POSCO Operation System (MES) Project Permalink (My Space)

October 14, 2005 9:39

 https://blog.naver.com/h\_tiger/20018068997

View exclusive viewer for translation

Creation of the largest J2EE myth in Korea: POSCO Operation System (MES) Project

Park Jung-bae | Korea Oracle

POSCO, which has introduced a ERP-based enterprise-wide integrated package and successfully built a corporate portal, has made a brilliant management system innovation.

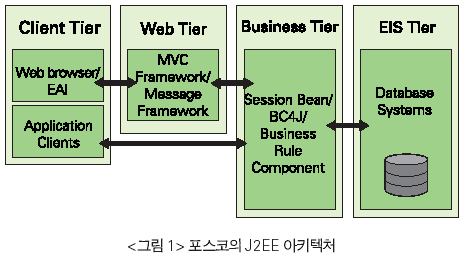
Confident that the introduction of leading IT technology is the basis of future competitiveness, POSCO has been able to operate two steel mills in Gwangyang and Pohang under a single management system as a standardized process through the newly constructed operating system. It is also possible to innovate not only the management system but also the production system.

Guide the reader to the field of POSCO mythology creation.

The need for a new system to replace legacy

POSCO's existing Manufacturing Execution System (MES) was built as a mainframe, but maintenance costs continued to increase due to equipment obsolescence and lack of professional personnel. This was not only the equipment but the software as well. The applications of Gwangyang and Pohang Works were not compatible with the newly added system. As new systems continue to be added, such as the Stainless Steel Factory System, it has come to an endless conclusion that keeping the system on both hardware and software is uneconomical. POSCO has invested heavily in the system since the beginning of the project, and decided that it is time to replace the system. As a result of the success of PI 1, POSCO entered into the construction of a new operating system as part of the PI 2 project .

The newly constructed operating system was very uncommon in reversing the existing one in both hardware and software. The hardware used to be a supercomputer-class Unix device, with the mainframe being discarded, and the software was built with Oracle's application server and database and a Java-based J2EE enterprise application.



Concerns about such an uncomfortable system replacement were also a concern, but the biggest concern was that Java was not as good as a performance system due to poor performance. However, this is an indication of the performance bottleneck that early Java showed, and recent Java shows stable performance under a robust and stable architecture based on J2EE.

The problem was not an architecture based on Java or J2EE, but how to implement the operating system. Oracle contributed to building a successful operating system by providing a framework for development of the operating system.

Introducing a robust application development framework

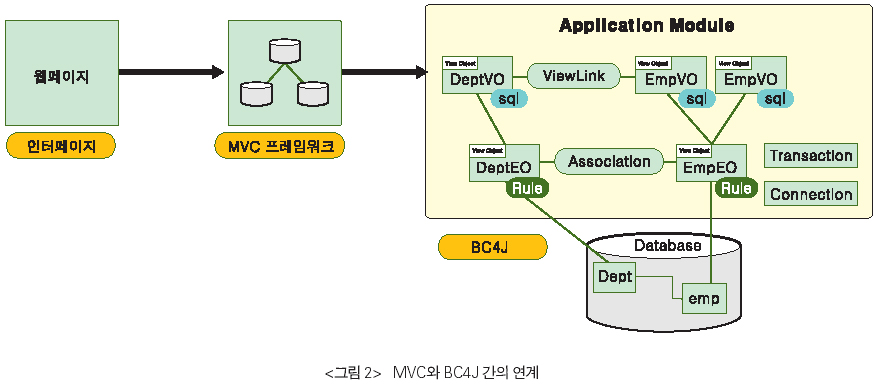
The operating system receives data from the plant's facilities, processes them, directs new work, and so on. The problem arises from the diversity of work that each factory has. Because each plant has different facilities, different processes, and different contents of work, packaged software has a limit to meet the requirements of the system. In the end, POSCO decided to implement the operation system directly.

POSCO's choice is based on component-based development (CBD), which consists of building core modules into components, which requires a strong development framework.

POSCO has invested heavily in reviewing all the technologies available for enterprise application development to build a production system. Initially, a lot of efforts were made to build applications using "EJB (Enterprise Java Beans)" technology, which is called "J2EE based application". However, EJB was not a suitable technology for many operating systems that had to deal with large amounts of data in conjunction with a database, or to perform complex business logic. A more powerful and easy-to-use application development framework for POSCO's operating systems was more appropriate. Here are some of Oracle's many powerful application development frameworks.

Oracle is not only a database company, it is also a business application company. Oracle's business applications are built primarily on the Oracle Application Framework framework, which provides reliable functionality for handling large amounts of data and supporting a large number of users. Among the various framework functions provided by Oracle Application Framework, BC4J, which is used for database interworking, is the best solution for its function, performance, and ease of use. The POSCO operating system used this BC4J in large quantities of data processing tasks and business logic components.

In addition, a web application development framework was also used. The user interface of the POSCO operation system is basically based on the web, so that the system can be easily accessed and operated. Oracle's Web application development framework, Oracle's MVC Framework (Oracle's product name, not the framework of the general MVC pattern), was used to develop the Web-based user interface. The Oracle MVC Framework is well-suited to modeling tools, allowing you to define the flow of user screens directly through the model, which is very useful for developing web-based applications with many user screens. <Figure 2>

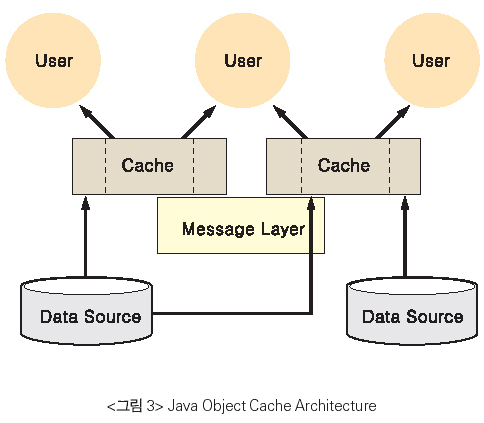


ERP technology was also used. POSCO's operating system is based on the need to process real-time data directly, but there are many things that need to be done in batches. The processing of these batches was done using Concurrent Manager in Oracle E-business Suite, which enabled stable deployment management and execution. Oracle Concurrent Manager, which is a main component of Oracle E-business Suite, is an entirely self-contained application with its own job execution engine, scheduling engine, HA functionality, and monitoring capabilities.

In the early days of the project, Oracle recommended the use of the Oracle Concurrent Manager, and decided to use it as part of the operating system for economic and stability reasons, in order to perform many applications requiring automatic and cyclical performance of operations. The use of Concurrent Manager as a batch processing system to drive a large scale ERP has brought economic and stability to the operating system. The throughput of Oracle Concurrent Manager processed by EAI and scheduling is expected to be more than 60% of the total TC.

The operating system is very strict in terms of performance requirements and there has been considerable consideration from the architectural stage to achieve its performance. POSCO's development system links MVC with BC4J to store the key reference data, rule data, and metadata used in the system in the database and query each time it is needed, There is a drawback that it is difficult to achieve good performance of the system inevitably.

To solve this problem, we used an object cache mechanism called Java Object Cache (JOC). The JOC provides the ability to store objects in the cache to enable applications to use the data they need more quickly. POSCO's operating system is configured to store the reference data, rule data, and metadata in the cache of the JOC so that it can be used for faster performance. <Figure 3>



These four frameworks are the foundation for Oracle applications, and are especially useful for developing applications based on Java. In addition, these frameworks can be used extensively across multiple business domains. However, in POSCO's operation system project, we do not just use these frameworks, we go a step further and develop components that can be operated on these frameworks, I extended the work.

Development of column and reuse component of operating system

Oracle's framework was very good in itself as an infrastructure for developing applications, but POSCO's operating system was large and there were a lot of people involved in the development, so we needed a way to make it easier to use. The methods used in the POSCO Operation System project are to develop a component that performs common functions and to define the detailed functions of each reusable component as metadata so that application development is possible only by modeling Respectively. It is a way to develop on the basis of reusable components, making it easier to develop applications.

More specifically, it implements components that perform common, common, and frequently occurring tasks such as data retrieval, data storage, and branching through comparison of values. When data is actually used when each component operates, The data that determines the actual function is extracted and managed by meta data, such as whether to look up, what value to store, and which values ​​to compare and branch to. By simply arranging the reusable components and metadata modeling, .

Since the Java used to develop POSCO's operating system was not easy to learn and use by developers who were familiar with the existing operation system and there were many people involved in the development, It was a necessary process.

The Oracle consulting team has developed a reusable component with POSDATA. The Oracle consulting team provided methodologies and ideas for designing and developing each reusable component, and POSDATA implemented it.

The reusable components that make up the POSCO operation system are divided into the following four categories.

● Application logic

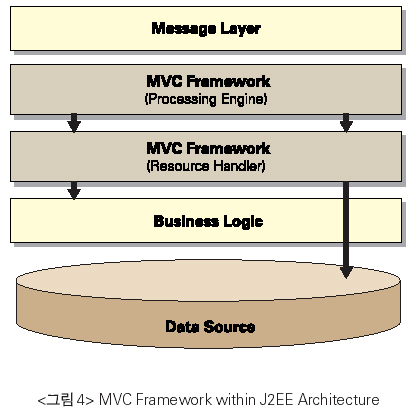
● Business logic

● User Interface

● Non-functional

The core of POSCO's operation system - application logic component

The application logic component handles the work flow of the operating system. The application logic component is the backbone of the POSCO operation system. It is largely divided into the non-UI that handles the user interface (UI) and the other part. The UI component is concerned with the web-based user interface and operates on the Oracle MVC Framework. The underlying structure of the Oracle MVC Framework is not much different from Struts and other Model-2 based web application development frameworks. The UI component handles tasks such as data retrieval, data storage, specialized writing, and expert transfer that are typically performed by the operating system on the Oracle MVC Framework. POSCO's operating system is automated, so there is not much work that needs to be handled directly by the user, but the system has been configured to enable control through the UI as a similar backup system.



While the UI component does work with the user interface, the Non-UI component is used to receive and process the data sent by the operating facility through the EAI. The data sent by the operating facilities are presented in the form of standardized professionalizations, and the standardization of these specializations was one of the most important tasks in the construction of the operating system. The POSCO operating system project team created and used the framework for the implementation of the non-UI component parts. This framework, called the "message framework," is not significantly different from the Oracle MVC Framework in its concept, but it is designed to meet the requirements of handling a large number of specializations. On top of this framework, we have created components that handle data retrieval, data storage, professional writing, and specialized transmission, as well as UI components.

Application logic components play a very important role as the backbone of the POSCO operation system. Hundreds of user commands and specializations per second are handled by application logic components and other reusable components are used around application logic components. POSCO's operating system implements most of its functions with business activity modeling (BAM) and metadata settings based on application logic components. This is possible because of the powerful framework called BC4J and Oracle MVC Framework.

Business logic components for business rule processing

The business logic component is divided into a business rule component that contains the business rules of the POSCO operation system and a business logic interface component that acts as an interface for manipulating the data by accessing the database at the application logic level. Both of these components were implemented by extending the basic functionality of BC4J.

The Business Rule Component is a set of business rules that exist for each process of POSCO's operating system through a system called 'entity object' of BC4J. Business systems are able to manage business rules in a consistent and effective manner through business rule components. Business rules gathered through business rule components are continuously managed and improved, making the system more stable over time.

The business logic interface component defines the portion of the application that accesses the database. The POSCO operating system can only access and retrieve data from the database through the business logic interface. The POSCO operating system defines a standard for data querying and modification with the business logic interface, and the application logic component uses the business logic interface according to the values ​​defined in the metadata. Since the POSCO operation system handles a large amount of data, it uses a lot of databases. By controlling the part that accesses the database through the business logic interface, it is possible to build the application more reliably.

User interface with tag library components

All areas of the POSCO operation system were based on the development methodology, adhered to most of the development life cycle, and the user interface was the same. In the early stage of development, user needs were accommodated and templateization work was performed on the screens desired by the user. Especially, it is possible to improve the productivity of developers and unify user interface by making JSP tag library centered on the components of user screen that need to be reused, especially in the area where data is rendered in JSP screen. This was a natural process for large projects with hundreds of developers in the project, and could be a requirement for project success.

The non-functional component, which is the assistant of the system,

While not a core part of the POSCO operation system, it was also important to implement utility functions common to all systems. In the POSCO system, these key items were defined as non-functional components and developed as reusable components. These include messaging / logging / error handling, security, and so on.

Non-functional components such as messaging, logging, and error handling helped to facilitate debugging and monitoring during the development phase, and allowed error handling in source code to be performed quickly. It is also designed to help operators speed up business and data consistency checks in the future.

One of the important parts of the operational system is the security related function in the operation system such as user management and authority management. The main users who use screens developed in the operating system are the workers of POSCO and Gwangyang Yang Works and it is not desirable to make all screens available to more than 15,000 users. Each worker should be able to access only certain screens according to his / her given task and authority, and should be classified into subdivided authority even in the screen for simple inquiry and input / modify authority. The basic model of the user and authority management of the operation system is the User Responsibility of the Oracle E-Business Suite. The user has implemented a menu system that allows individual access (classified as Responsibility, Menu, Function). The part corresponding to the authority management is owned by the operating system itself and is linked to the existing POSCO Enterprise Portal system.

The record of POSCO operation system

• Number of programs: 6,500

• Program line: 2 million lines (360 pages of 300 books)

• Data throughput per day: 2 gigabytes (36 pages for 2 years and 6 months for daily newspapers)

• Number of projects: 15,999M / M

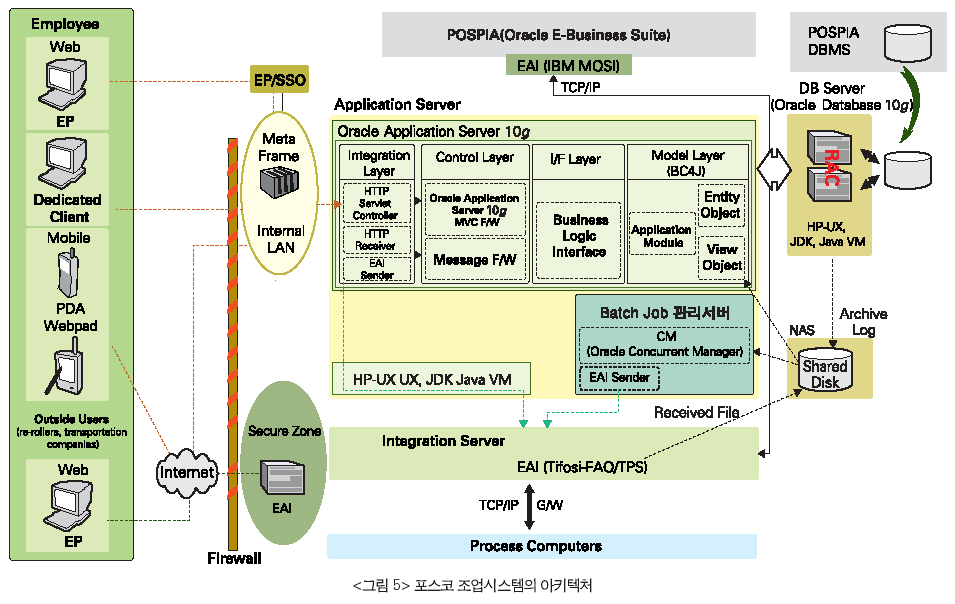
Master data / metadata that became the invisible basis of the system

Standardization and integration are basic ideas for the operation of the POSCO operation system, which has an annual production capacity of 28 million tons of crude steel. We have received orders for work from the already established Oracle EBusiness Suite System (POSPIA), numerous process computers that command production lines at dozens of factories in Pohang and Gwangyang, production processing tasks, The amount of information that POSCO's operating system spills is enormous, including logistics processing information such as shipments and yards. In order to process this huge amount of information accurately and within a time that will not interfere with the production of crude steel, we have established a standardized and integrated management system for master codes, business judgment criteria, calculation formulas, and interface error check standards. This was called "master data" in POSCO. In addition, the reusable components developed for the operation system were designed to manage the data required for business processing with metadata in order to improve development productivity and improve operability in the future.

These master data and metadata use the Java Object Cache (JOC) mechanism to improve performance and serve as a basis for moving the POSCO's operating system silently.

Architecture of the operating system

The architecture of the POSCO operation system is shown in Figure 5.



The middle tier on which the operating system is based

Oracle's middle tier server, Oracle Application Server 10g, provides mid-tier HA with work system applications. On the Web, where the HTTP protocol is the foundation, high availability can not be achieved by the application server alone, and is accomplished by borrowing a variety of session related technologies from the application. Oracle Application Server 10g provides cluster functionality, session replication (session replication) of OC4J, and object synchronization between JVMs through the Java Object Cache (JOC) library. This provides a complete infrastructure that meets the demanding high availability requirements of the POSCO operation system.

cluster

The basis of high availability is 'multiplexing'. On the basis of 'multiplexing', functions such as session sharing, object sharing and load balancing algorithm are drawn. In an enterprise environment, multiplexing has two main meanings: synchronization of resources and non - stop service through failover between multiplexing entities.

Oracle Application Server 10g provides a file-based cluster approach in addition to the database-based cluster approach provided by Oracle9i Application Server. File-based clusters are managed by WAS resources based on a small file database. The POSCO operating system borrows this method.

This is a new feature in Oracle Application Server 10g that reduces the need to manage the database for resource synchronization purposes, saves hardware, and saves time by reducing operation time.

Various algorithms are provided for load balancing between cluster objects. In a production system, the local affinity setting (a setting in which local OC4J is given priority when routing from OHS to OC4J) It is suitable for transaction processing. This is a very useful algorithm especially due to the characteristics of the operation, and it contributes a lot to the performance. For the user screen, there is no other consideration because the servlet session is preferred over the local affinity.

Duplicate Session Between OC4J

This feature is a feature to share sessions between JVMs and is configured in OC4J.

As you can see in Figure 5, the application in the production system uses the Oracle Application Server 10g MVC Framework as the controller part, which is based on the servlet session. It is assumed that the size of the session used in the framework is not enough to perform the session replication, so the operation system does not use it. Instead, the business system application is designed to reconfigure the session using cookie and HTML form parameters, so a complete failover is implemented from the service perspective.

Java Object Cache

Instead of using session replication, the production system application uses JOC (Java Object Cache) libraries to realize object synchronization between JVMs, where synchronization between metadata and master data is required. In addition to significantly reducing database traffic, the JOC also facilitates application design that meets the hardware by performing object synchronization between the JVMs.

In this way, the middle tier of the operating system combines the infrastructural features provided by Oracle Application Server 10g with the web-based technologies to provide the most important business requirements of the operating system without any doubt.

The choice of the middle tier of the POSCO operating system to enable the deployment of applications in business continuity and fault tolerance is proving its function in today's moving systems.

Significance and Expected Effect of POSCO Operation System

POSCO presented a new system that replaces the mainframe by rebuilding its existing mainframe-based operational system into a new enterprise application system based on application servers and J2EE. There have been many J2EE projects in the meantime, but there have been no large-scale projects like POSCO's operation system building project. It is more meaningful because it builds the operation system that controls and manages the production facilities.

Most of the time, when talking about a J2EE system, a service-oriented system that is not the center of a periodic task is often set as the target area. In the early days, when POSCO selected J2EE as the base technology of the production system, it became a problem that it was rarely used as the time system, that is, the production system as J2EE. However, the 3-year project, which has been carried out with the confidence that it is possible to implement the new paradigm, is now completed with the first Iteration of Pohang / Gwangyang operation followed by the 2nd Gwangyang operation starting on November 23 . Now we have left only the second main operation of Pohang.

Through this, it is expected that not only economical benefits but also time and personnel savings are achieved by using existing technology elements to expand production facilities or introduce new jobs. The POSCO operation system has excellent value, not only in economic terms but also in the symbolic sense. The production system known up to now is the maximum size J2EE, which is not only the number of people participating in the project, but also the area of ​​work that a single system is developed and covered all at once.

POSCO's operation system construction will be remembered as a solution to enterprise systems and whether J2EE can be applied to large-scale operation systems as a real e-business completion from production facility system to management information system as a whole.