Co-Evolutionary Service-Oriented Model of Technology Transfer in Software Engineering

Mikio Aoyama
Department of Information and Telecommunication Engineering
Nanzan University
27 Seirei, Seto 489-0863, Japan
mikio.aoyama@nifty.com

ABSTRACT

This article proposes a co-evolutionary service-oriented model, an organizational architecture for accelerating technology transfer between co-evolving organizations. Conventional model of technology transfer implies a unidirectional transfer from technology providers to technology requesters. However, this is not a case in successful technology transfer. In our experience, we observed a reciprocal feedback relationship between the providers and requesters. As the model of such collaboration, we propose a co-evolutionary model of technology transfer, which enables to adjust the level of maturity of technology to be transferred and accelerate the transfer process.

Categories and Subject Descriptors

D.2 [Software Engineering], K.7 [The Computing Profession]

General Terms

Software Engineering, Technology Transfer.

Keywords

Software Engineering, Technology Transfer, Service-Orientation, Co-evolution, Collaboration.

1. INTRODUCTION

Technology transfer has been recognized as critical issues in software engineering from the very early days [4, 5, 13]. However, we frequently hear pessimistic opinions that software engineering community lacks a confidence of successful technology transfer.

Several reports suggest the difficulty of the technology transfer and change the development process in software engineering. A survey report suggests that it takes 15 to 20 years to popularize new technology in software engineering [9]. Software CMM profile suggests 20 to 25 month of median time to move up one maturity level between level 1 to 4 [11].

On the other hand, we have experienced profound change in our computing platforms from mainframes, PCs, Client/Servers, to Web and embedded/ubiquitous devices. Such changes required the development of a set of new technologies in software engineering. Thus, practitioners have been repeatedly required to adopt such new technologies.

TT'06, May 22, 2006, Shanghai, China.

Furthermore, due to the high cost of labor-intensive software development, offshore development of software is becoming common. Even research facilities are distributed in multiple countries. These changes in research and development of software engineering make technology transfer more diversified and complicated.

The author spent 15 years in industry before moving to academia 10 years ago. This experience brought the author opportunities to play both roles of requesting and providing technologies, and involved many technology transfer programs.

By examining the experiences, this article proposes a set of technology transfer patterns in software engineering.

2. UNIDIRECTIONAL MODELS OF TECHNOLOGY TRANSFER

2.1 Unidirectional Models of Technology Transfer

A basic model of technology transfer is a unidirectional activity of transferring technology from technology providers to technology requesters. A survey of technology transfer in software engineering pointed out the following five models [8] illustrated in Fig. 1.

1) People Mover Model: People carry the technology and technical know-how, and move from providing organization to requesting organization.
2) Communication Model: Technology is transferred through documents or any form of publications.
3) On-the-Shelf Model: Technology is transferred by a product in a packaged system for ease of use [16].
4) Vendor Model: Technology vendor play as a gatekeeper of the technology.
5) Rule Model: Technology becomes a rule such as either *de fact* or *de jure* standards.

Here, we use the terminology of technology provider and technology requester in parallel with service-oriented architecture [7] since we view technology transfer is a kind of service.

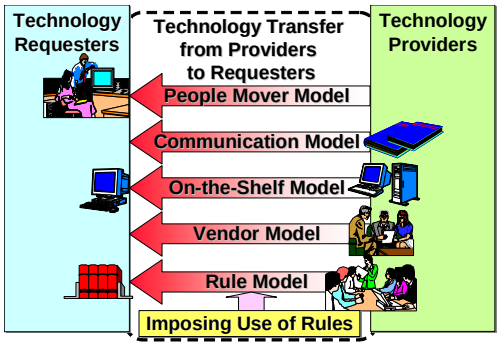

Fig. 1 Conventional Models of Technology Transfer

2.2 Anti-Patterns from Postmortem Analysis of Failed Cases

The author was involved in many technology transfer programs both inside a company and across the organizations. Some of them are considered successful while others are not. Here, two anti-patterns are described in a kind of pattern language from postmortem analyses of failure cases.

(1) Pattern: Maturity Gap

1) Context: Technology transfer from research lab to development team within a company

2) Structure of Technology Transfer: It intended to transfer advanced development methodology with support tool to an organization developing large-scale software systems.

3) Transfer Model: People mover model (A research people moved to software development team.)

4) Consequence: The technology transfer program was cancelled after a few years of painful efforts due to immaturity of the technology for practice and the unclear expected gain of the application of technology.

5) Major Causes of Failure: The technology transfer program is imposed by senior management with little assessment. Although research lab transferred one of the engineers developed the technology to the software development organization, the gap between providers and requesters was too wide to fill. Another critical problem is that the size of system under development is large-scale of millions lines of code. So, the risk of introducing a new development technology is extremely high.

6) Related Cases: Introduction of object-oriented technology to the organization with conventional development technique.

(2) Pattern: No Request:

1) Context: Technology transfer from an industry consortium to member companies

2) Structure of Technology Transfer: It intended to develop an integrated process for software procurement to be commonly used in the industry as a standard practice. Most of the technology elements are known best practices with some tailoring and improvement.

3) Transfer Model: Communication model through a set of guideline documents and manuals.

4) Consequence: After the completion of the development at the consortium, non of the participating companies introduced the technologies.

5) Major Causes of Failure: The consortium comprised only software developing companies. None of the customers, i.e. procurement authorities participated the consortium. So, the customers are not willing to introduce the technology developed by the developer consortium.

6) Related Cases: Unclear or unfocused conceptual frameworks.

2.3 Bidirectional Collaborations in Technology Transfer

These lessons learned indicate one crucial aspect in technology transfer of software engineering, that is, a need of right collaboration between providers and requesters. The five patterns of conventional technology transfer indicate that classification of the patterns is based on the media carrying the technology for transfer, such as people, documents, and system. The postmortem analysis emphasize on the importance of bidirectional collaboration between parties involving the technology transfer as follows.

a) In any technology transfer program, we should expect some inherent maturity gap between the technology providers and requesters. However, if the gap is too wide and hinder the right collaboration between two parties, such as Maturity Gap pattern, it will cause a serious obstacle to technology transfer.

b) Technology providers or transfer program should invite requesters from the beginning, just like a requirement acquisition process. One of the most influential requesters is customers of products developed by the technology.

3. BIDIRECTIONAL COLLABORATION MODELS OF TECHNOLOGY TRANSFER

We discuss patterns of bidirectional collaborations in technology transfer.

3.1 Two Categories of Collaboration Patterns: Direct and Mediated

Like any social relationships, collaborations in technology transfer can be either direct or indirect. Indirect collaboration is a mediated one through brokers or mediators. Here we use broker.

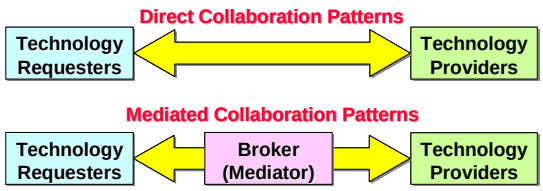

Fig. 2 Direct and Mediated Collaborations

3.2 Direct Collaboration Patterns

We classified direct collaborations in the following three patterns, as illustrated in Fig. 3.

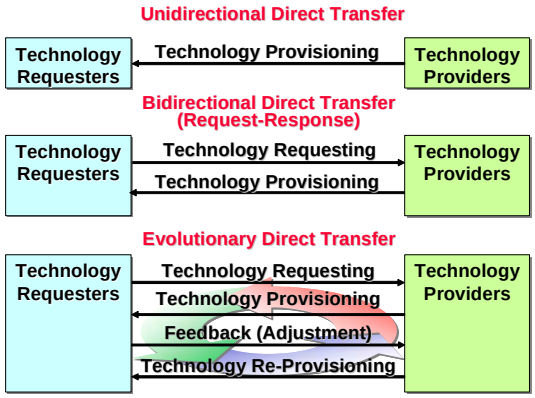

Fig. 3 Direct Collaboration Patters

1) Unidirectional Direct Transfer: Technology can be transferred from providers to requesters. This process is usually motivated by providers.
2) Bidirectional Direct Transfer: Technology can be transferred from providers to requesters by the request from requesters.
3) Evolutional Direct Transfer: After a process of technology transfer, requesters can provide some feedback to providers to adjust the technology transferred.

Unidirectional direct transfer assumes the transfer process is motivated by providers. Unless providers understand the real requirements of requesters, which is often difficult, this process causes miss-matching between requesters and providers.

Bidirectional direct transfer is initiated by requesters, followed by the response of providers. The transfer process is rather static in the sense that requests of technologies are clearly defined and fixed over time.

. In many cases, requesters are changing or evolving along with the change of external conditions including requirements to systems, and underlying platforms and engineering techniques. Thus, the technology transfer process should be evolutional.

One of the analogies of evolutionary technology transfer is organizational learning process. In successful learning process, evolutionary interaction with feedback is observed between trainers/teachers and trainees/students, such as fifth discipline for learning organization [12, 15].

3.3 Mediated Collaboration Patterns

We assume brokers or mediators between requester and providers so that the brokers can promote the transfer process by elaborating the expertise in technology transfer. For example, brokers can help requesters to find appropriate technology providers.

Similar to direct collaboration patterns, we classify mediated collaborations into the following three patterns, as illustrated in Fig. 4.

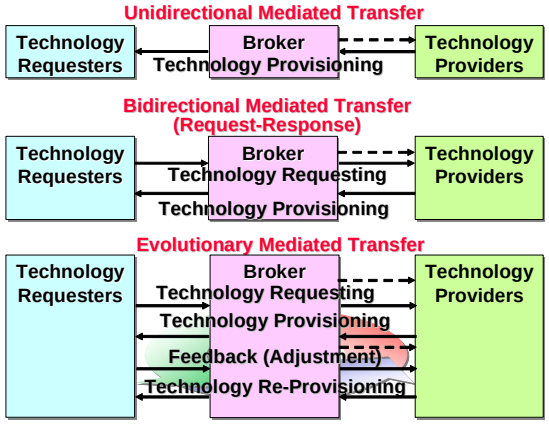

Fig. 4 Mediated Collaboration Patters

1) Unidirectional Mediated Transfer: Technology can be transferred from brokers to requesters. A critical difference from direct collaborations lies in that brokers can act as a proxy for a requester and proactively collect technologies for evaluation in advance.
2) Bidirectional Mediated Transfer: Similarly, technology can be transferred from brokers to requesters by the request from requester. This pattern is popularly observed.
3) Evolutionary Mediated Transfer: Collaboration between requesters and providers is enriched by two concurrent evolutionary collaborations: One is between requesters and brokers, the other one is between brokers and providers.

The mediated collaboration model enables more flexible and dynamic collaboration in technology transfer, similar to dynamic service coordination by service broker in service-oriented architecture [7].

4. CO-EVOLUTIONARY SERVICE-ORIENTED MODEL OF TECHNOLOGY TRANSFER

4.1 Co-evolution is Collaboration & Evolution

Collaboration structure can be found in a successful evolution process of business and its supporting information system [1, 6]. For example, a series of evolutions of CRS (Computer Reservation System) is such a case [3]. Development of new information technologies create new business model, such as yield management and frequent flyer program, which in turn demands further advancement of information technologies.

This leads to a successful evolution process of collaborating parties. We observe co-evolutionary process, which evolution of one party affects the evolution of the other party [14]. Unlike conventional unidirectional transfer model, we believe a co-evolutionary model is essentially necessary to successful technology transfer.

4.2 Co-Evolutionary Service-Oriented Technology Transfer Model

We propose evolutionary mediated transfer model as a co-evolutionary service-oriented technology transfer model for a

matured framework of technology transfer. As illustrated in Fig. 5, the model embodies the following key aspects of technology transfer.

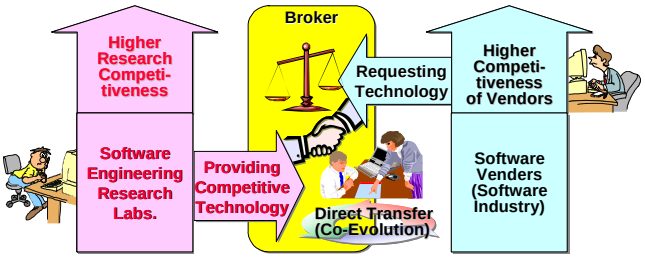

Fig. 5 Co-evolutionary Service-Oriented Technology Transfer Model

1) Service-Oriented Model:

We view technology transfer as a service provisioning from service provider to service requester. The underlying architecture of service-provisioning is publish/subscribe architecture [7]. In a simple collaboration pattern of publish/subscribe architecture is that providers publish their service information through any directory services then requesters look up the service endpoint from the directory service and invoke the service of the endpoint. This is collaboration is substantially initiated by requester, and considered as a unidirectional bidirectional direct transfer pattern.

To provide right services, a service broker can be introduced as a mediator who looks up appropriate services from directory services and provide a set of appropriate services to the requesters. Brokerage is especially effective where requesters and providers do not know each other. In this mediated bidirectional transfer pattern, brokers play an important role to mediate two parties, and to provide right services [2, 10].

2) Explicit Feedback Structure

Bidirectional structure of service provisioning can evolve into an evolutionary mediated transfer: with explicit feedback structure between two parties. Feedback can be either positive or negative. Positive feedback accelerates the effect, while negative feedback increases the controllability of closed system.

3) Double Spiral Structure

Requesters, brokers and providers are mutually influencing through the technology transfer and its feedback process. On the other hand, requesters, brokers and providers can concurrently evolve by their own. So, the three parties form double spirals of mutually evolving collaborations.

4.3 Service Broker Model for Accelerating Co-Evolutionary Technology Transfer

Provided technology is eventually evaluated by the end users who use the technology in the form of software products or services. Thus, we extend the co-evolutionary service-oriented technology transfer model to include end users in the transfer loop, as illustrated in Fig. 6.

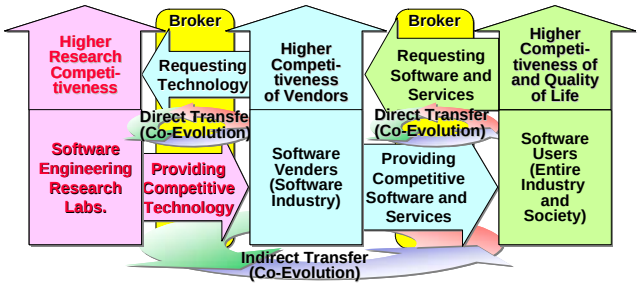

Fig. 6 Extended Co-evolutional Service-Oriented Technology Transfer Model

In contracted software development, which is popular in Japan, users often make influence in the selection of technologies to be applied in the software development. In such case, the collaboration between users and vendors can be either direct or mediated by so-called "IT planning division" in the user's organization. This pattern indicates the success of technology transfer is crucially depending on the collaboration between users and vendors, rather than vendors and research labs. From our experience, this model is well reflecting the various issues in technology transfer in contracted software development.

5. A COMMON FRAMEWORK AND ELEMENTS OF TECHNOLOGY TRANSFER

By combining the proposed collaboration-based patterns of technology transfer with conventional five media-based categories of technology transfer, we can map out representative, although not complete, elements of technology transfer techniques into a framework of Table 1.

Table 1 Framework and Elements of Technology Transfer

	Media	Unidirectional	Bidirectional	Evolutional
Direct	People	Technology Evangelists	Mentor	Joint Development
	Documents	Un-refereed Publications, Web Page	Collaborative Writing, Blog, FAQ	Collaborative Writing, Blog,
	System	Broadcast System	Help System, Helpdesk	Bulletin Board System
	Organization	Sales (Product Out)	Sales (Market In)	Direct Consultation
	Rule	Vendor-Forced Standards	Agreements	Community Standards
Mediated	People	Technology Evangelist	Conferences	Consulting, Workshops
	Documents	Refereed Publications	Wiki (WIkipedia)	Open Source
	System	Knowledge Base	Inquiry System	Expert System
	Organization	Brokers, Publishers	Brokers (Mediated Consortium)	Brokers (Mediated Consortium)
	Rule	Industry/De facto Standard	Community Standard	Community Standards

Although we still need to look into each element in Table 1, the framework may provide a comprehensive classification scheme for technology transfer techniques.

6. CONCLUSIONS

This article points out the importance of collaboration between participating organizations in technology transfer, and proposes several collaboration patterns in technology transfer. As a pattern for effective technology transfer, co-evolutionary service-oriented technology transfer model is identified.

Regard technology transfer as a service, service-oriented architecture can be applied to accelerate right technology transfer.

We will further explore the co-evolutionary service-oriented technology transfer model and its support environment based on the service-oriented architecture.

REFERENCES

[1] M. Aoyama, Continuous and Discontinuous Software Evolution: Aspects of Software Evolution across Multiple Product Lines, *Proc. IWPSE 2001*, ACM Press, Sep. 2001, pp. 87-90.

[2] M. Aoyama, Creation and Mediation of Trust for Web Services, *TR of IPSJ SIGSE*, Vol. 2002-SE-139, No.11, Oct. 2002, pp. 63-68 (In Japanese).

[3] M. Aoyama, Coevolution of Information Technology and Aviation: Emergence of Global Aviation IT Networks, *IPSJ Magazine*, Vol. 44, No. 12, Dec. 2003, pp. 1253-1259 (In Japanese).

[4] T. Bardini, Bootstrapping: *Douglas Engelbart, Coevolution, and the Origins of Personal Computing*, Stanford University Press, 2000.

[5] C. Jones, Why is Technology Transfer so Hard?, *IEEE Computer*, Vol. 28, No. 6, Jun. 1995, pp. 86-87.

[6] A. Y. Lewin, et al., The Coevolution of New Organizational Forms, *Organization Science*, Vol. 10, No. 5, Sep.-Oct. 1999, pp. 535-550.

[7] E. Newcomer, et al., *Understanding SOA with Web Services*, Addison-Wesley, 2005.

[8] S. L. Pfleeger, *Understanding and Improving Technology Transfer in Software Engineering*, DACS State-of-the-Art Report, SP0700-98-4000, 1998, http://www.dacs.dtic.mil /techs/techtransfer2/.

[9] S. T. Redwine and W. E. Riddle, Software Technology Maturation, *Proc. 8th ICSE*, Aug. 1985, IEEE CS Press, pp. 189-200.

[10] D. Rombach, Fraunhofer: The German Model for Applied Research and Technology Transfer, *Proc. 22nd ICSE*, Jun. 2000, ACM Press, pp. 531-537.

[11] SEI, *Process Maturity Profile*, Sep. 2005, http:// www.sei.cmu.edu/appraisal-program/profile/pdf/CMMI/ 2005sepCMMI.pdf.

[12] P. M. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday, 1990.

[13] M. Shaw, Prospects for an Engineering Discipline of Software, *IEEE Software*, Vol. 7, No. 6, Nov. 1990, pp. 15-24.

[14] J. N. Thompson, *The Coevolutionary Process*, University of Chicago Press, 1994.

[15] D. Ulrich, et al., *The GE Work-Out*, McGraw-Hill, 2002.

[16] N. Wilde, et al., *Web Support for Technology Transfer: Process and Mechanisms*, SERC-TR-80F, University of Florida, 1996, http://www.cs.uwf.edu/~wilde/publications TecRpt80F/TecRpt80F.html.