분산 개방형 EMS 응용 소프트웨어의 개발 기술

이지영.신철균.이석진.최양석.이정호.김상철
현대중공업

문영현.박정도.류헌수.국현종
연세대학교

Development Technique of Application Software for Open Distributed EMS

J.Y.Lee,C.G.Shin,S.J.Lee,Y.S.Choi,J.H.Lee,S.C.Kim
Hyundai Heavy Industries Co., Ltd.

Y.H.Moon,J.D.Park,H.S.Ryu,H.J.Kook
Yonsei University

Abstract - This paper presents a software development and management technique for open distributed EMS applications. The definite client-server configuration is proposed, and programming languages, software tools, application MMI and database are analyzed to make considerations to determine the development technique. This paper contains specific software management techniques for the development of EMS applications.

1. Introduction

EMS software is measured as having over two million lines of high level code above the operating system. Therefore, the development or implementation or integration of EMS software requires quite sophisticated software management techniques. This paper presents the distinctive solutions and the general factors to make the solution.

The concept of client and server application is the core technology of open distributed EMS. Several books have been written on client-server architecture. However, the concept can be expressed in one sentence. A client is a requestor of services; a server is a provider of services. The distinctive configuration for client-server programming is proposed. It is the prime factor to be considered for development of the technically and commercially competitive software. Some other factors are briefly discussed in the section 2 to provide some aspects of the considerations used for software development technique. In addition, a solution of the software internationalization is presented to provide capability to handle different languages.

Specific software development technique is presented in the section 3 to provide a consistent software development strategy, which has the prime importance in professional software development project.

2. Software Development Factors

A few factors presented in this section have the different considerations to make the software development technique proposed in the section 3.

2.1 Client-Server Architecture

An open distributed EMS consists of client PCs or workstations which run the man machine portion of applications and some other applications, and servers running UNIX or Windows NT as the operating system. The majority of applications run on the servers. The servers can be subdivided into two general classes: 1) database server- accesses database files to return requested data to the client, and 2) compute server- performs computations for the client process.

2.2 Programming Languages and Too

Three programming languages may be used: FORTRAN, C and C++. In Addition, macro languages which are included in general Windows applications such as word processors, spreadsheet programs, and database front ends may be used within these products. These products should be used whenever they contain

functionality required when developing a new applications. For example, Excel contains extensive graphing capability. An Excel spreadsheet and macro should be used to load and graph data if Excel provides a graph of the appropriate type.

2.3 Application Man Machine Interfac

All customer man machine displays related to a single application should be contained in one program. Extensive use of existing application man machine programs and Windows programs should be made to reduce the amount of programming required for the application. In fact that, program results stored in a text file may be viewed by having the application man machine program start a text editor such as Notepad to load and display the contents of the text file. The Application designer should consider using commercial Windows software to reduce the amount of programming required for the Application.

2.4 Database

When designing software that uses database techniques, commercial databases should be used whenever practical. This results in much more cost effective software, as less code has to be developed, and provides greatly increased functionality, as it enables the use of the wealth of commercial software that interfaces to standard commercial databases. There are cases where commercial databases are not suited for reasons of performance and functionality, for example the real-time database. However, future advances in database technology may enable the use of commercial database products for these databases in the future.

2.5 Internationalization

There are a number of factors which must be considered when designing and coding Applications which are intended for an international market. Differences in language, character set, and numerical and pictorial presentation must be considered. The programming considerations are based on the use of the Unicode character set to design and code Applications to support multiple languages. Most commercial products will support for Unicode in the near future. Consequently, as much work as possible should be performed now to simplify the change to Unicode in the future

3. Software Development Techniqu

The presenting concise software development technique includes methodology, planning, strategy, and management of the software.

3.1 Project Team

3.1.1 Configuration Manager

The following are the responsibilities of the configuration manager: 1) monitoring the development and maintenance process to ensure the reliability and integrity of software application releases, 2) specifying who can read and/or write(update) the stored information, 3) specifying who can insert new data structures, and 4) reviewing, approving, disapproving changes made to data structure.

3.1.2 Programmers/Engineers

Their responsibilities are 1) making changes to the text data structure, 2) inserting new code or testing of any kind or bug fixes, 3) checking out the appropriate software, and 4) checking it back, ensuring there are no conflicts resulting from other changes being checked in since the code was checked out.

3.2 Application Development Plan

The steps outlined below should be followed in implementing new Applications.

1) Determine the best tools, languages, etc. for software development.

2) Work with the database administrator to define the directory structure to be used.

3) Each developer should create a working directory structure on his workstation or PC to match the defined project directory structure.

4) As each program is developed and documentation is written, the developers check-in the source files to configuration control.

5) When the configuration manager has determined that sufficient functionality exists for integration testing, he copies the contents of development to test environment.

6) When all software has been developed and tested, the configuration manager requests approval to release the software.

3.3 Application Distribution for Client Programming

In order to determine which Applications belong on what hardware, the rules below should be followed: 1) Man machine processes should not be concerned with the file structures and formats of databases on servers, 2) All formatting of data for display purposes should be performed by the man machine process, 3) Calculations which are performed solely for display purposes should be done in the man machine process, and 4) All Applications related calculations should be performed by a calculation server, and not a man machine process. Since consistency leads to easier maintenance, these rules should be followed for every Application.

3.4 Software Management

3.4.1 Software Source Control

Source control plays a significant role in the software development and maintenance life cycle. It is a set of tools or products that provide a method for managing, controlling and tracking all of the systems development and maintenance projects. Source control ensures that all software and information needed to build that software, is always available and adequately backed up, thus ensuring that the investment in the development of that software is never lost, as would be the case, if software is maintained in personal files or if only one person knew how to build the software.

3.4.2 Source Control Database Structure

Information is stored in a database where it is organized as a hierarchy of data structures. These data structures consist of database, systems, configurations, modules, and items. The following are the different configuration database structures: 1)Base configuration (initial version), 2)Development Configuration(base and successive production configuration for development work), 3)Test configuration(testing and integrating new developed software),and 4)Production Configuration (integrated version of the application package). Each of four configurations should have the following modules under them: 1)LIB-contains routines collected as libraries, 2)INCLUDE-contains header files shared by all software, 3)BIN-contains all executable, 4)DOC-contains all necessary documents, and 5)Application package modules-contains the source code and objects.

3.5 Software development under Prog Environment

3.5.1 Change Cycles

The most common Application management is through a software life cycle. A software life cycle is a formal model that identifies discrete stages in the software development process. Developers make changes to code, build executables, and pass the complete set of changes to test personnel. Test personnel test the changes and if approved, a production version is built and released. These distinct phases: development, test and production can be used as separate environments using the source control manager.

3.5.2 Concurrent Development

Source control manager prevents other developers from checking out items that are already checked out. This ensures that only one developer can make changes at a time. But this safety feature can impose severe time restraints when modification conflicts occur frequently in a project. Source control manger also supports concurrent development for those projects that has frequent modification conflicts.

4. Conclusions

The proposing techniques are designed for a large software development, integration and implementation project. These are Hyundai Heavy Industries software development techniques for ongoing open distributed EMS development project. The project is in progress and has promising results.

5. Acknowledgement

This project is financed in part by a contract from Korea Electric Power Corporation.

[References]

They are various internal publications of Hyundai Heavy Industries, which are not available to the public.