

Adaptable Software Systems with Business Objects

5 R - 7

Kazushi Kuse, Fumihiko Kitayama, Kazuhiro Konno,
Saburoh Hirasuka*, Yukiko Noro*

IBM Tokyo Research Laboratory
IBM Consulting Group *

1 Introduction

Business environments are changing rapidly and radically, but existing software systems are too rigid to be adapted quickly to new requirements. We therefore propose a method for developing adaptable software systems by using business objects, and a corresponding run-time environment. An adaptable software system can be easily enhanced and customized to meet new requirements not anticipated during its development phase. We evaluated our development method and run-time environment through the actual development of a client/server system for branches of a finance company. On the basis of this experience, we explain the technical issues and key factors involved in realizing an adaptable software system. Three major elements must be considered: (1) a development method, (2) a run-time environment, and (3) development and enhancement tools. We describe our approach in terms of these elements.

2 Development Method Based on Business Objects

2.1 Business Objects

The benefits of object-oriented technology are not limited to software, but are now being applied to business processes through what is called object-oriented business engineering [1]. This has given rise to the concept of business objects. Business objects are represented by business terms rather than computer system terms, and are thus easy for system users to understand [2].

2.2 Development Process

We built a development process by introducing business objects into traditional object-oriented methodologies such as the object-modeling

technique(OMT). The following is a rough sketch of our development process:

- a) Business structure and flow analysis
- b) Business object analysis
- c) Analysis of changes in requirements
- d) Business object and workflow design
- e) System object design
- f) Implementation

The outputs of the first four steps, such as business object diagrams, are described in ordinary business terms, and are used as the entry points for future system enhancements. When a change in the requirements occurs after the development of the system, it can easily be represented by means of these outputs, and reflected systematically in the implemented system. For this purpose, the process needs to be traceable, and to be supported by tools. Among the above steps, the analysis of changes in requirements is the most important.

2.3 Analysis of Changes in Requirements

Some current software systems are already capable of being adapted to new requirements, but their adaptability was built into the systems in ad-hoc ways, and is limited to possible changes in requirements identified in their analysis and design phases. Using our method, a software system can easily be adapted to new requirements even if they were not anticipated during the system's development phase. Analysis of changes in requirements is a systematic way of adjusting to various requirements, consisting the following steps:

- a) Identification of concrete changes in requirements
- b) Classification
- c) Abstraction
- d) Selection of adaptation mechanisms

The adaptation mechanisms include abstract class, design patterns, and workflow. One of the key issues here is how the analysis results are represented in analysis and design diagrams. Each adaptable portion needs to be clearly described in those diagrams in order to be used as an entry point for future system enhancements.

3 Run-Time Environment for Adaptability

3.1 System Assumptions

Our target systems are distributed systems that support cooperative work. Business and system objects are basically implemented with distributed objects. The system includes UI components to support interactive work.

3.2 Workflow Management Systems

We introduced workflow models in some steps of our development method, so a workflow management system can be used in implementing a target software system. We actually used a workflow management system to implement a pilot system, as we will describe later on. However, this involved some problems, particularly with respect to the use of a distributed object system. First of all, its execution model does not match those of distributed objects. A workflow manager controls a target software system according to a global execution view called a work task. A work task consists of several work steps, which are connected by a workflow. On the other hand, a distributed object is implemented as a self-contained object. Its execution method consists of sending messages to each object. There are also some duplicated items in the two systems, such as servers and data-sharing mechanisms. Resource consumption and difficulty of system management are further points that need to be considered.

3.3 Distributed Objects with Workflow Capability

To solve the above problems, we propose an extended distributed object system, which provides a workflow management capability. The key issue is how we keep the workflow description independent of the application logic im-

plemented in business objects. The use of meta-classes might be one way of realizing such independence.

4 Development and Enhancement Tools

We need two kinds of tool to effectively develop an adaptable system, and to enhance it in accordance with future requirements. The outputs of development phases will be used as the entry points of future enhancements, so these two kinds of tool are closely related to each other. We built prototypes of the tools, in which each element of a diagram is realized as an visual programmable object.

5 Experience

To evaluate our method, we developed a pilot system for branches of a finance company in a five-month project involving seven people. The system supports cooperative work by customers, tellers, customer advisors, and a branch manager. We used our method to analyze and design the system, and implemented it by using a distributed object system, a workflow management system, and a visual programming tool. After completing the development, we added several enhancements such as Internet-based customer services. The new requirements were not anticipated in the development phase, but the existing system was easily adapted to them.

6 Discussion

We proposed a method for developing an adaptable software system based on business objects. The development process, run-time environments, and enhancement tools are major elements of the method. We evaluated the method through the development of an actual system, and obtained some good preliminary results.

References

- [1] Object-Oriented Experiences and Future Trends, IEEE, Vol.38, No.10
- [2] OMG Business Application Architecture, Business Object Management Special Interest Group, Object Management Group, Inc., 1995