OSS | Distributed Systems | User Customization

Technology Reports

Knowledge System Renewal Using Distributed Data-Driven Architecture

Networks have become more complex and sophisticated with the introduction of LTE services and All-IP networks, and in turn, the functionality required of the OSS used to monitor these networks has diversified. As a result development times have tended to increase. Because of this, there is a demand for OSSs that are able to track network changes flexibly. In the renewal of knowledge system equipment, we have changed to a D3A and developed a system that allows maintenance personnel to add functionality easily. This allows us to handle addition of various functions and network updates flexibly.

Core Network Development Department

Yuya Kuno
Masahiro Abe
Nobuya Minakata
Masanori Furutani

1. Introduction

There has been an increasing number of terminals, such as smartphones and photo frames, with a wide range of functions and diversifying needs on communications networks in recent years. In addition, to provide new services with these devices, and to provide high speed services such as LTE[*1], there is a tendency toward frequent introduction of new equipment. As a result, the variety and volume of equipment are increasing dramatically, and the maintenance work for monitoring this equipment is becoming more diverse and complex. A major issue for communications operators is the use of Operations Support Systems (OSS)[*2] to properly monitor this equipment as it increases in sophistication and complexity, and to continue to provide stable communications services.

For this issue, one of the OSSs, called the Network Operations Knowledge System[*3] (hereafter referred to as "Knowledge System"), provides an environment for maintenance personnel to create and modify maintenance tasks and work flows, and allows for flexible tracking of new equipment additions and increases in complexity of maintenance tasks [1]-[4].

This equipment was becoming outdated, so in FY2010, NTT DOCOMO carried out a renewal of the knowledge system. Rather than simply replacing the equipment, we studied how it had been used in the past and identified points for improvement, in order to increase utilization of the system. In this article, we describe the new architecture which we have adopted in renewing the knowledge system.

2. The Knowledge System

2.1 Knowledge System Overview

The Knowledge System supports the work of maintenance personnel by

*1 **LTE**: A specification for extending the 3G mobile communications format specified by the 3GPP, called the 3.9G communications format. NTT DOCOMO began a commercial LTE service called Xi in December 2010.

*2 **OSS**: Enterprise operational support systems. For communications operators, this can include some or all of: fault management, configuration management, charging management, perautomatically executing maintenance work-flows, which are descriptions of maintenance know-how in work-flow form. **Figure 1** shows the knowledge system in the overall system context.

The Knowledge Systems has three main functions.

- Gather information in real-time when alarms are issued, and take primary measures.

  Distinguish software or hardware causes for each alarm, gather analysis data, etc.

- Batch execution of set tasks

  This includes all preventative maintenance tasks in response to alarms and not requiring immediate action, detection of silent faults *4 through traffic analysis, automatic on-site repair of hardware failure, etc.

- Interactive measures initiated by maintenance personnel

  Includes selecting measures to be taken based on gathered data, etc.

To implement these functions, the Knowledge System is organized as shown in **Figure 2**. The Knowledge Systems is a platform that provides three types of functionality, including work scenarios, which describe maintenance tasks in workflow form, a work scenario engine, which interprets and automatically executes work scenarios, and Application Programming Interfaces (APIs) *5 for the work scenario engine. When a maintenance work flow needs to be modified or added, a maintenance person can edit or revise a work scenario in real time to change the maintenance tasks automatically executed by the Knowledge System [1].

In the previous Knowledge System, describing the work scenarios could be

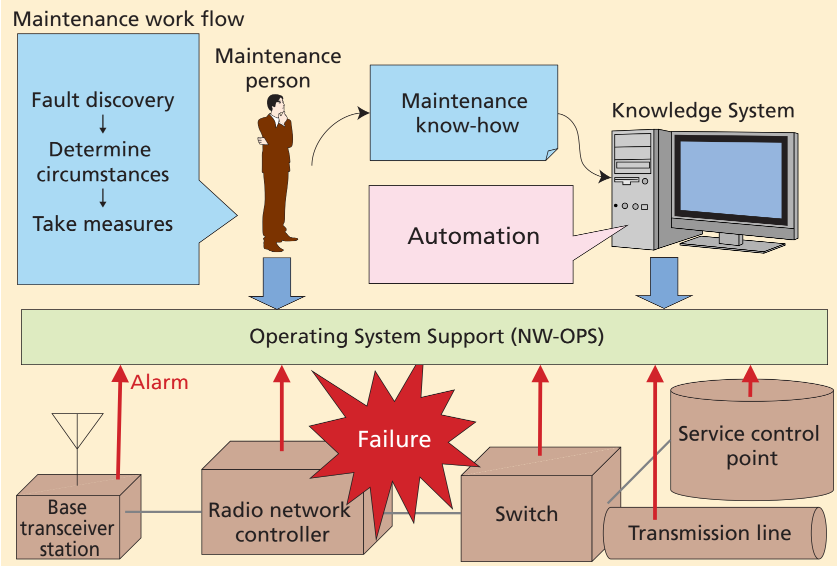


Figure 1 The Knowledge System in system context

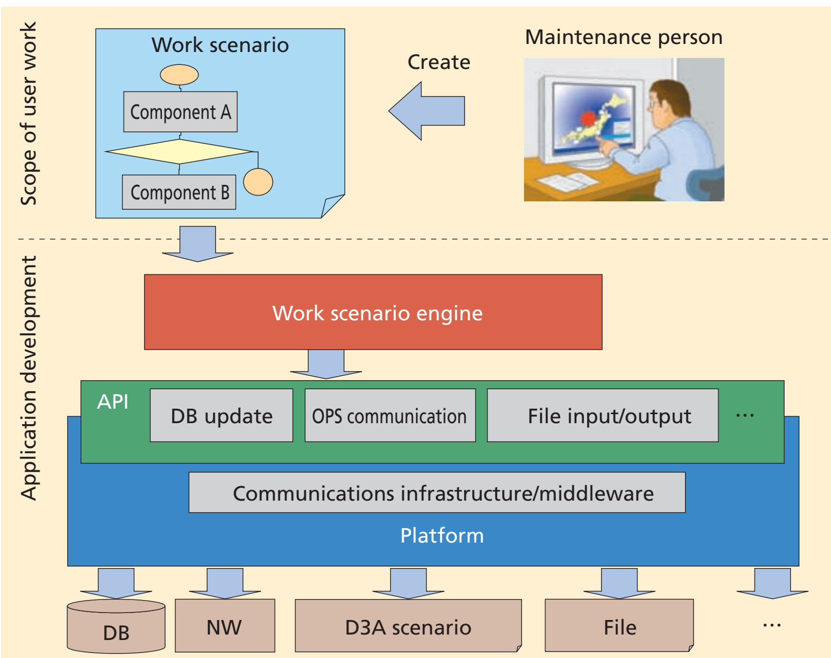


Figure 2 Knowledge System architectural diagram

formance management, and security management; for the networks and systems operating the services being provided.

*3 **Network Operation Knowledge System**: One of the OSS used by NTT DOCOMO, replacing maintenance workflows and supporting the work of maintenance personnel by executing workflows automatically.

*4 **Silent fault**: A failure that the maintenance personnel cannot detect such as those caused by breakdowns of the fault detection package and main processor, so that the equipment itself cannot recognize the fault.

*5 **API**: A set of commands and functions that can difficult depending on the type of information required, as shown in **Figure 3**, so some parts of the system were difficult for maintenance persons. Thus, for the new knowledge system, our objective was an architecture that allows work scenarios to be created by combining components in a GUI [*6], as their sophistication increases, to make them easier to create and to extend the range of what can be described.

2.2 New System Concept

The new Knowledge System was revised according to the following objectives.

1) Objective 1: Expand the Range of Work Scenarios that can be Described

The goal of the Knowledge System is to support maintenance personnel, so it is targeted at fixed tasks that are done frequently. Such work includes many tasks that can be analyzed and measures taken based on information from OSSs, so gathering information from OSSs has been a focus of the Knowledge System. However, because of the increasing sophistication of work scenarios, it is clear that the following types of functions will be needed in the future.

- Analysis methods and priorities that differ according to geographic characteristics such as susceptibility to typhoon or heavy snow, or time characteristics such as time of day or night.
- Switching from handling software faults to handling hardware (hereinafter reterred to as "HW") faults based on the results of measures taken in the past, such as the frequency and status of repairs (hereinafter referred to as "performance results").
- There is a need to utilize additional information from statistical analysis of performance results, such as traffic or alarm volume, as well as non-verbal know-how from maintenance personnel, whether it involves characteristics of the work, such as a fireworks display or other event, or particularities of the base transceiver station location, such as alarm tendencies.

In the new Knowledge System, expanded range of information can be used for input and output to work scenarios, allowing work characteristics and performance results, including the above. Multiple work scenarios can also be initiated according to task characteristics.

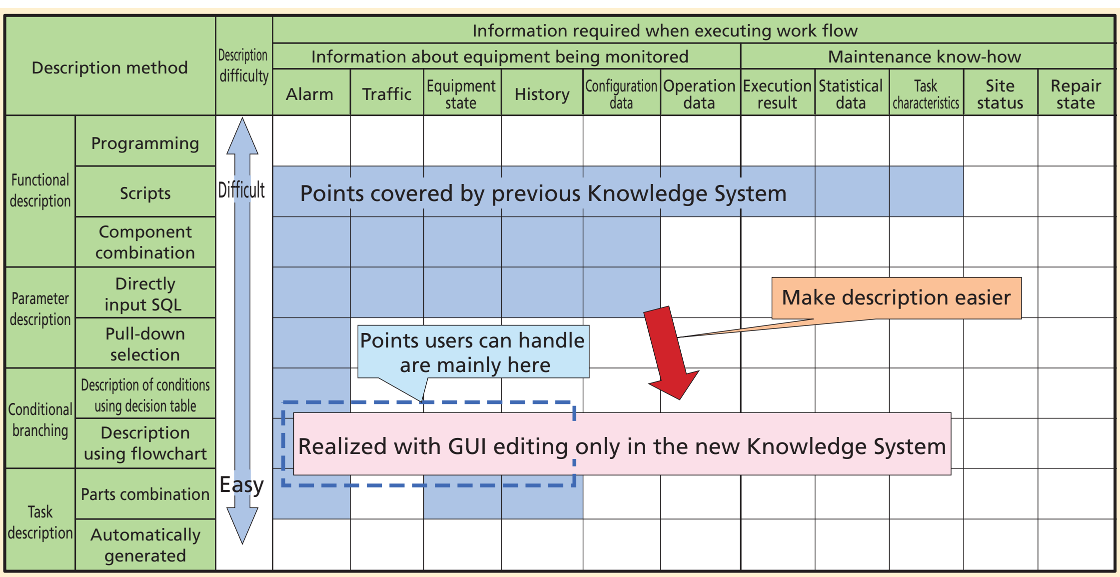


Figure 3 Work scenario expansion

be used for developing software for a particular platform.

*6 **GUI**: A user interface that uses many icons and other graphics on which basic operations are performed mainly with a pointing device.

2) Objective 2: Improve Performance through Scale-out

By achieving Objective 1, work scenarios can become more sophisticated, as maintenance know-how accumulates. As work scenarios become more sophisticated, the performance required also increases. It is difficult to estimate the volume of equipment to be monitored or the number of messages in the long term, so it is desirable to be able to increase performance at any time.

In the previous Knowledge System, it was necessary to concentrate all information needed for maintenance in a centralized model in order to infer causes, so it was necessary to scale up (replace with a higher-performance server) in order to improve the performance.

The new Knowledge System was designed to allow performance to be increased more flexibly, by simply adding hardware (scale out).

3) Objective 3: Promote Use of Internal and Open Source Products

The Knowledge System uses a commercial middleware[*7] products. The main components are given below.

- OS: Solaris™[*8]
- Communications engine: Common Object Request Broker Architecture (CORBA)[*9]
- Rule engine: JRules™[*10]

In renewing the system, it was necessary to select middleware able to achieve Objectives 1 and 2 while conforming to existing functionality, but we also wanted to promote use of Distributed Data-Driven Architecture (D3A)[*11] [5], which is one of our products, as well as Open Source Software.

3. Implementation Details

In order to achieve the objectives described in Chapter 2, we took the following measures in the new Knowledge System.

3.1 Expanding the Range of Work Scenarios that can be Described

To achieve Objective 1, we expanded the range of customization possible in the Knowledge System. In the previous Knowledge System, it was possible to customize processing logic in the form of work scenarios. In the new system, it is also possible to customize work scenario input and output information.

1) Input Data Customization

An overview input data customization is shown in **Figure 4**.

According to Objective 1, there are three types of input information required for work flows: monitoring, control, lookup and distributed data obtained through interfaces from monitoring, performance results from the work flow, and task characteristics. In the previous Knowledge System, work scenarios used information obtained from the monitored equipment. The new Knowledge System can also use work flow performance results and task characteristics as input to the work scenario. This data is input directly by maintenance personnel, or is input or referenced using work scenarios created by maintenance personnel, so it must be easy to use. Thus, for the new Knowledge System, we simplified work scenarios and direct-input tasks by preparing a database (hereinafter referred to as "DB") that maintenance personnel can

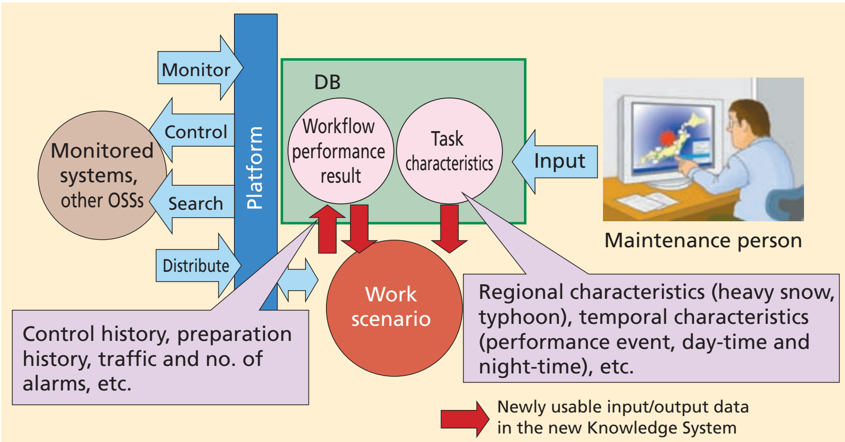


Figure 4 Input data customization image

*7 **Middleware**: Software positioned between the OS and actual applications providing common functions for diverse applications thereby making application development more efficient.

*8 **Solaris™**: A UNIX-style OS developed and sold by Oracle Corp.
Oracle and Java are registered trademarks of Oracle Corp., its subsidiaries, and affiliates in the United States and other countries. Compa-

access freely as well as DB access components.

2) Output Data Customization

Screens are required to allow maintenance personnel to check work scenario performance results. Information required by maintenance personnel on these screens includes data from the equipment being monitored (alert history, traffic, etc.), and the results of performing the work scenario. In the new Knowledge System, all of the information needed on the display screens, including work scenario performance results, is stored in the DB as shown in **Figure 5**, and screens can retrieve the required data from the corresponding DB using DB access components. Also, the retrieved data can be displayed in any format and layout using screen display components, increasing usability. We used Adobe® Flex™*12 components for display of DB content, implementing drag and drop operations to customize screens as shown in **Figure 6**.

3.2 From Scale-up to Scale-out

As described in Objective 2, scaling-up was required to increase performance of the previous Knowledge System. For the new system, we used virtualization technology, which allows us to increase performance by "scaling out."

The key to making systems that require centralized data distributed, is to know the location of the data, and to maintain the processing sequence while increasing the possible communication paths. For this project, we used D3A to achieve this.

D3A is middleware for building large-scale distributed systems by separating the functions of an application into "Elements (hereinafter referred to as "EL")" and then implementing selection of a communications path to the appropriate EL according to the scenario [1]-[4], [6].

An overview of D3A operation is shown in **Figure 7**. By using D3A to redefine the Knowledge System func-

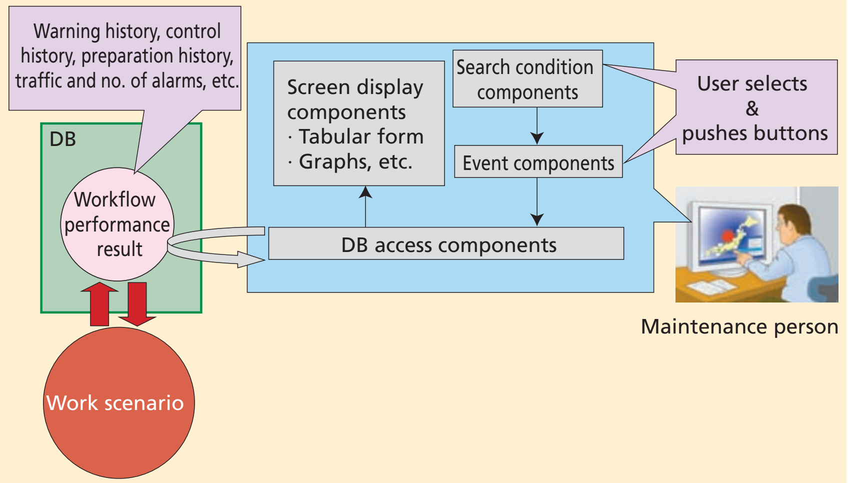


Figure 5 Screen customization image

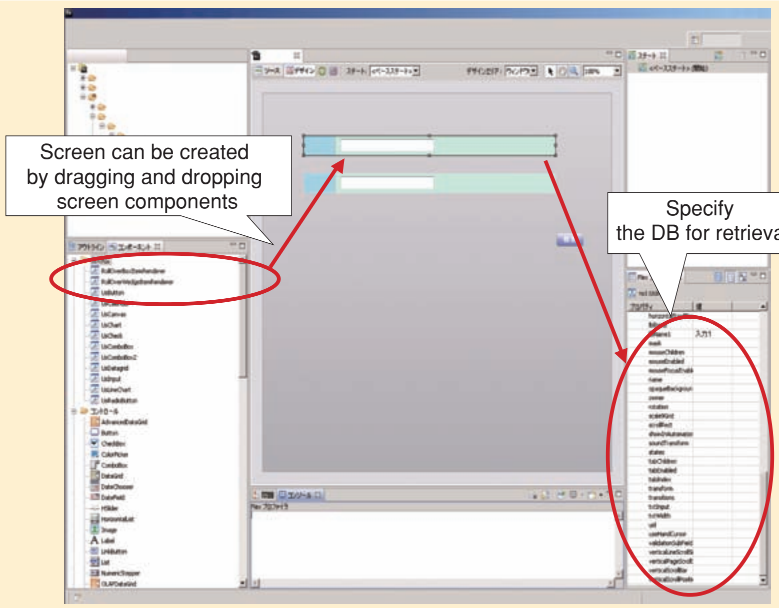


Figure 6 Screen customization methods

ny and product names appearing in the text are trademarks or registered trademarks of the respective companies.

*9 **CORBA**: An architecture that allows software written in a variety of programming languages on different computers to work together. A standard specification defined by the Object Management Group (OMG).

*10 **JRules™**: A business rule management system from ILOG Inc. (now a subsidiary of IBM Corp.). Allows collaborative work among users, from application development through to the business-rule management life cycle. A registered trademark of IBM Corp., United tions as EL, D3A is able to automatically select paths to the servers maintaining the required information, and it is no longer necessary to centralize data on physical hardware. Thus it is possible to increase performance by adding hardware.

Similarly, in the previous Knowledge System it was not possible to increase the number of servers for the DB, so the DB also had to be scaled up to increase performance. For the new system, we used a combination of Oracle Real-Application Clusters (RAC)[*13] and Oracle Automatic Storage Management (ASM)[*14], as shown in **Figure 8**, making it possible to scale out the DB as well.

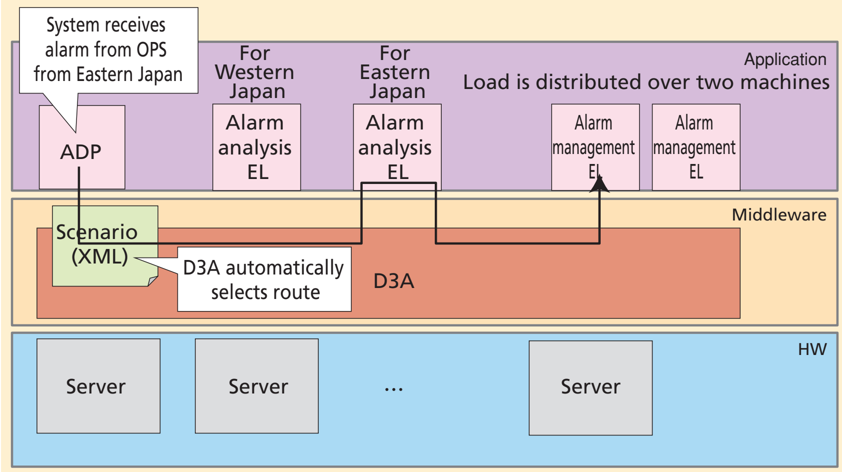
In RAC technology, a connection from a client is automatically allocated to the DB server with the lowest load, so multiple DB servers appear as a single, giant virtual DB server. In this way, using RAC allows performance to be increased by simply adding DB servers if DB server performance is inadequate.

Using ASM also allows multiple storage devices to appear as a single, giant storage device. Thus, if the DB becomes low on storage, capacity can be increased by simply adding storage devices.

3.3 Architecture Changes, Including Middleware

As described in Objective 3, we used software developed by NTT DOCOMO and Open Source software in development of the new Knowledge System.

We used D3A, which is used as communications infrastructure for the network operations system (NW-OPS)[*15], to handle scale-out as described in Section 3.2 and because it is an NTT DOCOMO product[5][7]. To minimize the amount of change to applications for the new Knowledge System due to the change in communications middleware, we developed CORBA/D3A conversion functions



eXtensible Markup Language (XML): A markup language proposed by the World Wide Web Consortium (W3C) for describing the meaning and structure of text and data. It is extensible, and users can specify their own tags.

Figure 7 Overview of D3A operation

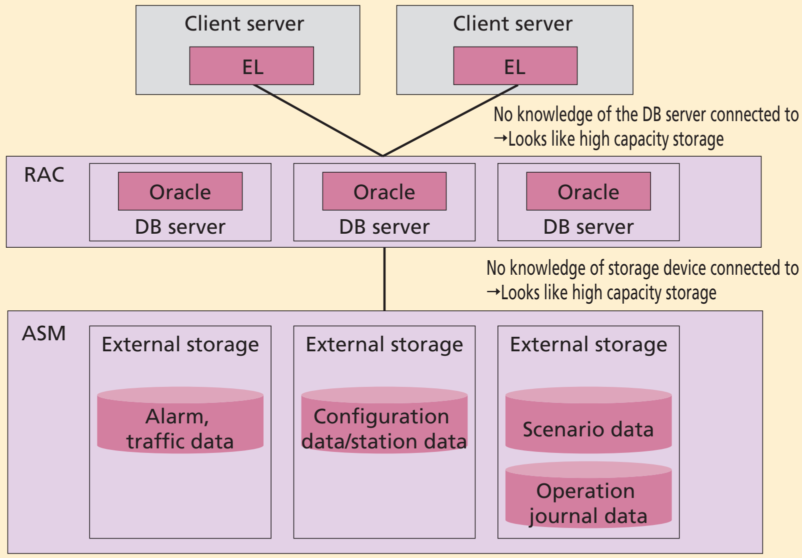


Figure 8 DB virtualization

States.

*

1

1

D3A: An architecture developed at NTT DOCOMO, which groups multiple IA servers to achieve high performance. IA servers use microprocessors from Intel Corp., which are less expensive than servers using microprocessors from other manufacturers. This allows low-cost, high-performance systems to be built easily.

between the applications and D3A (D3A wrappers), creating the structure shown in **Figure 9**. This allowed us to use existing applications with D3A.

We also used Work Management System (WMS)[*16] [7], which is used for node NW-OPS, as the work scenario engine. WMS is used in node NW-OPS mainly for updating switch files automatically, and this automated work is implemented by describing it with scenarios. To develop the work scenarios used by WMS, we developed tools to convert JRules work scenarios to WMS work scenarios, and conversion functions for JRules and WMS. This allows us to use the work scenarios from the previous Knowledge System.

We also needed a flow editor, as was provided earlier for JRules, to create work scenarios, so we adopted the Open Source Eclipse™[*17] and Java[*18] Workflow Tooling (JWT)[*19], allowing us to create work flows by dragging and dropping work scenario elements. An overview of work scenario creation is shown in **Figure 10**.

Developing the D3A wrappers, JRules and WMS conversion functions, and work scenario conversion tools allowed us to reuse applications, and to promote introduction of NTT DOCOMO products, D3A and WMS, as well as Open Source software.

3.4 Further Improvements Using the New Architecture

Using the new Knowledge System architecture, we have been able to reduce changes to adapters (hereinafter referred to as "ADP") controlling interfaces with other systems (Network Element-OSS (NE-OSS)[*20]) when adding new equipment.

The basic interface elements are sequence control and data conversion. We have enabled calls to these sequence control and data conversion tables (StyleSheets[*21]) to be implemented in work scenarios. By applying the WMS engine mentioned earlier in an ADP, we enabled new equipment to be

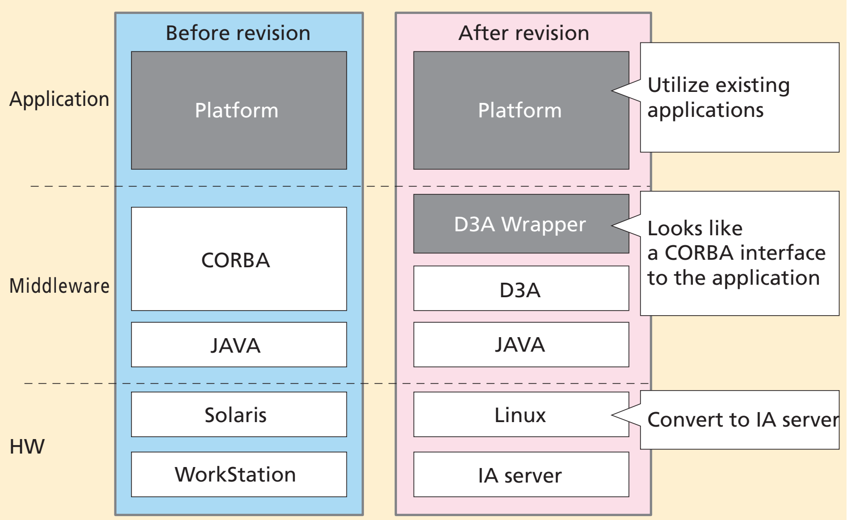


Figure 9 Middleware replacement

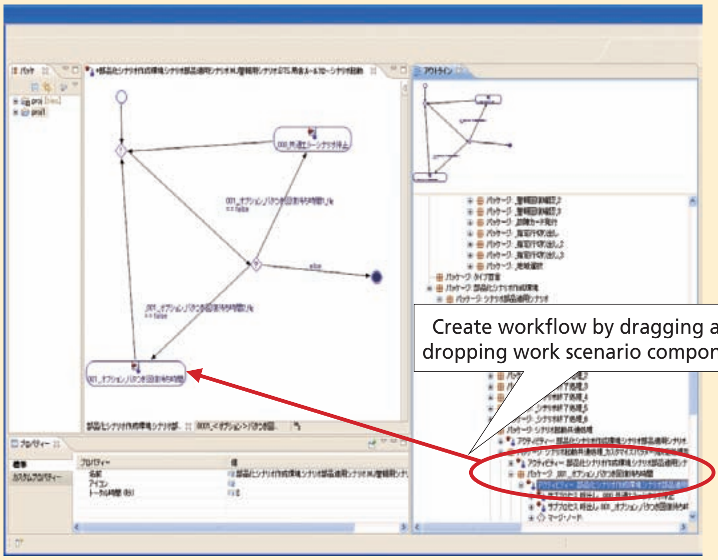


Figure 10 Work scenario creation

*12 **Adobe® Flex™**: An integrated development environment and software development kit for rich Internet applications, from Adobe Systems INC. A registered trademark of Adobe Systems INC., United States.

*13 **Oracle RAC**: Software for DB redundancy from Oracle Corp. Multiple Oracle DBs are run on separate servers, with all DBs sharing a single data file, so even if a fault occurs, there is almost no degradation in the system.

*14 **Oracle ASM**: Software for implementing a supported by changing only the work scenarios and StyleSheets, making development much easier.

4. Conclusion

We have described the introduction of a new architecture in the renewal of our Knowledge System. The new Knowledge System began full operation in January, 2011. In the changes, we were able to reduce installation and running costs, and we were also able to further expand the support available for maintenance work by expanding the scope of customizations possible by maintenance persons. In the future, we will confirm the effects of these changes, and study horizontal expansion of this customization architecture to other systems.

REFERENCES

[1] S. Oyamada et al.: "A Network Operation Knowledge System," NTT DOCOMO Technical Journal, Vol. 12, No. 2, pp. 43-50, Jul. 2004 (In Japanese).

[2] S. Oyamada, M. Furutani, T. Ohno and N. Tanigawa: "A Study on Function Allocation method for Network Operation Knowledge System," IEICE technical report, NS, Vol. 102, No. 348, pp. 99-104, Sep. 2002 (In Japanese).

[3] S. Oyamada, T. Tahara, M. Furutani and T. Ohno: "A Study on Operation Scenario Creation Environment for Network Operation Knowledge System," IEICE technical report, NS, Vol. 103, No. 442, pp. 103-106, Nov. 2003 (In Japanese).

[4] S. Manako, T. Ohno, S. Oyamada and N. Tanigawa: "A Study on efficient Journal-Data Collection for Network Operation Knowledge System," Proc. of the IEICE General Conference, B-6-38, Mar. 2004 (In Japanese).

[5] K. Fuzii, T. Watanabe, K. Akiyama, K. Takahashi and N. Tanigawa: "An implementation and performance evaluation of the Distributed Data Driven Architecture," Technical report of IEICE, TM 103(43), pp. 1-6, May 2003 (In Japanese).

[6] K. Akiyama et al.: "Economizing Technology for Operation System—Distributed Data Driven Architectures—," NTT DOCOMO Technical Journal, Vol. 13, No. 2, pp. 36-46, Jul. 2005 (In Japanese).

storage grid by managing multiple disks as a virtual disk group.

*15 **NW-OPS**: A system which conducts maintenance monitoring for equipment in the NTT DOCOMO network.

*16 **WMS**: One of NTT DOCOMO's OSS systems, which automates network management work.

*