A case study of development of the transportation plan and the operation control system for a high-speed railway using a relational database

T. Nakamura[1], H. Shimizu[1], H. Suzuki[2], S. Mogi[2] & S. Hasegawa[3]
[1]*JR East Japan Information Systems Company, Japan*
[2]*East Japan Railway Company, Japan*
[3]*NS Solutions Corporation, Japan*

Abstract

We renewed the transportation planning and the operation control systems for the high-speed railways of the East Japan Railway (JRE) using a relational database management architecture in 2008. The JRE has five bullet train lines (SHINKANSEN). It has an information system named COSMOS to control and operate SHINKANSEN properly. The transportation planning sub-system has six function groups which are developed by different manufacturing companies. We decided to adopt an integrated relational database model and database interfaces in order to restrain consistency failure of database transactions. We chose NS solutions (NSSOL) as the manufacturing company for database modeling and interface development, since NSSOL has deep knowledge and experiences to construct a database model for railways. We and NSSOL designed and developed the database that has about 1400 tables which contain several hundred gigabytes of data. Since we developed the database on our own, we are able to understand the whole structure of the database CRUD (Create, Read, Update or Delete data on database) and to fix failures in the architecture.
Keywords: train traffic scheduling, real-time rescheduling, relational database.

1 Introduction

The East Japan Railway Company (JRE) has five bullet train (SHINKANSEN) lines (Tohoku, Joetsu, Hokuriku, Yamagata, and Akita). The network map of JRE

WIT Transactions on The Built Environment, Vol 135,
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR140481

is shown in Figure 1. Table 1 shows the characteristics of each line. The five SHINKANSEN lines link five areas in East Japan originating in Tokyo.

A part of the Yamagata and Akita lines is a lower speed conventional area and not only the bullet train but also conventional are operated in the area.

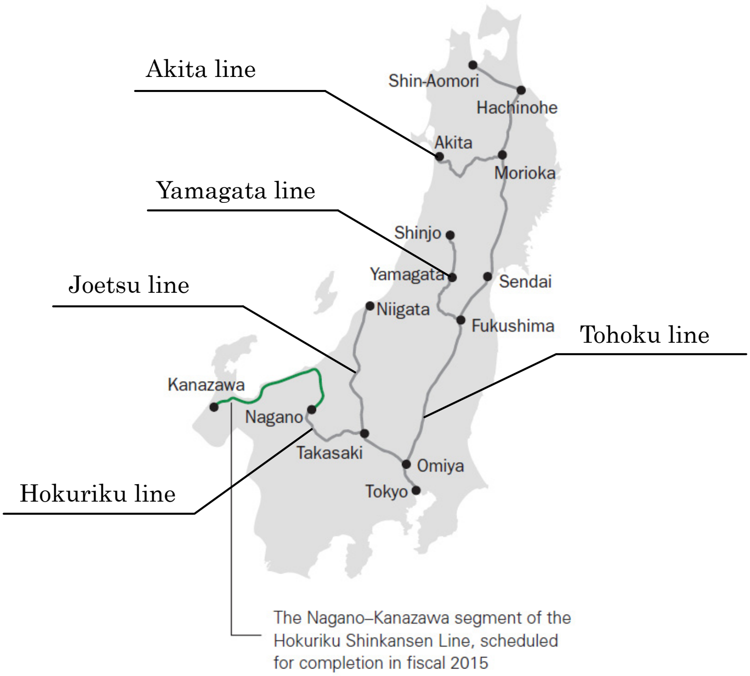

Figure 1: SHINKANSEN network in JRE [1].

Table 1: Characteristics of each route of JRE.

Route name	Route kilometers	Nickname of vehicle	Operating number per day
Tohoku SHINKANSEN	631.9km Tokyo – Shin-Aomori	Hayabusa Yamabiko Hayate Nasuno	About 180
Joetsu SHINKANSEN	333.9km Tokyo – Niigata	Toki Tanigawa	About 110
Nagano SHINKANSEN	222.4km Tokyo – Nagano	Asama	About 70
Yamagata SHINKANSEN	421.4km Tokyo – Yamagata/Shinjo	Tsubasa	About 40
Akita SHINKANSEN	662.6km Tokyo – Akita	Komachi	About 40

www.witpress.com, ISSN 1743-3509 (on-line)

Yamagata SHINKANSEN run merged with Tohoku SHINKANSEN between Tokyo and Fukushima. Akita SHINKANSEN also run merged with Tohoku SHINKANSEN between Tokyo and Morioka.

The JRE owns various types of the SHINKANSEN vehicle in order to meet diverse passenger needs. I show the characteristics of each vehicle as of April 2010 in Table 1 and the number of trains, vehicles and crew operations per day in Table 2 in order to show the transport scale of SHINKANSEN.

Table 2: The various overview of SHINKANSEN diagram of JRE.

Number of trains	about 1000
Number of vehicle operations	about 200
Number of driver path	about 250
Number of crew path	about 350

2 Characteristics of SHINKANSEN transport in JRE

JRE has five SHINKANSEN lines and operates about 800 trains per day. It dispatches nearly 200 rolling stock, 250 drivers and 350 conductors per day in order to operate the trains. Expansion of the SHINKANSEN network, such as new stations opening and enlarging is still going on and trains run faster and consequently the number of trains is going to increase. As a result, a wide variety of vehicle organization appears and the transport system becomes complicated and diverse. Since the trains are operated in multiple lines in a high-density diagram, there is a possibility that a transport failure occurring on one route affects the others.

Therefore, it is necessary to perform an accurate transmission of information to customers at the time of the transport confusion, and optimize operation management.

3 COSMOS overview

3.1 Overall

As mentioned above, it is necessary to manage the SHINKANSEN transport accurately and rapidly in JRE. Therefore, we have built a system to support human decisions. It is named COSMOS (Computerized Safety Maintenance and Operation systems of SHINKANSEN). COSMOS has eight sub-systems [4] as shown in Figure 2: Transportation Plans Sub-System; Operation Control Sub-System; Rolling Stock Control Sub-System; Railway Yard Work Management Sub-System; Facility Monitoring Sub-System; Maintenance Work Control Sub-System; Electric Power Control Sub-System; Facility Control Sub-System. By close cooperation with them (eight systems), we realize an advanced information management and control. The sub-system overview is shown in Figure 2.

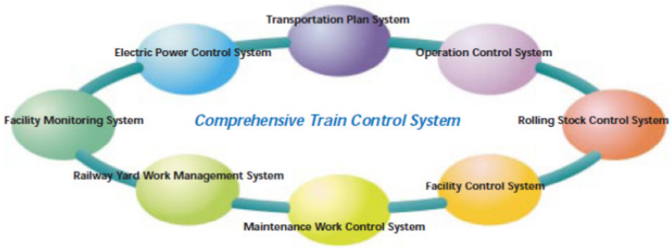

Figure 2: The COSMOS sub-system overview.

3.2 COSMOS replacement overview

COSMOS began to be used in 1995. At its 12-years of useful life in 2007, replacement of aging equipment was required. Then, we decided to adopt the latest technology in software development at the same time as the obsolescent hardware was replaced in the transportation plan sub-system and operation control sub-system of COSMOS. We have to involve new and improved functionalities requested by our users as much as possible. In addition, we abolished the host system architecture and we adopted Windows architecture across the board. We determined the multi-vendor development by fitting a vendor with a good field of each to realize new features and this was built by the latest technology.

3.3 Database development overview

When we renewed COSMOS, we adopted relational database management architecture [3] in the transportation plan sub-system and the operation control sub-system. The transportation plan sub-system has six groups of functions – train timetable planning, driver and conductor scheduling, rolling stock scheduling, rolling stock inspection scheduling, instruction paper delivery for train operation and printing the timetable diagram. To obtain an appropriate quality and reduce cost, each function is developed by a different manufacturing company. Data structures and data access method to be used for each function are developed in a different concept for each vendor with the application.

Data to be used by COSMOS are used with each other in multiple transportation plan sub-systems. They are used by each other in the transportation plan sub-system and the operation control sub-system. For example, driver and crew scheduling and rolling stock scheduling are created based on the data that were created in the train timetable planning. Previously, we had managed in duplicate for each application with similar meanings. Therefore, if the failure that is caused by the data of other function occurs, it is difficult to make a survey. In addition, since the design concept is different for each vendor, it is difficult to see the impact of each other. Therefore, it was our desire to manage the multiple influence of the entire database [2].

So, we decided to develop a database of train planning and operation control data that are integrated to work with all functions. Access to the database is done directly from the application using the database interface. As a result, a database interface failure did not affect the critical application. This is because of a separation of the data layer and the application layer.

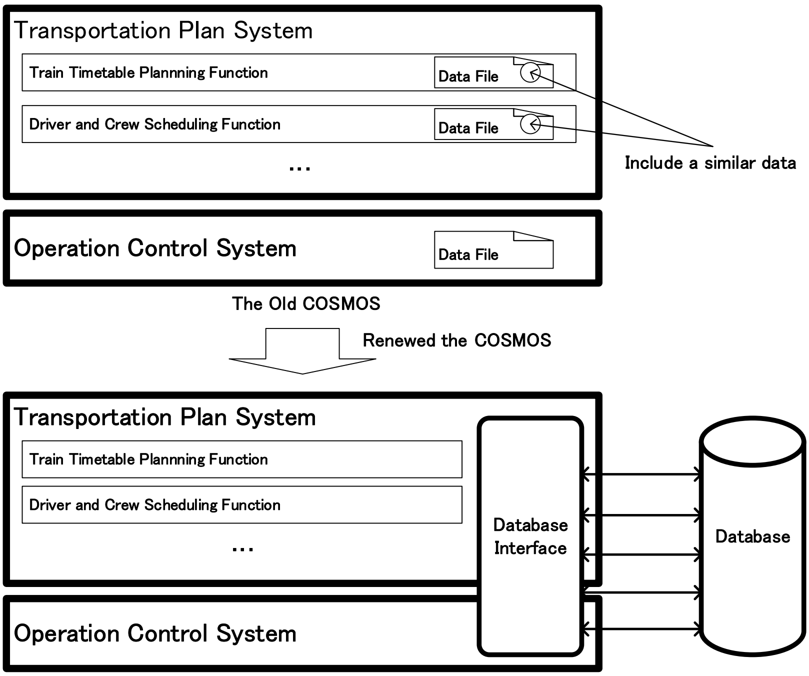

Figure 3: Cooperation of each sub-system and database.

3.4 Database development system

As described above, in order to build a database that was what we required, it was necessary to design a unified database design concept. However, we did not have experience in the design and development of the database. We chose NS solutions (NSSOL) as the manufacturing company for database modeling and interface development. It had a track record of building a train schedule database for the West Japan Railway Company. Therefore, we thought that it would be able to realize the design principles of the database structure in our company. As a developing company of a database interface module, we also chose the same company as our partner company because we wanted to establish common parts and improve the quality of the database interface module.

4 Case study

4.1 Survey and measure of the degradation of application performance

If degradation of application performance occurs, it often tends to go along with data access. If the data are accessed only by a single function, in most cases, problems are due to the fact that the amount of data and load status of the execution are large. Therefore, it is easy to find the cause. In the case of multiple data access function, a survey is difficult. In particular, if the data access from other functions raises performance degradation, t makes research difficult. The trace log of the database management system can be used to identify the SQL statements that are executed in a time zone when the problem occurred. However, it is difficult to determine which function executed the SQL. Having constructed the unified database, we are able to understand the structure of the CRUD (Create, Read, Update or Delete data on database). Firstly, we refined the database interface so that it can access the target table from the CRUD structure table. By confirming that the specifications of the database interface are narrowed down, it is possible to identify the database interface that issued the executed SQL statements. When we know the CRUD structure, it is possible to carry out the surveys mentioned above. As a result, we reduced the overhead of the investigation request to multiple vendors. In addition, we have been able to make a rapid failure investigation.

When we change data structures or database interfaces as an action to the problems found by the investigation, we can quickly determine the effect on other database interfaces by knowing the CRUD structure. In addition, we have a unified data access design concept by the database interface in COSMOS. Therefore, we are able to prevent deadlocks caused by changes of the data access order before it occurs.

4.2 Updating a database version

We had to upgrade the commercial database management system because of a bug that we found in 2010. There is a possibility that the database system is stopped if the problem occurs; this was the only measure. However, there is a possibility that the behaviour of the database interface and application change. Therefore, we had to consider the impact carefully and perform sufficient validation before upgrading the production environment. Firstly, we investigated the effect on the database interface and application due to the upgrading. If we manage the database by dividing each application, we had to request all application vendors to analyze the influence on their application. Because of the difference in design philosophy, there is a possibility that a difference occurs in the degree of influence.

Because databases were built to be unified, we were able to do cross-cutting impact studies on common changes in a short time. In addition, by understanding the CRUD structure, it has become possible to confirm the performance effects caused by the upgrade even on the data table that is accessed by multiple functions. As a result, we were able to upgrade the database system in a production environment successfully.

5 Conclusion

We accomplished the design and development of the database that has about 1400 tables and several hundred gigabytes of data and is used by the transportation plan and operation control system in the SHINKANSEN of JRE. We were able to develop several thousand database interfaces to access the database. Since we have developed the database on our own, we were able to understand the structure of the database CRUD. As a result, when a failure occurs in the system, we can restore and examine the failure quickly. In addition, having created centralized material for the failure investigation, we were able to use it in common. Thus, we were able to reduce the cost of the failure investigation.

As an issue, even when a local problem is caused by only one function, it is necessary to confirm the effects on the entire database. If changes of the data structure occur, we need to know the effects and inform all personnel. This is because this database is used in a unified manner with multiple functions. In addition, we need to train the database administrator to understand the configuration of a huge database and to be able to determine the necessary actions.

References

[1] East Japan Railway Company "2012 Annual Report", http://www.jreast.co.jp/e/investor/ar/2012/index.html
[2] T. Fujita and M. Arisawa: Train information system based on the database, Technical Report of IEICE, SS99(72), 23-28, 1999
[3] H. Fukuda: Development of Human Error Database System for Railway Transportation, Information Processing Society of Japan, 403-404, 1995
[4] K. Sasaki: New Shinkansen Operation Management and Control Systems for JR-East Shinkansen Network, Reliability Engineering Association of Japan, 34-42, 2000