Experimenting Component-Based Technology in Industry Settings

Mitsuru Ohba and Yasuomi Sato

Computer Science Department, Hiroshima City University 151-5, Ozuka Numata-cho, Asaminami-ku
Hiroshima, Japan 731-31 e-mail: ohba@cs.hiroshima-cu.ac.jp

1. The Project

The component-based software development was new to the local software industry in Hiroshima, Japan. It seemed to be promising to a certain extent. At least in academic settings, it worked very well. Although the benefits of applying the technology for applications development in industry were often advocated, the local professionals were still very skeptical.

One of the reasons of this skepticism is that they have not tried to apply the technology. They do not see what the real benefits are. They know, however, that a certain amount of required for them to master such a new way of developing software. They also know that there are risks of experimenting a new technology in real projects.

For the development of the local software industry in Hiroshima, we proposed the city government and some local companies to experiment the new technology in real industry settings. It can be seen that the experiment was an organizational learning for those who participated the project to know if the technology is really promising. The city government supported the project for two years starting 1995.

2. Goal, Objectives, and Method

The total goal of the project was to transfer the component-based software technology from academia to local industry if it can significantly improve software productivity and quality. The primary objective of the project was to learn the component-based software development in industry. It meant to know how it was difficult to learn and use the new technology for average industry people. It also meant to know what are advantages and disadvantages of using the technology.

The secondary objective was to assess benefits of the new technology in applying to developments of typical industry applications. It meant to measure how the technology may improve software productivity in industrial settings. It also meant to understand what the key components are.

The method of study we used was experimentation. We examined the technology in realistic industry settings. This means that software engineers from the local software industry redeveloped several existing and new applications using the proposed technology although the process to be applied was unknown.

3. Experiments

A preliminary study was initiated by the university researchers and a group of industry people to understand what the new technology was and how it should be used. Two existing applications were selected to conduct the study: one was a sales analysis application for a large supermarket, and the other was a BM (bills of materials) management for an auto-manufacturing company. The two existing applications were redesigned completely and prototyped using component-ware and Smalltalk. One of the key issues of this study was to define the process to be applied for the future experiments.

The first full-scale experiment was planned and conducted in 1995. The experiment consisted of three small experiments: the first experiment was a development of a commonly used parts for applications development, the second experiments was a development of a customer information management system for public services and the last experiment was a development a material inventory and production control system for a manufacturing company. The two existing applications were redesigned completely and prototyped using component-ware, Smalltalk and the generic Smalltalk class library.

The second full-scale experiment was planned and conducted in 1996. The experiment consisted of three small experiments: the first experiment was a development of a commonly used parts for applications development, the second experiments was a development of a customer information management system for public services and the last experiment was a development a material inventory and production control system for a manufacturing company. The two existing applications were redesigned and prototyped using component-ware, Smalltalk and the generic Smalltalk class library.
retrieval for Products Liability in manufacturing companies, and the last experiment was a development of a customers' personal information management system for public service companies. The two generic applications were designed and developed using Smalltalk-based component-ware, C++, SQL database system and the common C++ class library.

4. Results

As a result of the preliminary experiment, we found that the two existing applications we examined handled a set of information which could be structured by nodes and links. For the experiment, the two teams independently developed similar Smalltalk class libraries for handling nodes and links.

On the basis of the preliminary study, we decided to develop a generic class library for handling information which could be structured using nodes and links. Most of information which our applications handled was hierarchically structured. There were some cases in which information was network-structured. In most cases, nodes were used for representing organizational units (e.g., production lines), installations (e.g., warehouses) and facilities (e.g., ships). Links were in most cases used for representing activities (e.g., assembly, shipping) and relationships (e.g., products and parts).

The base-process of the first year experiment was: 1) analysis of existing applications, 2) development of specifications prototypes using component-ware, 3) review of the specifications prototypes, 4) development of full-scale prototypes using Smalltalk.

On the basis of the conclusions of the preliminary study and the first year experiments, we put our focus on the implementation issues in the second year experiment. The purpose of the second year was to develop applications using C++ and develop a generic C++ class library for handling network-structured multimedia information.

Two new generic application systems were defined by generalizing the applications examined through the first year experiment. One of them was a BM and production information management system for a medium-scale manufacturing company. The other was a customer service information management system for a public service company.

The base-process of the second year experiment was: 1) definitions of application functions, 2) development of prototypes using component-ware and Smalltalk, 3) review of the prototypes, 4) development of full-scale systems using C++.

5. Conclusion

From the first year experiment, we concluded: 1) component-ware which we used for developing the graphic user interface and the database interface drastically improved the productivity, 2) the generic class library for handling nodes and links were significantly reused and contributed to the reduction of our development period, and 3) the object oriented technology was effective to develop a class of applications which needed to handle network-structured multimedia information.

From the second year experiment, we concluded: 1) know-how of linking components written in different languages and developed separately was the real key to success, 2) system integration experience was the basis of the know-how, and 3) application development using component-ware and the generic class library was relatively straightforward. We have not reached to the final conclusion on to what degree the component-based software technology improves the present software productivity and quality in industry.

6. Acknowledgment

We thank the Department of Industry Development, Hiroshima City for their support. We also thank the local companies that participated the project: Mazda, Chugoku Electric Power, Jukken, Hiroshima Gas, Hiroshima Bank, HEK, NEC Chugoku Software, Chugoku Information Systems Service, Mazda C&T, SRA Chugoku, Bien, Mishima Graphics, Ryosen Engineers, NTT, Fujitsu, IBM Japan, and Hitachi.