

# **PARADIGM SHIFTS IN BUILDING INFORMATION SYSTEMS**

**Toru TAKESHITA**

The College of Business Administration and Information Science  
The Research Institute for Information Science  
Chubu University

1200 Matsumoto-Cho, Kasugai, Aichi, 487, Japan  
Tel (Japan) : 0568-51-1111 ex.2808  
Fax (Japan) : 0568-52-1505

This paper was presented at the symposium,

**International Business in a Competitive Shrinking World**

which was held at Ohio University, USA on September 17, 1993.

## Paradigm Shifts in Building Information Systems

Toru TAKESHITA

### Abstract

Information systems (IS) are playing ever-increasingly vital roles in business operations, decision making, and end-user computing in enterprises. As computer and communication technologies have advanced, the information systems have evolved in terms of functions, processing modes, and in the ways they are built.

The latest paradigm shifts observed around the business world are 'downsizing' and 'distributed processing'.

Business re-engineering, namely, process re-designing is changing the ways in which organizations do business. It is necessary and essential to meet the changing consumers' needs, technologies, products, and marketing strategies. It also means significant savings of time and money as an estimate in the US indicates that there is room for an average of 75% improvement in efficiency. Software re-engineering technologies help identify those processes which should be re-engineered and facilitate the transformation of the current processes to new ones.

The latest technologies to advance information systems are discussed with some mention of the US-Japan differences.

As the concluding remarks, the future directions of the information systems are described, and more international collaborations which would make the most of individual countries' technological strong points are proposed and predicted.

### 1. Introduction

For the past 35 years the author has been involved in software research, development, marketing, and systems engineering support, and taken part in the development of one of the first COBOL compilers, managed the development of the 1964 IBM Tokyo Olympic Information System, which was the first on-line information system (using commercially available computers) that had ever been installed in Japan. The efficiency of this system was quite impressive in that the final Olympic Record Book, which had previously taken more than six months to be compiled, was submitted to the Olympic Committee Chairman at its closing ceremony. The success of the project gave a significant impact on the rapid installation of on-line systems in Japan. From 1980 to 1983, the author managed a project to develop a high-reliability, high-speed DBMS which is currently being used by many banks around the world. He spent a total of nine years on three assignments to the US for software development, future system product planning, and as a corporate staff member consulting on software technologies.

With this background, he has been greatly interested in software engineering

technologies as well as in information systems installed in private and public enterprises.

Software engineering has been advancing since its name was coined in 1968. CASE (Computer-Aided Software Engineering) tools began to be built in the 1970s and gained popularity as a result of appearing in the September 24, 1986 issue of the Wall Street Journal. It has been reported that there were 300 CASE vendors and more than 850 CASE products on the market in 1991.

Rapidly advancing computing and communication technologies have enabled changes of information systems in their functions, processing power and modes. Information systems are playing ever increasingly vital roles in enterprises, governments, homes, etc.

Information is a valuable asset in any enterprise, equivalent to conventional assets such as people, materials, and money. The information systems which transfer and/or transform information are essential in any organization, business or government. They are the key to success of an enterprise's domestic operations as well as to that enterprise's competitiveness in the shrinking world of international business.

This paper retrospects the various paradigm shifts in processing by information systems and in developing them, and reviews the current status (limitations of conventional IS and their alternatives), and state-of-the-art technologies used to build IS.

Several information system development methodologies have been proposed and practiced, and various powerful CASE tools to support them have appeared. The characteristics of information systems and the ways in which they are built vary from country to country. In this paper, some differences between the US and Japan will be mentioned.

Conventional information systems have achieved a great deal in serving the enterprises and consumers, but have also shown some limitations or deadlocks. Because of the problems caused by existing IS and because of the debuts of cost effective workstations (WS) and personal computers (PC), the down-sizing of the systems installed in private enterprises began to take place toward the end of the last decade. Their subsequent implications will be discussed.

Business re-engineering is considered to be high-priority as its potential pay-off is so large. The re-engineering of application programs is often very difficult and costly. The good news is that modern technologies and support tools for software re-engineering can help facilitate this.

Finally, the paper will conclude with the future perspectives of IS technologies and a proposal/prediction for international collaborations which would make the most of individual countries' technological strong points.

## **2. Past Paradigm Shifts: Processing Modes and Building Methods**

### **2.1 Processing Paradigm Shifts**

During the early days of computing, all the processing was conducted in a batch mode. In the middle of the 1960s, on-line, centralized data processing was started with the debut of the magnetic drum and disk devices supported by data base management systems, coupled with digital communication services over telephone networks. Toward the end of the same decade, time-sharing processing of application programs as well as

program development on a single CPU were made possible, significantly speeding up the development and execution of scientific and engineering applications.

With the appearance of cathod-ray tube (CRT) display devices, man-machine interaction became remarkably easier. Supported by data base management systems and end user-oriented languages (forth-generation languages, 4GLs), management information systems (MIS) or management decision support systems (DSS) and end-user computing (EUC) were introduced. More recently, management information systems are called 'strategic information systems (SIS)'.

In addition to supporting enterprises' business operations with fast, timely, and accurate processing of massive amounts of data for everything from sales to information query, presenting easy-to-understand formatted text or graphical data made some information systems serve as highly effective, easy-to-use MIS systems. During the 1960s, it was generally recognized that information systems in Japan were more sophisticated than US counterparts as the former were designed to take care of many exceptions and to provide extra services. Japanese users tried to make the most of the computing equipment which was very expensive at that time.

Initially, it was typical that all the raw data was sent to the central location and processed there. In the early days (till sometime in the 1960s), card decks and magnetic tapes were mailed to the center. Then, with increases in the data to be processed and the availability of mid-size machines and mini-computers, decentralization of the processing was started to help speed local decisions, increase autonomy, and reduce the amount of data to be sent to the headquarters.

The configuration of the distributed systems at that time were mostly hierarchical with smaller machines located at branch offices, plants, or departments, to which mini-computers or non-intelligent terminals were connected. The branch office, plant, or departmental machines were connected to the central mainframes via communication lines. Horizontal networks were rare till the second half of the 1980s when client/server model connections of UNIX machines became popular.

## 2.2 Development Paradigm Shifts

The long delay of the IBM OS/360 development in the mid-1960s led to the establishment of formal software development processes and management control. In Japan, software development management realized the importance of well-defined, disciplined production of software with standardized processes like those which are seen in production plants of the manufacturing industry. Thus, the idea of a 'software factory' (also called 'software plant') was implemented by Japanese Computer Manufacturers (JCMs). The first such factory was built by Hitachi in 1969, followed by NEC, Toshiba, and Fujitsu establishing similar institutions from 1976 to 1979.

In 1981, NEC launched SWQC (software quality control). In the next year or two, other JCM companies followed suit. There is no doubt that the QC activities have greatly contributed to the improvement of both quality and productivity in software development by JCMs and their subsidiaries.

Until the middle of 1960s, programmers tended to write programs in the ad hoc, code-and-fix mode. The structured programming paradigm was initiated by E. W. Dijkstra who proposed goto-less programming in 1960.

A structured program consists of three kinds of basic constructs: sequences, alternations (if statement, case statement, etc.), and repetition (while statement, until statement, etc.). This new paradigm became a driving force to expand software engineering to industrialize software production and gave birth to several new design methods and programming languages in the 1970s.

"Structured Design" co-authored by L. L. Constantine, G. J. Myers, and W. P. Stevens and printed in the IBM Systems Journal made the design method conceived by Constantine called "structured design". Myers enhanced the Constantine's method to establish the composite design method. The two methods are combined and often called the composite/structured design methodology.

E. Yourdon and T. DeMarco established a structured analysis methodology and C. Gane and T. Sarsen developed a similar one, both of which use data flow diagrams (DFDs) and structure charts (SCs).

From 1970s to 1980s data-oriented design methodologies were emphasized. A method created by M. A. Jackson is known as the Jackson method. Another initiated by J. D. Warnier and enhanced by K. T. Orr is called the Warnier-Orr method. Then in the middle of 1980s, James Martin came up with the Information Engineering methodology for enterprise modeling and structured implementation in a topdown fashion.

In the second half of 1980s, software reuse began to draw general attention. The functions for creation (with a special editor), storage in libraries, retrieval, synthesis, and customization were included in the CASE environments developed in Japan. In the meantime, various projects to promote software reuse were organized in the USA and within JCMs and NTT (Nippon Telephone and Telegraph Corp.).

In 1967, the first object-oriented programming language (OOPL), Simula67 appeared. In 1979, the first internal version of Smalltalk-80 was completed at Xerox PARC. The licencing of C++ which was developed at AT&T's Bell Laboratory was started in 1985. Some other object-oriented programming languages such as Object Pascal, Objective-C, and Eiffel also made their debut around this time period. The CODASYL OO COBOL Working Group was formed in 1989.

A good number of OOPL programming support environments have been released. At the end of the 1980s, several OOA and OOD methodologies and their CASE tools began to appear. Thus, the new development paradigm can now be applied from the analysis phase through the implementation phase for building information systems.

There are two schools of thought regarding the object-oriented approach: those who consider OO as an extension to structured technologies, and those who believe that OO is a radically different methodology.

J. Martin, who started promoting information engineering in the mid-1970s, has recently conceived object-oriented information engineering (OOIE) and in 1992, published a book containing its detailed analysis and design activities with graphical notations.

It is not easy for structured technology practitioners to change to the OO paradigm. But as experience in the use of the OO approach in business applications become more wide-spread, general adaptation of this new paradigm to the development of information systems will prevail. Increasing interest in this paradigm is recognized in Japan, but the actual use of it for building IS is still rare.

### 3. Descriptions and Comparisons of Development Processes and Tools: the US vs Japan

#### 3.1 Processes and Tools for Building Information Systems

Early in the 1970s, JCMs' management realized that they had caught up with US computer manufacturers in the development and production of hardware, but that JCMs were far behind in the software development process. They first defined software development management methodologies and then established detailed processes for software development with project management and quality control. These methodologies and processes were compiled into user manuals and provided to their own customers for their use in building their information systems.

Toward the end of the 1970s, they began serious efforts to automate development processes using computers by developing various software tools and integrated support systems. The first such system, which was called 'SWB (Software Workbench),' was internally released at Toshiba Fuchu Works in 1976. Hitachi's Software Works and its wholly-owned subsidiary, Hitachi Software Engineering jointly developed CASD (the Computer Aided Software Development system) and CAPS (the Computer Aided Production Control system for software) at the beginning of the 1980s. Hitachi presented ICAS (Integrated Computer Aided Software engineering system) at COMPSAC'81.

In 1983, NEC began presenting their integrated software development environment called SEA/I (Software Engineering Architecture/I) at the national conferences of the IPSJ (Information Processing Society of Japan).

JCMs and NTT have been ahead of the USA in the use of graphical representations and software reuse. Graphic notations (tree structure charts) were used for the representations of detailed designs, from which source code is generated and the animation of which is used for dynamic debugging. Also the use of software parts is more advanced in Japan than in USA. Commonly used frameworks or patterns are provided as skeletons and subroutine-type code fragments as plug-in parts from JCMs. The users can also add their own parts.

When JCMs and NTT were devoted to improving the lower CASE tools, US software vendors were developing analysis and design (SA/SD) tools which employ various diagrams and charts taking advantage of large, high-resolution display screens on the PC. US vendors first released upper CASE products in 1983. It took several years before JCMs began to incorporate similar capabilities into their CASE environments.

JCMs and NTT have been continuing their R&D efforts to enhance their CASE environments to become completely integrated systems which would support the entire life cycle - from the initial (enterprise analysis) phase through maintenance. Unlike US computer manufacturers, each JCM provides a total set of tool functions on their CASE environments, which have been closed (not allowing for users to easily include other vendors' tools) until recently.

#### 3.2 The Information Engineering Approach vs the Japanese Style

Several information system development methodologies have been proposed and supporting tools have been made available. J. Martin's Information Engineering (IE)

which is one of the best known information system development methodologies defines the IS development life cycle as consisting of four major phases:

1. Information strategy planning: goals-and success factors, how to utilize IS to increase competitiveness and new opportunities
2. Business area analysis: business processes, their relationships, and required data; creating corporate data models and process models
3. System design: decomposing business processes into processing procedures, and how to implement the procedures; participation of end users
4. Implementation: generation of code.

This kind of topdown approach seems to be well-accepted in the USA. However, there is a skeptical view in Japan that the end users may not be able to communicate their real requirements because they may not understand fully nor express precisely what they need. Therefore, the generated code often does not meet users' needs, causing many redos and reworks. To minimize these problems, end-users' participation is encouraged in the design of procedures and the validation of prototypes in the USA.

In the building industry, the users' requirements are confirmed and validated at the very beginning by showing the client a set of the descriptions, drawings, and pictures of the components similar to those which will make up the new building to be constructed. A similar practice is often observed in some sections of the Japanese software industry. In case similar applications have been developed, a bottom-up approach is taken as shown below:

- Discussion with users
- Selection of reusable parts
- Validation/refinement with users
- Customization/additional code
- Final products satisfy users

In this approach, users who may not know at the beginning how to state precisely what they need, are shown something close to what will be delivered to them. This way, they can more clearly tell the developers what changes they would like, thus eliminating misunderstanding about their requirements before the real development work is started.

Therefore, the final application systems satisfying the users' needs are developed in a relatively short period of time. The short-coming of this approach is that when a new requirement belongs to an entirely different domain, it may take much longer to develop the system.

## 4. Latest Paradigm Shifts and Technologies

### 4.1 Conventional Paradigms' Strengths and Weaknesses

Centralized data processing using mainframes in the computing center contributed a great deal to the rationalization and efficiency of the enterprise by the fast and accurate processing of massive amounts of data and by the timely supply/retrieval of critical

information for decision making. On the other hand, it brought about many problems, such as:

- Increasingly large DP cost
- Accumulated backlog
- Long development turn-around time
- Complex, expensive maintenance and operation management
- Difficulty in recruiting and developing professionals.

## 4.2 New Technologies for Alternatives

Dramatic technological advancement has brought about the following new technologies and trends since the 1980s:

- High-performance PCs
- Better cost-performance RISC
- Advancing networking technologies (including network OS)
- Open systems using UNIX workstations
- Relational DBMS with SQL and 4GLs
- Downsizing as a natural trend.

As a viable alternative to the centralized processing, users have chosen downsizing to networks of PCs and workstations.

Management of enterprises are demanding DP managers consider downsizing information systems, and do not easily approve a replacement by a larger system or enhancements of a mainframe system. Thus the DP staff people tend to or are forced to, select distributed processing using smaller machines (PCs and WSs).

## 4.3 Implications of Downsizing

Distributed processing using PCs and WSs available at much lower cost and connected to LAN and WAN has several merits. End-users can develop their applications in a much shorter period of time without depending on the DP department, and can have much easier access to the data they need. Its total equipment and running cost is perceived to be much lower than the centralized processing cost.

Some of the apparent advantages are listed below:

- Better total cost performance
- Elimination/reduction of backlog
- Establishment of end-user computing
- Reduction of total operational cost.

## 4.4 Downsizing Does Not Always Succeed

However, there are some demerits and risks also. Application development by end-users may create a great diversity of data types, and homonyms and synonym problems. If ready-made, programming-less packages are not sufficient for a specific application domain, different end-users may individually create different programs for the same functions at additional cost to the enterprise. These separate programs could be covered by a single one if written by the centralized DP shop. Also as the end-users

are rotated and transferred out, the problem arises of who will maintain the programs created by the previous users.

Because of many different machines, OS's, communication protocols, and software pieces, management of the whole system is rather complex and time consuming. In addition, there are complex back-up and recovery problems as well as security concerns, none of which are easy problems to solve.

It was recently reported that the total cost of a new system downsized to PC LAN at a few companies in the US is three times as high as the previous centralized processing on the mainframes.

Some predict that, after many users have experienced problems associated with distributed processing, they may consider a partial move back to the centralized approach, or may strike a compromise, blending centralized and distributed processing modes using 'right-sized' machines and networks.

If the enterprise finds it difficult to keep up the IS organization for lack of expertise or because of increasing cost to cover the multi-vendor machines, and different software pieces and communication networks, then more dependency on vendors is required for:

- Heterogeneous computing network design
- Network operation and management
- End-user computing support.

Vendors are also required to provide CASE environments to enable distributed application development and software tools to help prepare for and manage distributed system operations.

#### 4.5 Increasing Solution Services Offered by Vendors

Till the 1980s, Japanese companies using computers seldom depended on outside people for development of information systems. Ready-made business application packages were rarely used in Japan. It was during the last decade that large banks and companies started to hire temporary outside programmers for coding and testing application programs. But the analysis and design were performed by their own analysts/designers in-house. In Japan, multi-vendoring was started in the 1970's to facilitate business connection and to reduce cost and risk.

By the end of the last decade, powerful WSs with horse-power almost equivalent to that of mainframes had been made available, and the capacity and performance of PCs had improved remarkably. These technological advancements have caused the downsizing of computing systems into multi-vendors' WSs and PCs and the switching from a closed environment, containing equipment and software supplied by a single vendor, to an open system environment. Also, as the price of PCs and WS's have dramatically gone down, it became a lot easier for end-users to purchase computing machines and software and start their own end-user computing.

The mixed use of different machines and OS's connected to various networks make the in-house system design, management and (software) maintenance of all the components very difficult. Thus, starting around the end of the 1980s, many enterprises began to contract with manufacturers and vendors for system analysis and design, system integration, network management, facility management, and so forth. Several companies have gone so far as a complete outsourcing, depending on a vendor for the

entire DP, data network services, and end-user support.

The author is concerned that too much dependency on outside vendors may cause problems in the future.

## 5. Business and Software Re-Engineering

### 5.1 Profit Increase Restructuring in Enterprises

In the past few years, the news media has reported almost daily about enterprises' mergers, facility shut-downs, early employee retirement programs and sudden layoffs, slimming and restructuring of business operations and organizations and so forth. These actions are taken to cope with business depression, to make up for a slow recovery from economic and industrial slumps, and to stage a final bid for corporate survival.

Some people are afraid that the sharp reduction of the DP budget and suspension or reduction of key information systems may make decision making and resource utilization very difficult.

How business operations should be restructured is outside the scope of this paper, but the author wishes to include a Japanese approach because a similar one may be applicable to IS restructuring. Instead of choosing only business downsizing which will cause negative side effects such as unemployment, lower morale, and lack of strategic planning, alternatives should be carefully explored. One of these is a bottom-up, customer-oriented approach. This approach has front-line working people finding potential growth products or services to bring about the survival and expansion of the business.

The following schematics illustrates how bottom-up restructuring business operation may be carried out:

- Rather than limiting to slimming and restructuring organizations and activities to reduce the operating cost
- High added-value products/services should be identified and expanded
- Reverse (bottom-up) approach.

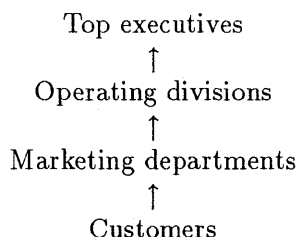


Figure 1. Bottom-up business re-engineering.

Front-line people familiar with customer needs should form a study team, identify growth products or services, and submit the new ideas to higher levels of the enterprise.

Information system restructuring should be prioritized to help promote the new

growth products and profit-increasing services.

## 5.2 Business Re-Engineering

To keep abreast with both the internal and external changes, many existing business processes need be re-designed and re-implemented. Business re-engineering, namely, business process re-design, means that organizational functions are re-adjusted along more strategic lines. The processes currently supporting the business operations are reviewed and re-designed to reflect efficient ways to achieve the enterprise's goals. From the results of re-engineering which has been done so far in the US, it is estimated that potential productivity could be improved by more than 75%. According to a survey conducted by Information Week in 1991, 86% of the companies responding were undertaking business re-engineering.

There are success cases and failures. It is said that short-term, makeshift re-engineering after reaching a crisis often results in failure, while long-term, strategic undertakings conducted well ahead of time, have a good chance of success.

As business processes are generally supported by highly automated information systems in the enterprises of today, re-design of business processes may have significant affects on existing information systems. The recent strong trends of equipment downsizing and distributed processing are bringing about the need for the drastic re-engineering of existing information systems as well as the business processes supported by them. This is over and above the ordinary maintenance work needed to meet new requirements, additional pieces of equipment, or new versions and releases of software products.

When the re-engineering of existing information systems is considered for whatever reason, its impact on the organization, the status of its functions and the technological quality of them should be considered before deciding on the strategy to be followed. The IS department is recommended to choose one of the four strategies listed below:

1. Extension of current functions and technologies
2. Shift to advanced functions
3. Shift to advanced technologies
4. Shift to advanced functions/technologies.

## 5.3 Software Re-Engineering

As the workload for the maintenance of information systems has increased – more than 90% of the IS resource in some enterprises – the productivity of software re-engineering has become a matter of primary concern. As a response to this, computer manufacturers and software vendors have begun to develop and provide reverse engineering and restructuring tools along with software metrics analysis tools which measure complexity and quality of the software. The outputs of the analysis tools can be used for making judgements on whether to re-engineer or develop software from scratch.

To facilitate re-engineering of information systems, various methods and supporting tools have been developed. They include functions for re-documentation aids, design

recovery, business rule extraction, and restructuring. It is desirable to have CASE tools generate higher level specifications (especially so-called "business rules") from the existing codes and facilitate changes to these specifications. This would accommodate the new requirements and lead to forward engineering.

Thus, software re-engineering technologies may very well assist business re-engineering. To facilitate not only program understanding and re-documentation, but also the recovery of designers' intents and plans, various research and development efforts are still under way. Ultimately domain and design knowledge and their easily accessible data bases are considered as keys to design recovery.

Re-engineering services are now offered by computer manufacturers and software vendors using these tools. One of the features of Japanese software re-engineering tools is the capability to convert mnemonic names of data items to the Kanji names which are used in daily business operations. Another recent research focus is how to extract business rules from source code. For example, Fujitsu has developed a research prototype to extract business rules from COBOL source code in a pragmatic way.

## 6. Latest Technology Observation

### 6.1 Technology Focuses for IS

Software engineering technologies are continuously growing and expanding. The latest focuses from the viewpoint of building and maintaining information systems are listed below:

- Re-engineering
- Reuse
- Information engineering (modeling)
- Object-oriented technologies
- Data dictionary and repositories
- Visualization
- Multi-media
- Artificial intelligence, especially knowledge-based systems
- Process programming, navigation
- Total life cycle support with CASE
- Open, distributed CASE environments
- Groupware: computer supported cooperative work (CSCW).

As stated above, in order to ease maintenance and salvage some of existing IS components, software reverse engineering and restructuring technologies are becoming increasingly important. They have also become essential components of integrated CASE environments because it is rare to build information systems completely anew and because there are so few application systems which don't have points in common with existing application programs, data bases, and so forth.

The object-oriented paradigm is not a 'silver bullet' to magically solve the software crisis problem, but it is now recognized as a new powerful paradigm which has the potential to improve productivity and quality by an order of magnitude. Its effect does not appear at its first trial in most cases, but the accumulation of classes and

experiences will enable the substantial improvement of quality and productivity in domains where this paradigm is best suitable.

Visualization at the analysis, design, and implementation phases, and various visual languages are contributing significantly to the easier understanding and faster construction of information systems.

Those who develop and maintain information systems want to have a truly integrated CASE environment, but customizable to their development convention and skill/experience levels.

As software products are now being developed by people scattered throughout remote locations, project coordination and management supports have become very complex. Several research and development efforts, as well as trial runs are being conducted using so-called groupware for CSCW (computer supported cooperative work).

## 6.2 Information System Technologies: the US vs Japan

The following are some comparisons of the technology level related to information systems:

The US seems to be maintaining a superior position in the following areas:

- OS and other S/W architectures and designs
- Analysis and design methodologies and supporting tools
- Development process architecture
- Large system project management
- IS education in universities.

On the other hand Japanese superiority areas are as follows:

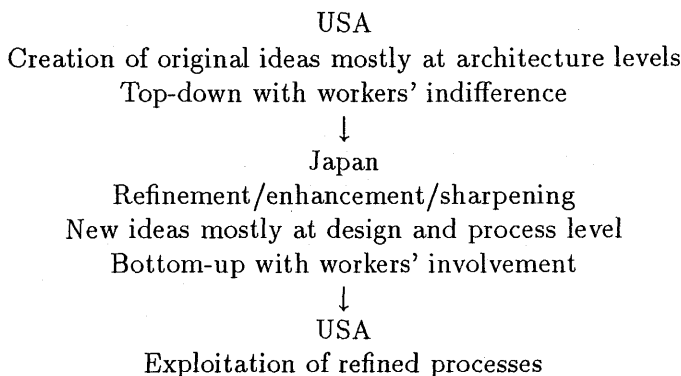
- S/W development processes (QC, reuse, visualization, testing, integration)
- Carefully thought out user friendly services
- Game software
- Consumer electronics
- Short development cycle.

The first item is more or less limited to JCMs and NTT, and not generally applicable to small software houses in Japan.

## 6.3 Technology Development/Transfer

The author has observed several cases where Japan borrows ideas from the US and developed products superior to US counterparts and export them back to the US. One excellent example is QC which was introduced to Japan by Dr. Deming in the 1950s. As Japanese industry faithfully implemented it, Japanese products greatly improved in quality and have threatened US industry. Recently the refined QC techniques have also been exported back to the US.

The following diagram depicts a typical technology creation and transfer formula between the US and Japan.



**Figure 2.** A typical technology creation and transfer formula between the US and Japan.

## 7. Conclusion and Proposal

### 7.1 Information System Technology Directions

As the amount of existing software assets which is totalling billions of lines of code, their protection and exploitation are essential and should be made possible by:

- Coexistence
- Re-engineering
- Reuse.

The reuse of software parts at the code level has been commonly practiced, but their granularity is so small that productivity gain is marginal. Large productivity improvement can be realized by the customized reuse of requirement specifications and automated code generation:

- Data and process models
- In the same domain or from a different domain
- AI + CASE.

The AI (especially knowledge-based system) technologies which are incorporated into CASE environments may greatly help in finding reusable parts and in customizing them.

Radically new information systems are expected to be created with multi-media will include the following:

- AV for faster, higher quality communication
- Hyper-media for faster retrieval of static and dynamic information on the same screen.

As experiences is gained and the merits and demerits are assessed, the re-centralization and re-integration of information systems may happen in not the so distant future. As a proverb says: - "The pendulum swings backs."

### 7.2 Proposal/Prediction

Like art, science and engineering have no frontiers. This also holds true for

information system technologies. As the exchanges of ideas and experiences are encouraged among countries and/or regions, particularly between the US and Japan, the productivity and quality of business processes in various enterprises can be improved faster and to a greater extent through mutual stimulation and research.

The US is superior in creating innovative ideas at the architecture/system design level as well as breakthrough paradigms and fundamental methodologies. Japan has proven herself to succeed in refining ideas and putting them into practical use, in manufacturing processes to produce high quality products and in shortening development cycles.

The author wishes to propose that through increasing personnel exchange and holding more international conferences and symposiums, the mutual utilization of each other's strong points will be accelerated to both sides' advantages.

In the author's mind this is the inevitable conclusion of recent developments and the most logical way to proceed.

American and Japanese companies have recently put a great deal of effort into in the development of multi-media technologies which will be used in the future information systems. The author is looking forward to faster progress and larger rewards as a result of international technical cooperation.

The following is the summary of those items which the author thinks should be promoted:

1. Making available information regarding an individual country's specialities for private research
2. International collaboration for evaluation and promotion of new paradigms
3. Optimum international division of labor to take advantage of country or regional strong points (creativity, skills, processes, efficiency, cost, etc.)
4. Standardization to facilitate open systems and portability
5. Reuse libraries containing reusable parts (including object-oriented classes) and easy-to-use mechanism
6. An international distribution mechanism for reusable software parts.

## References

1. Chikofsky, E. (ed.): Computer-Aided Software Engineering (CASE), Second Edition, IEEE Computer Society, 1993.
2. Davenport, T. H. and Short, J. E.: The New Industrial Engineering: Information Technology and Business Process Redesign, Sloan Management Review, Summer 1990, pp. 11-27.
3. Hammer, M.: Reengineering Work: Don't Automate, Obliterate, Harvard Business Review, Jul.-Aug., 1990, pp.104-112.

4. Information Processing Society of Japan (ed.): Information System Planning and Design (in Japanese), Baifukan Co., Ltd., 1991.
5. Krass, P.: Building a Better Mousetrap, Information Week, March 25, 1991, pp. 24-30.
6. Martin, J. and Finkelstein, C.: Information Engineering, Vols. I & II, Savant Research Studies, Lancasland, England, 1981.
7. Martin, J. : Object-Oriented Information Engineering, Prentice Hall, 1992.
8. Mii, N., and Takeshita, T.: Software Re-Engineering and Reuse from a Japanese Point of View, Information and Software Technology, Vol. 35, No. 1, Butterworth-Heinemann Ltd., pp. 45-53, 1993.
9. Takeshita, T.: CASE: Computer-Aided Software Engineering (in Japanese), Kyouritsu Shuppan Co., Ltd., 1990.
10. Takeshita, T.: Software Maintenance/Re-Development (in Japanese), Kyouritsu Shuppan Co., Ltd., 1992.
11. Takeshita, T.: Information System Paradigm Shifts, the Journal of the College of Business Administration and Information Science, Chubu University, Vol. 6, No. 1, pp. 17-30, 1991.
12. Takeshita, T. and Mii, N.: Trends and Directions of Reengineering Technologies as Essential Elements of CASE Environments, CASE '93 International Workshop, Singapore, July 1993.
13. Ulrich, W. M.: Business Re-engineering and Software Re-engineering: the Relationship and Impact, CASE Trends, Sept/Oct issue, Software Productivity Group, Inc., 1991.