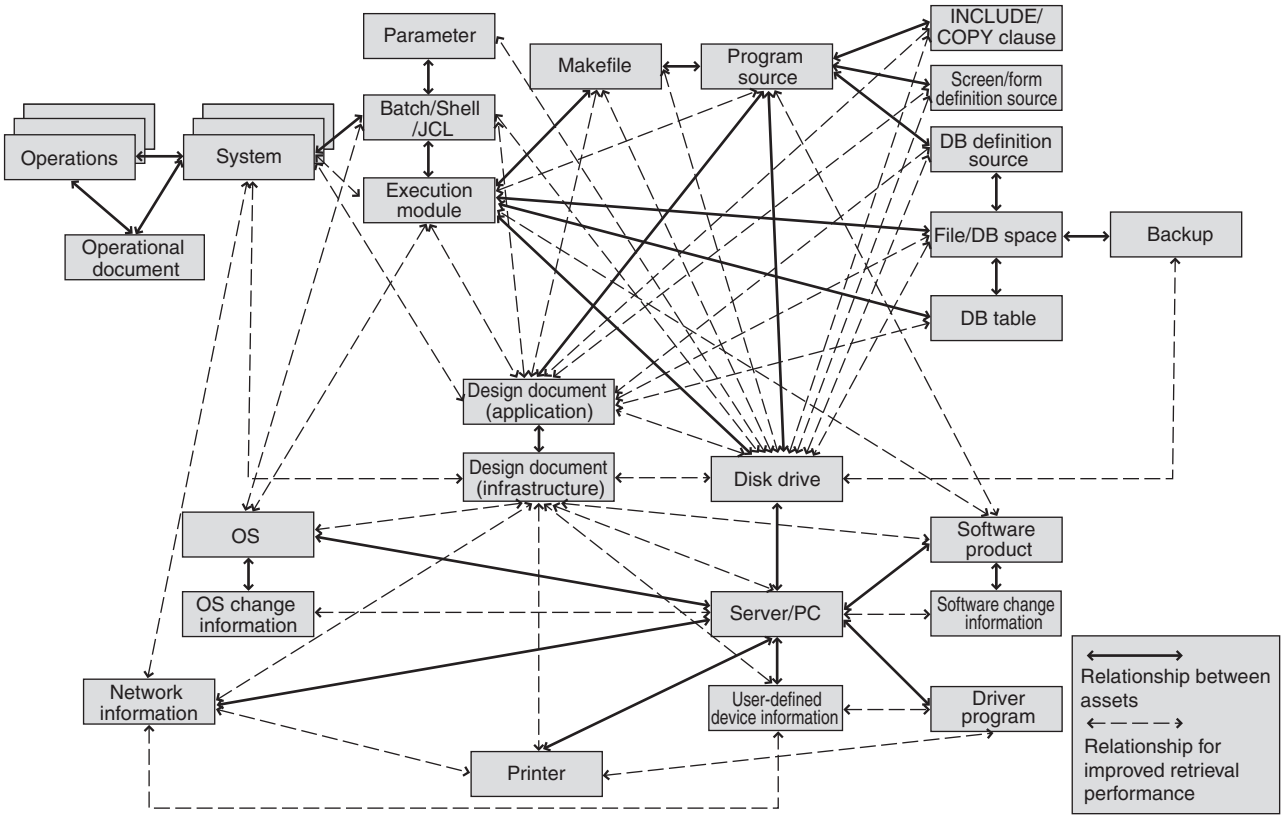


Figure 4
Information interrelationships of configuration management.

and the extent of impact of maintenance can be efficiently identified.

## 6. Implementation of management model

We have conducted a simulation from a problem occurring to the identification of its cause based on configuration information. To conduct this simulation, we used Fujitsu's CentraSite (**Figure 5**) as the registry of correlations of the actual configuration information.

This product allows registration of relations (correlations) between program sources, execution modules, servers, OS, and such like as the registry and relationships can be retrieved and displayed from various viewpoints. For example, relations such as those between a program and other programs or documents and between a program and operation server can be registered for reference. We have built a verification environment based on the correlations included in the configuration management information shown in Figure 4. The following presents an example of the simulation conducted in this environment.

### 6.1 Case 1: Checking for differences with the installed middleware

We have conducted verification based on an actual problem that occurred. The purpose was to see if the middleware required by the execution module of an application had been installed in the operational environment of the application in question. In the past, it was possible to investigate the relations between applications but it was assumed without question that the necessary middleware had been installed in an operational environment, and this was not even investigated. In addition, engineers in charge of infrastructure configuration management did not have information on when the middleware required for additional applications would be actually required. For this reason, the necessary middleware was not available and the application

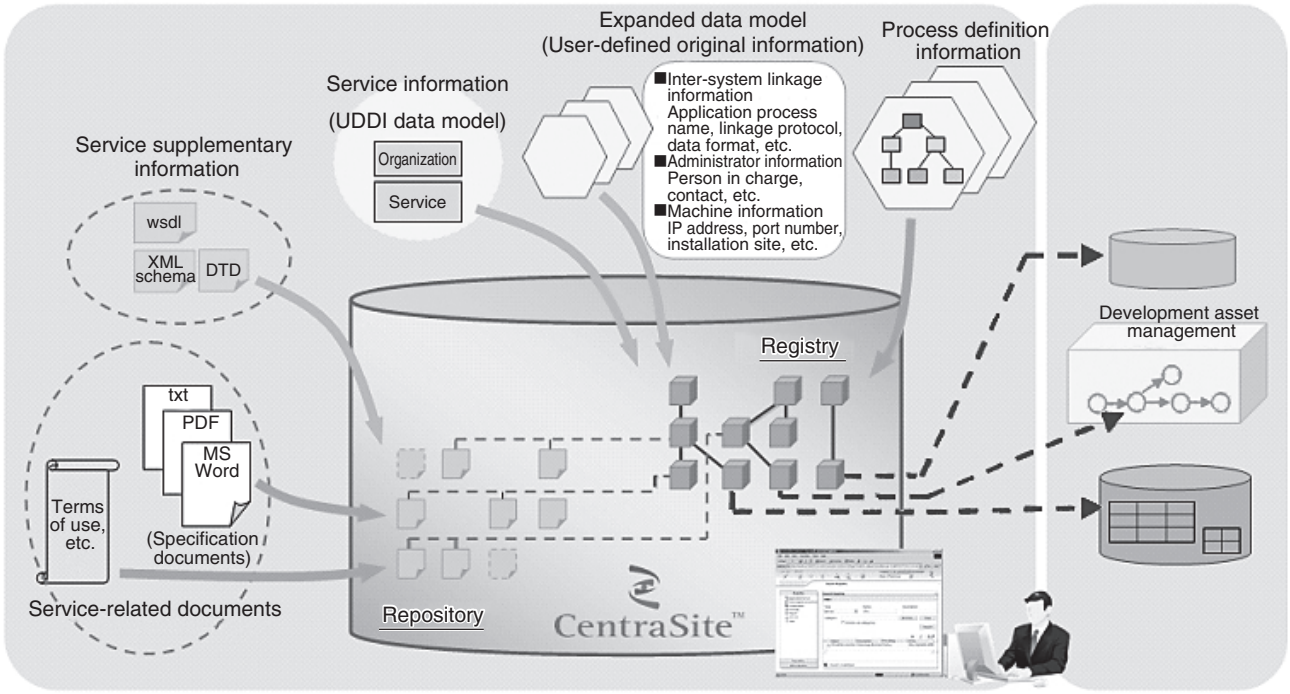


Figure 5
Overview of CentraSite.

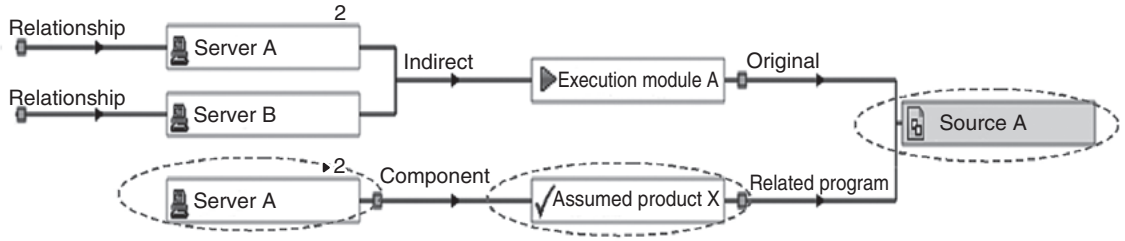


Figure 6
Relationships among server, middleware and execution module.

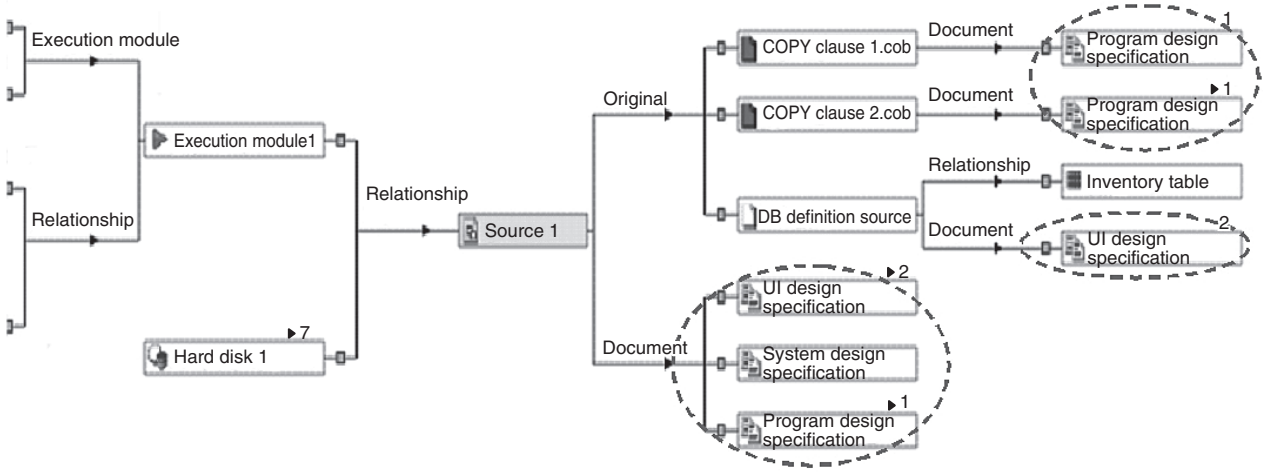


Figure 7
Target source and related documents.

in question could not be executed.

In the CentraSite registry, the relations among the server, OS, middleware and application execution modules are registered. **Figure 6** describes retrieval from this registry. A search starting from the program name ("Source A" in Figure 6) allows retrieval of information on the execution server and installed middleware. A search starting from the middleware name ("Assumed product X" in Figure 6) also allows retrieval of the server on which it is installed or execution of the module name run on the server in question. We confirmed that, even if different engineers in charge of impact investigation use different starting points of retrieval, the same investigation result can be obtained.

## 6.2 Case 2: Knowledge of documents in relation to the target program

To make changes to an application, it is essential to refer to the program design specification and documents of conventions required for making changes (naming conventions, coding conventions, change conventions, etc.). Retrieval in such a case is described in **Figure 7**. First, the program source asset is selected from the application layer ("Source 1" in Figure 7). The name of the execution module of the program in question is shown on the left and the name of the design specification and name of the documents of conventions along with the relations with sub-modules and data definitions used by the program in question are shown on the right, and this allows the required documents to be referred

to.

In this way, making the relevant design specifications and documents of conventions available for referencing can improve the quality of impact investigation.

One advantage of using CentraSite for investigating the configuration information is that it allows efficient investigation of correlations between applications and between applications and infrastructure with the same level of accuracy regardless of the person who conducts the investigation. With the conventional method of investigation, information other than program assets needed to be investigated from documents, which may require engineers in charge to take manual procedures such as reading from lists or extracting setting values from text depending on the type of information to be investigated. This caused variations in the result of impact investigation depending on the engineer. Use of CentraSite has eliminated this variation and improved the accuracy and speed of impact investigation.

Furthermore, browsability and data retrieval by tools can be improved, thereby providing a means of sharing information when passing on know-how via a functional transfer between engineers, for example.

## 7. Future development

The application configuration management that has been implemented so far has been expanded to allow retrieval of infrastructure configuration information together with application configuration information. This has allowed a more reliable investigation of the impact of application specification changes, which in turn helps to improve application maintenance quality and eliminate problems during operation.

We believe that, in the future, the sharing of change management, which is currently implemented separately between applications and infrastructure, will help to further improve operational quality. For business continuity planning (BCP), there is a need to study measures according to the level of importance of systems and visualize system vulnerability. It is essential to clarify the configuration of IT assets that support critical operations requiring continuity. We intend to expand the use into the BCP area in addition to the configuration management for application maintenance.

## 8. Conclusion

This paper presents an expansion of application configuration management into modeling of information in five layers including the operations, application, infrastructure, data and backup layers for visualization, which improves the quality and efficiency of impact investigation in the process of application changes. We wish to continue to enhance Fujitsu's operation & maintenance service technology and help to improve the quality of application maintenance.

## References

1) FUJITSU Family Association LS Research Committee: System Reconstruction Taking Advantage of Existing Assets II. (in Japanese), Edited by LS Research Committee, 1995.
2) K. Masui et al.: Software Evolution Research Consortium: Software Maintenance—Based on ISO 14764. (in Japanese), Software Research Center, 2007.
3) JIS X0161: 2008. Software Engineering—Software Life Cycle Processes—Maintenance. (in Japanese).

**Junichi Kamakura**
*Fujitsu Ltd.*

Mr. Kamakura started his career in Fujitsu Ltd. in 1978. He was engaged in many projects relating to application program conversion. Those projects were for the systems of some major Japanese companies, Japanese governmental organizations, and Japanese local governmental units. Because of his knowledge and experience, he is currently the general manager of the Application Portfolio Management Competency Department in Fujitsu Ltd. He is also developing Fujitsu's trusted Cloud service and putting the method described in this paper into service.