Featured Articles

Techniques for Migration and Refactoring (Downsizing) of Application Assets

Atsushi Awakawa
Katsuya Tokuda
Toshimitsu Shibayama
Takuro Matsuzawa

OVERVIEW: In formulating its e-Japan Strategy in 2001, the Japanese government introduced policies covering such matters as the establishment of IT platforms, the use of IT in government, and the promotion of e-government. Furthermore, a 2002 report by the Board of Audit of Japan about the problem of legacy system procurement began moves toward system reform in government information systems and elsewhere through the migration of legacy systems running on mainframes to open systems. Hitachi Government & Public Sector Systems, Ltd. launched its migration service for application assets in March 2002. Hitachi's migration service transforms application assets from existing systems so that they are able to run on a new architecture. It has undertaken more than 160 such projects to date (totaling approximately 110 megasteps). Along with migration, the service also performs "refactoring" (downsizing), which in this context means reorganizing application assets and reducing their size. This refactoring process was added as an official service option in June 2015. In the future, Hitachi intends to continue contributing to system reform by providing services with more advanced functions and higher quality.

INTRODUCTION

Background to Migration

IN formulating its e-Japan Strategy in 2001, the Japanese government introduced policies covering such matters as the establishment of information technology (IT) platforms, the use of IT in government, and the promotion of electronic government (e-government). The progress of IT became increasingly rapid from around this time, with a steady stream of new hardware and architectures being released (see Fig. 1).

A 2002 report by the Board of Audit of Japan about the problem of legacy system procurement in government information systems belonging to public agencies began moves toward system reform through the migration of legacy systems running on mainframes to open systems.

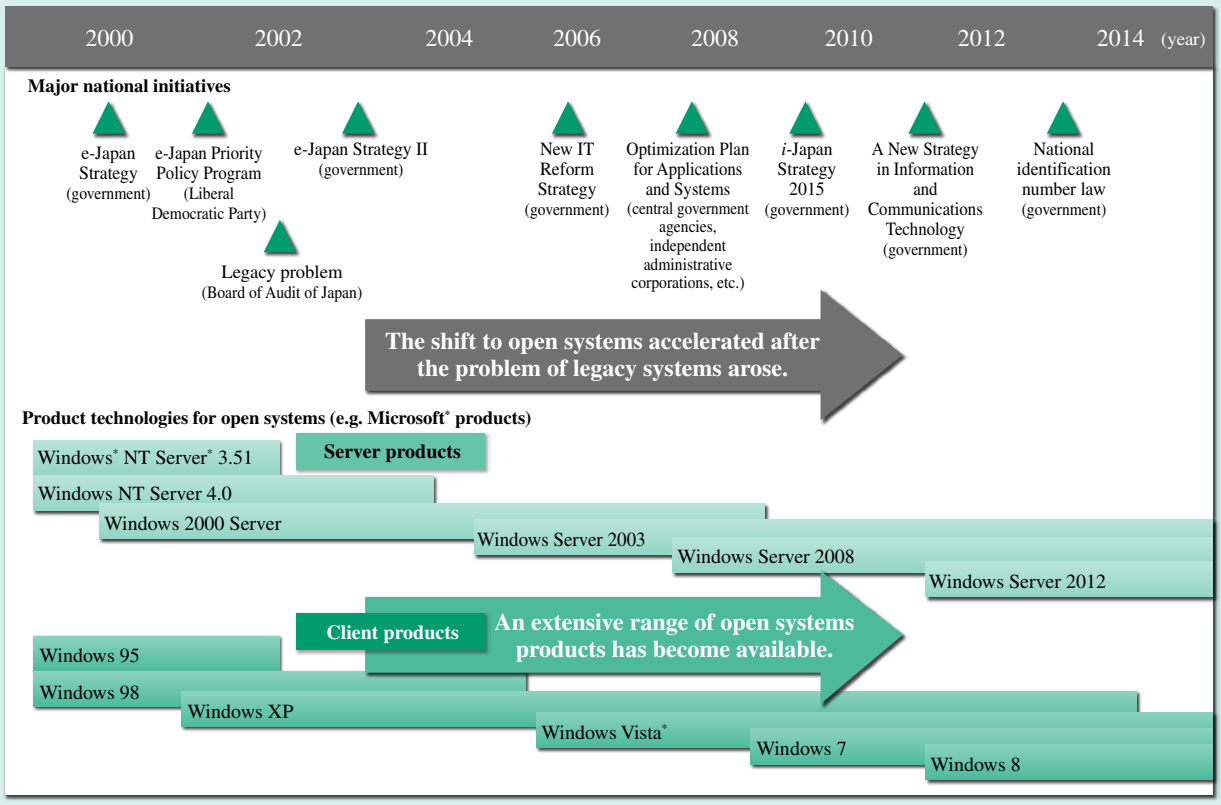
In response to these moves toward system reform through the adoption of open systems, Hitachi launched a migration service for application assets in March 2002 that has undertaken more than 160 projects to date (totaling approximately 110 megasteps).

Whereas the tendency when the service was first introduced was for the adoption of open systems to be prompted by the expiry of mainframe maintenance contracts, projects of this nature have tailed off in recent years. Nevertheless, the reform process for government information systems belonging to public agencies remains incomplete with large systems running on mainframes still in existence. There are also systems that have been through the reform process but need to be migrated to a common government platform that commenced operation in 2013. Consequently, it is anticipated that demand for migration will grow in the future.

Background to Refactoring (Downsizing)

As systems undergo numerous modifications over the course of their operating lives, there are many cases in which application assets have ballooned in size and grown complex. Because performing maintenance on such systems can have widespread implications, the cost tends to increase. Hitachi is frequently consulted by customers adopting open systems who want to downsize programs and minimize maintenance costs, and has undertaken refactoring (downsizing) projects involving the reorganization of application assets and reducing their size.

Hitachi has consolidated this know-how and added refactoring as a service option in June 2015. The



IT: information technology
* Microsoft, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Fig. 1—Trends in Open Systems.
The adoption of open systems has been accelerating since 2002, with a comprehensive range of open system products now available.

service reorganizes application assets by identifying programs that are unnecessary or similar to other programs, and then eliminating or consolidating them.

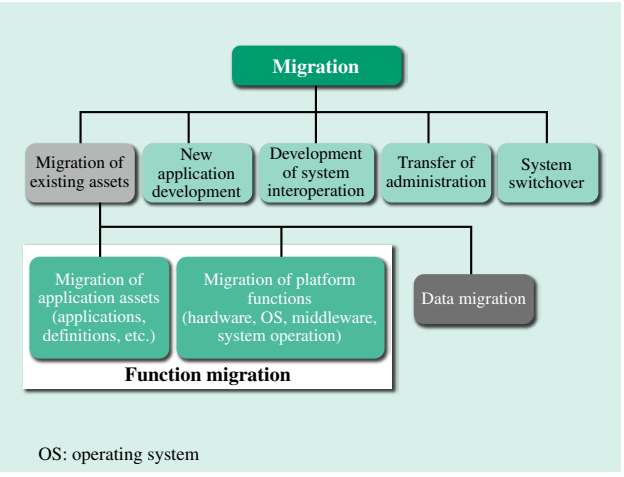
The first of these five steps, migration of existing assets, can be further broken down into the following three steps (see Fig. 2).

MIGRATION

Migration Process

When migrating a system to a different environment, it is necessary to deal with a variety of problems that arise due to the differences between the new and old systems. Examples include the influence of differences in hardware and architecture on existing assets, system switchover, post-switchover operation, interoperation with other systems, and how to satisfy new requirements. Hitachi believes that undertaking migration in a way that deals with these issues involves the following five steps (see Fig. 2).

(1) Migration of existing assets
(2) System switchover
(3) Transfer of administration
(4) Development of system interoperation
(5) New application development



OS: operating system

Fig. 2—Migration Process.
The migration process can be broadly divided into five steps. The migration service supports the migration of application assets and data migration.

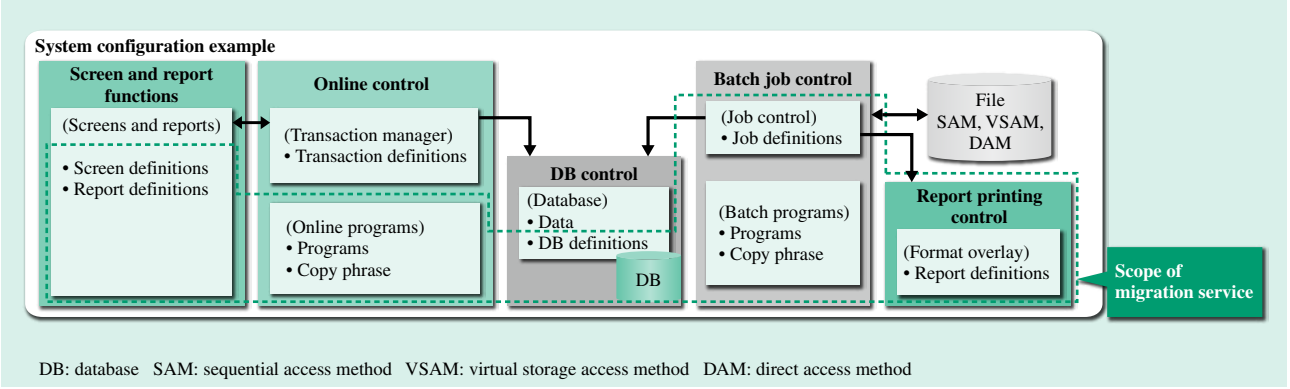


Fig. 3—Scope of Migration Service (System Configuration Example).
The migration service covers programs, job definitions, screen and report definitions, and DB definitions.

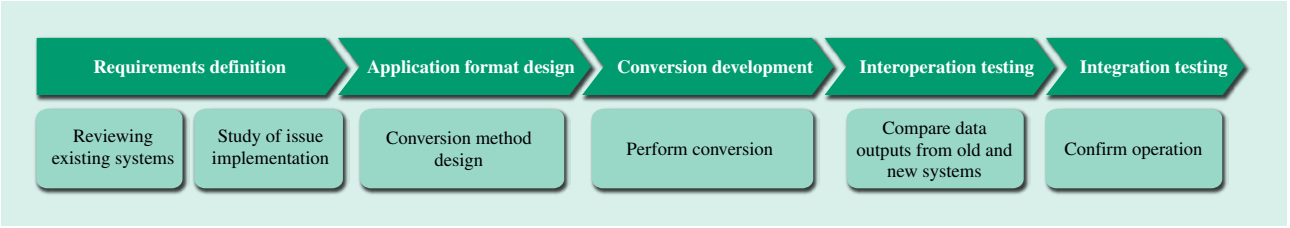


Fig. 4—Migration Process.
A key point with refactoring (downsizing) is to eliminate unused assets during the phase of reviewing the existing systems.

(1) Migration of application assets
(2) Migration of platform functions
(3) Data migration

The migration service supports the migration of application assets and data in particular, drawing on Hitachi's extensive know-how and techniques for analyzing existing systems to strive to make maximum use of the valuable assets that customers have built up over time and to avoid letting them go to waste.

Migration of Application Assets

The application assets of a system come in a variety of different types. The migration services can deal with a wide range of different types of assets, including programs such as those written in Common Business Oriented Language (COBOL), job definitions, screen and report definitions, and database (DB) definitions (see Fig. 3).

Migration Service Options

In addition to supporting the adoption of open systems for application assets, the migration service also includes a language upgrade service that includes languages such as COBOL and Java[*1]. The refactoring (downsizing) service is also offered as an option.

*1 Java is a registered trademark of Oracle and/or its affiliates.

Many customers use the refactoring service in conjunction with a service that supports migration to open systems and appreciate its ability to reorganize application assets as well as performing the migration. The following section describes how refactoring is done.

KEY POINTS IN REFACTORING

A key point when undertaking refactoring as part of migration is to perform this in the requirements definition phase. The reason for this is that, while the migration process starts with requirements definition and continues up to testing, the workload in the latter stages is significantly influenced by eliminating unused assets during the review of existing systems that forms part of the requirements definition phase (see Fig. 4).

REFACTORING PHILOSOPHY

The refactoring service for application assets is best undertaken in terms of both technical and business considerations. The service undertakes asset rationalization (downsizing) in accordance with technical considerations and supplies the information used as a basis for making decisions regarding the

reorganization of business functions with respect to business considerations. The services are described below.

(1) Technical considerations

(a) Elimination of unused assets

This identifies unused assets and highlights those that can be eliminated by analyzing information about the operation of the system and the relationships involved in calling application assets.

(b) Downsizing by consolidation

Code cloning techniques are used to find similarities between programs and identify program code that can be merged or shared.

(c) Elimination of redundant processing

This identifies unnecessary logic such as processing that is never executed.

(2) Business considerations

(a) Provision of information to facilitate elimination of functions

This involves analyzing logs to review system operation and provide information that can be used as a basis for making decisions about when functions can be eliminated.

(b) Provision of information to facilitate merging of functions

By identifying similar functions, this provides information that can be used as a basis for making decisions about which functions can be merged.

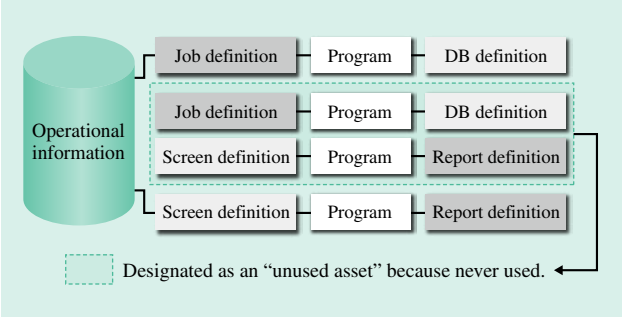


Fig. 5—Elimination of Unused Assets.
Application assets that are never used are designated as "unused assets."

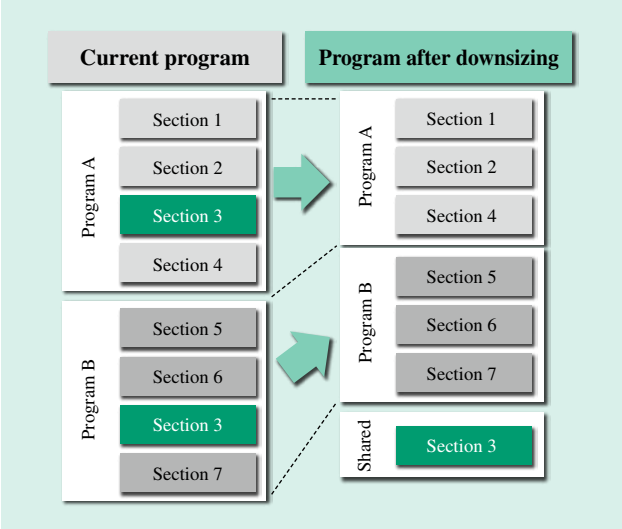


Fig. 6—Downsizing by Consolidation.
The sizes of program A and program B are reduced by factoring out the part they have in common (Section 3).

REFACTORING TECHNIQUES

The following techniques are used to deal with the technical considerations described above.

(1) Elimination of unused assets

This analyzes information about system operation to identify jobs or transactions that are never executed, and designates the associated job, program, and DB definitions as unused assets (see Fig. 5).

(2) Downsizing by consolidation

This reduces the size of the system by consolidating and factoring out data definitions or equivalent sections used by multiple programs (see Fig. 6).

(3) Elimination of redundant processing

This reduces the size of the system by identifying and deleting statements or sections of programs that are never executed (see Fig. 7).

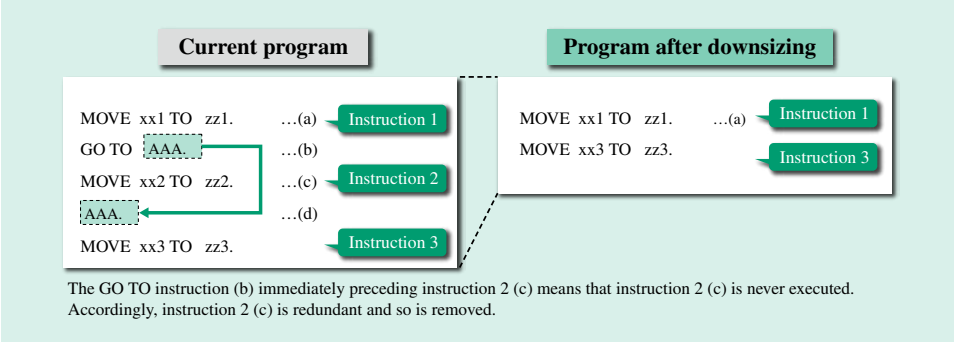


Fig. 7—Elimination of Redundant Processing.
Processing that is not executed is assumed to be redundant and is removed.

TABLE 1. Benefits of Downsizing Application Assets
The table lists the benefits of downsizing application assets in three case studies.

	Development date	Size of existing asset		Size of asset after downsizing		Reduction in size
Case study A	1980s	757 kilo-steps	1,520	526 kilo-steps	1,023	30.5%
Case study B	1980s	676 kilo-steps	1,687	361 kilo-steps	914	46.6%
Case study C	2000s	1,128 kilo-steps	1,438	1,064 kilo-steps	1,407	5.7%

Breakdown

	Technical considerations			Business considerations	
	Elimination of unused assets	Downsizing by consolidation	Elimination of redundant processing	Downsizing by eliminating functions	Downsizing by merging functions
Case study A	20.2%	2.4%	2.9%	3.3%	1.7%
Case study B	21.7%	2.5%	2.7%	19.7%	0%
Case study C	0.1%	5.4%	0.2%	Not used	Not used

CASE STUDY

Benefits of Downsizing Application Assets

Table 1 lists the benefits of downsizing application assets (programs) in projects where refactoring was undertaken.

Cases A and B were systems that had been in operation for several decades and had a large number of unused assets. Similarly, work on reorganizing functions also succeeded in downsizing application assets (programs) by roughly 30% to 46%.

Case C was a comparatively new system and had almost no unused assets. However, because there had been considerable reuse of existing programs and writing of new programs, benefits were achieved through consolidation.

Maintenance and Operating Cost Savings

Along with the benefits of downsizing application assets described above, the reduction in application assets is also expected to deliver the following maintenance and operating cost savings.

(1) Lower costs for modifications and other maintenance due to reduction in application assets

(2) More efficient maintenance due to consolidation of application assets

(3) Lower operating costs due to rationalization of operational processes through reorganizing of applications

CONCLUSIONS

Work has gotten underway in recent years on reforming large government information systems belonging to public agencies that still run on mainframes and that have been seen in the past as being difficult to reform.

In the future, Hitachi intends to continue contributing to the reform of large systems by working on refactoring (downsizing) and other migration techniques with more advanced functions and higher quality, and by providing these as services.

ABOUT THE AUTHORS

Atsushi Awakawa
Migration Service Department, Solution Division 1, Hitachi Government & Public Sector Systems, Ltd. He is currently engaged in all aspects of migration services.

Katsuya Tokuda
Migration Service Department, Solution Division 1, Hitachi Government & Public Sector Systems, Ltd. He is currently engaged in pre-consulting for migration services.

Toshimitsu Shibayama
Migration Service Department, Solution Division 1, Hitachi Government & Public Sector Systems, Ltd. He is currently engaged in mainframe migration.

Takuro Matsuzawa
Migration Service Department, Solution Division 1, Hitachi Government & Public Sector Systems, Ltd. He is currently engaged in mainframe refactoring (downsizing).