

Title	A structure of co-creation in an open source software ecosystem: A case study of the eclipse community
Author(s)	Mizushima, Kazunori; Ikawa, Yasuo
Citation	2011 Proceedings of PICMET '11: Technology Management in the Energy Smart World: 1629-1636
Issue Date	2011-08-01
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/10938
Rights	Copyright (C) 2011 PICMET. Kazunori Mizushima, Yasuo Ikawa, 2011 Proceedings of PICMET '11: Technology Management in the Energy Smart World, 2011, 1629-1636.
Description	

A Structure of Co-creation in an Open Source Software Ecosystem: A Case Study of the Eclipse Community

Kazunori Mizushima and Yasuo Ikawa

Japan Advanced Institute of Science and Technology(JAIST),
Graduate School of Knowledge Science, Ishikawa, Japan

Abstract—In an open source software (OSS) development community supported by spontaneous volunteers, personal interests, technical capabilities, hunger for fame, and the satisfaction of contributing to the public good are said to be motivating factor for participation. In that community, companies always play auxiliary roles, and integrate the result of OSS into their business activities.

However, in the Eclipse open source software community, the main role of OSS development activities is taken over by companies. The relationship between individuals and companies is reversed. Therefore, it becomes important to maintain the motivation of the development community, promote innovation and link the activities to the profit of a company. In other words, management of co-creation and competition are being conducted at the same time.

This paper tries to clarify a structure of co-creation in an OSS ecosystem led by companies considering the Eclipse community as one particular case. It also constructs a co-creation process model to promote sustainable development for an OSS ecosystem following two axes 1) value sharing and value acquisition, and 2) quantitative development and qualitative development. Some mechanisms that drive this process are embedded everywhere in the Eclipse ecosystem.

I. INTRODUCTION

The word “Ecosystem” has high visibility in the media nowadays. In this background, success in the ecosystem comes to mean success in business. In that, people who not only provide products and services but also build an ecosystem around them with stakeholders can lead their business to success by using market creation and the penetration power of the ecosystem.

In this sense, the stakeholders who participate in the ecosystem share a common destiny. The ecosystem as the common destiny develops competition with other outside ecosystems. Inside the ecosystem, participants simultaneously compete and cooperate with each other. Inside and outside of the ecosystem, complex relationships are constructed among participants.

On the other hand, it has been a long time since a software development activity called “open source” was widely noticed in the software industry. In an open source, a source code that is software intellectual property must be opened to the public, and developers all over the world develop software by cooperating with each other. Originally, an open source activity began spontaneously based on researchers’ and developers’ volunteer work. However, people have realized the power of open source that can build functions and qualities developed by volunteer developers equal to those of a business product,

and create new business opportunities since the success of Linux. It gathers momentum to leverage the open source community.

Eclipse, which this research targets, constructs an ecosystem by using the power of open source software. The aim of this research is to clarify the structure of co-creation in an open source ecosystem by investigating Eclipse as a case study.

A. Previous Research

Moore[1], [2], [3] introduced the concept of an ecosystem into management strategy or inter-organizational relation at the beginning and called it “business ecosystem.” The phenomenon in which competition and co-creation coexisted simultaneously in the relation among companies was recognized. The word “ecosystem” which was a term borrowed from ecology was applied to and explained for this phenomenon.

Iansiti and Levien[4] approached the business ecosystem using the knowledge of ecology or network theory. The role observed in a business ecosystem was classified, and also the indicator of the health of an ecosystem was proposed.

Eclipse is building the ecosystem by using an open source software as a core.

As for the open source, there are some researches which studied the open source strategy as an open innovation[5].

West and Gallagher[6] research the various licenses and business models in the open source, and summarize how value offer and value acquisition are balanced. The open source as an open innovation strategy is classified into four strategies.

Thomas and Hunt[7] explain the cause for the open source activity of a community basis to start, and why developers participate in. According to them, every open source project starts in order to satisfy a developer’s needs. Most reasons why developers participate in an open source probably boil down to a combination of need, pride, ambition, or fame of community.

West and O’mahony[8] studied 12 open source projects initiated by corporate sponsors, and identified three design parameters that help form a participation architecture. Johri, Nov and Mitra[9] investigated the effect of OSS steward company. By studying a case of MySQL, they found that the impact on participation depends on how the acquiring company was perceived.

These previous researches studied some factors to forming successful OSS community. In this paper, we discuss a structure of co-creation and business competition in Eclipse ecosystem, and clarify the mechanisms in which the factors work to form the ecosystem.

B. Research Questions

The purpose of this research is to clarify the structure of the co-creation Eclipse. Thus, research questions are as follows.

- RQ1 What mechanisms are embedded in the Eclipse ecosystem?
- RQ2 What is the reason to participate in the Eclipse ecosystem?
- RQ3 What is the developing process of the Eclipse ecosystem?

First, we shall clarify the mechanisms implemented in the Eclipse ecosystem from the side of both technology and system. Next, we show clearly through an interview how those mechanisms are working effectively. Finally, we will suggest a model of the developing process of the Eclipse ecosystem.

II. ECLIPSE ECOSYSTEM ARCHITECTURE

Fujimoto[10] tried to understand various corporate activities by applying to them the concept of the architecture. Fujimoto's architecture is a concept for understanding the character of a system, and he observes how to divide and how to tie a system. Namely, the architecture is a fundamental concept of how to divide a certain artificial system into components, how to allocate a function to each component, and how to design their interface where each component connects.

In this section, we clarify the OSS activity called the Eclipse ecosystem from the two aspects of "technology architecture" and "system architecture." The technology in the Eclipse ecosystem is software, and the system refers to a license, regulations and an organization. To explain these architectures, we introduce the purpose of the Eclipse ecosystem declared in the bylaw of the Eclipse foundation. Then, this purpose is implemented as the architecture of "technology" and "system."

A. Purpose of Eclipse

In section 1.1 of the Eclipse Bylaw, the purpose of the Eclipse technology and foundation is stated¹.

- Purpose of Eclipse Technology
To provide vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools (Eclipse platform).
- Purpose of Eclipse Foundation
To advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services.

In the Eclipse, a value acquisition (competition) and value sharing (cooperation) boundary is drawn up clearly, and the mechanism of promoting co-creation is implemented in both the "Technology" and the "System."

B. Technology Architecture

In a software development project, a development environment is tailored to a target language and system. Because the development environment is optimized for the purpose, there are some slightly different parts of functionality and also many common parts. Despite the common functions, different departments often develop a similar function separately for each programming language. And, in order to develop tools independently, they are not seamlessly integrated, and their usability will be worse.

IBM had a similar issue, so that providing a common tool platform gathered momentum. According to [11], Eclipse started to avoid dual tool development in November 1998. The common platform avoids dual developments, works smoothly together with tools built on the platform, and increases usability by providing basic operations.

Thus, the Eclipse software architecture of Fig. 1 was designed as follows:

- The common function is provided as a platform, and additional functions are implemented as plug-ins.
- The platform APIs are defined to use the common function.
- The plug-ins are developed by using the APIs. A plug-in development tool is also provided.

In 1998, Eclipse software architecture implemented the modular architecture which separated the platform and the plug-ins of IDE (integrated development environment). Eclipse 3.0 released in 2004 adopted the OSGi (Open Services Gateway initiative) standard framework for modularity. The OSGi framework is a module system and service platform for the Java programming language that implements a complete and dynamic component model. OSGi added a more flexible modularity management function to Eclipse. In addition, Eclipse 4.0 has restructured the Eclipse platform to develop plug-ins by using common Web technology such as CSS (Cascading Style Sheets) and JavaScript.

Gawer and Cusumano [12] point out the following:

For a modular architecture to encourage third-party innovation, the interfaces should be open.

Eclipse software architecture evolutions seem to intend to gather more participants than ever before and to promote innovation by adopting open standard module technologies.

C. System Architecture

This section describes how the system shown in Fig. 1 contributes to the development of the Eclipse ecosystem.

- Eclipse Public License (EPL)
- IT Infrastructure
- Development Community Support

¹http://www.eclipse.org/org/documents/EclipseBYLAWS2003_11_10Final.pdf

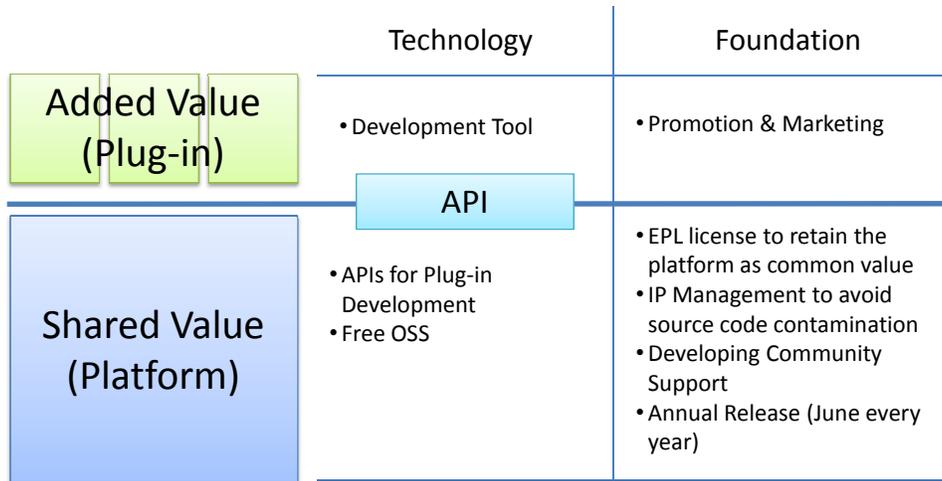


Fig. 1. Eclipse Platform Architecture

- Intellectual Property (IP) Management
- Ecosystem Development (Promotion and Marketing)

The above services except EPL are provided by the Eclipse Foundation and defined in the document "About the Eclipse Foundation²."

D. EPL and Value Sharing

This section explains the EPL, which is one of the open source licenses adopted by Eclipse, and show how it works as a mechanism to share knowledge and value in the Eclipse ecosystem.

1) License, Development Model and Business Model:

Before explaining the EPL, the relationships among the open source license, the development model and the business model are clarified. Software is a type of pure intellectual product that has no physical substance. A software license grants the use of software to users under certain conditions. In other words, the software license is a method to control intellectual property. Since open source software is a software, if its source code is disclosed, its intellectual property is also disclosed. Open source activity is a development activity of public intellectual property. In general, it is formed joint development. Thus, open source business is based on public intellectual property.

A software license has a significant impact on the open source development model and the business model. Generally speaking,

- A too closed license doesn't make the development communities,
- A too open license limits business opportunities.

Therefore, a well-balanced license is very important.

²<http://www.eclipse.org/org/>

2) *EPL*: We will show how the balance is maintained by the EPL. In general, the open source license prescribes recipient's the rights and obligations. The rights and obligations of the EPL³ are specified as follows:

- Rights:
 - A non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense, and royalty-free patent license.
- Obligations:
 - Providing software in source code form and distributing it with a license equivalent to EPL.

In the EPL, many software rights are permitted, and it is possible to execute, modify, distribute and sell it. Furthermore, it is the free of a licensing even if it is patented. That is, rights to distribute, extend and develop are fully granted.

On the other hand, there is an obligation to re-distribute a source code with the EPL. If a developer modifies the software licensed under EPL and newly create a derivative from it, the developer has to grant a royalty-free copyright license and a patent license to recipients of the derivative. However, this obligation is applied only when the base software is licensed with the EPL. If a developer creates an add-on software like Eclipse plug-in, one can apply any license, including a commercial license.

In Eclipse, by building a plug-in on top of the Eclipse platform, the value which Eclipse provides can be acquired and a developer's original value can be added in the form of a plug-in. New values by changing the platform, such as an improvement, are shared with others since the license employs the obligation to open the source code. That is, a boundary of value acquisition and sharing is clearly prescribed by the license, and the boundary brings a balance.

³<http://www.eclipse.org/legal/epl-v10.html>

E. Development Community Support and IT Infrastructure

Everyone who sees the success of the Linux open source must dream of utilizing the market creation power of open source, its high productivity and high development speed. However, it is difficult to start an open source project and form its community by oneself because of a deficient know-how.

Therefore, Eclipse prepares a support scheme to start a project. Support schemes are enumerated below:

- Stipulated incubation process of new project
The responsibility range of a participating company is clarified by this document.
- Established support organization
Responsibility and a role are made clear. The PMC (Project Management Committee) of a technology top project is responsible for an incubation project.
- Risk reduction measure
A mentor is assigned at the time of the establishment of a project, and helps in developing the project. Moreover, the project is to be reviewed at some checkpoints in the middle of an incubation process.
- Guarantee of openness and transparency
At the time the project established, a proposal is opened to public and its aim, scope, road map, etc. are clarified. Then, participants are collected. Moreover, a review result and the minutes are also opened to public on Web.
- Support of IT infrastructure
The Eclipse foundation is preparing IT infrastructures required for project management, such as the website of a project, a repository of a source code, issue and bug management.

Through the above supports, developers are not so busy about complicated community management, and can concentrate on technology development.

F. IP Management: Clearance Procedure

When developing a software product based on open source codes, it is difficult to avoid violation of a license and/or a patent. It is troublesome if the GPL license is mixing in open source codes. This is because all the source codes may be continuously contaminated by GPL if a GPL licensed source code mixes. Since the open source of the GPL license is actually used in some proprietary products, there are a number of vendors who finally open all the source codes of a product.

Mixing of the patent of the other company is also a troublesome problem. A patent infringement case occurs since the patent of the other company is unawarely mixed in the open source code used as the base of a product. SCO sued DaimlerChrysler and U.S. AutoZone which are the users of Linux noting that the patent of its company was used for Linux [13].

Since open source software is not necessarily developed from the scratch in a company, it is difficult to eliminate the possibility that unknown developers have copied and pasted a

source code of doubtful origin. Moreover, the OSS developer who introduce the patent into OSS will not be sued, but the vendor who uses and sells the software in which the patent mixed will be sued. Furthermore, since patent litigation requires time and expenses, once a patent litigation occurs, especially for a venture business, the impact on business is severe.

As mentioned above, contamination of a license and a patent is one of the severe problems for the vendor who considers to use open source to increase productivity.

In the Eclipse, in order to cope with this problem, the clearance of intellectual property is institutionalized. Thanks to this system, neither the source code of a GPL license nor a third party's patent mixes in the open source which Eclipse provides, and royalty-free use will be guaranteed even if the patent is contained.

In the Eclipse project, a source code is managed by a repository. A repository is a system which keeps and manages source codes. In the Eclipse, before committing a code into a repository, a developer has to pass through an examination procedure called "Eclipse Legal Process." This procedure prevents a code with a license other than of the EPL or a lawsuit risk code from mixing.

In this procedure, the source code is examined as follows:

- 1) In the case of a code 100% developed from scratch, the code will be committed with the EPL, since there is no room for mixing.
- 2) Also, when the source code is developed based on the EPL licensed code, the code will be committed with the EPL, since there is no room for mixing.
- 3) In other than the above, the Eclipse Management Organization (EMO) judges whether it can be provided with the EPL. In that case, it is registered in the IP tracking system, and log of the judgment and process is recorded.
- 4) When it is judged that the EPL cannot be applied, the source code is deleted.

In the above procedure, intellectual property-related clearance is carried out. Because of this procedure, a vendor is reassured and can use a source code without a lawsuit risk.

Janet Compbell who designed this IP managing process says, "It is designed to reduce a developer's load as much as possible."

G. Marketing and Promotion

In order to develop the Eclipse ecosystem, the Eclipse foundation conducts common marketing with member companies, and community conferences (EclipseCon and Eclipse Summit Europe) annually in the U.S. and Europe, respectively. In the conferences, Eclipse-related vendors and engineers gather from all over the world, and opinions on state-of-the-art technology and information of Eclipse are exchanged.

Moreover, the Eclipse foundation provides Eclipse Live⁴, which offers on-line seminars called Webinar, and Eclipse

⁴<http://live.eclipse.org/>

plug-in on-line catalog site Eclipse Plugin Central called EPIC⁵. These activities can increase the name recognition of Eclipse and of Eclipse plug-ins, and also present demonstrations of usages and descriptions.

Furthermore, when a member company releases an Eclipse-related product, the Executive Director of Eclipse will endorse its press release if requested.

H. Resonance: Simultaneous release

The Eclipse ecosystem clarifies the boundary line of value acquisition and of value sharing by the software architecture and the license. It also has incorporated a mechanism of co-evolution in which participants cooperate and compete simultaneously. So, what factor maximizes co-evolution of an ecosystem?

To maximize the co-evolution, it is important that the participants in an ecosystem synchronize and resonate mutually.

In open source software, the release date of the new version is uncertain in many cases. Even if the release date is decided, it is often postponed. It is difficult for participants to make a business plan for the release date. In the Eclipse platform, the version of the platform and the plug-in must synchronize because the plug-in depends on the platform API.

In the Eclipse, from Eclipse 3.2 in 2006, most of the projects synchronize and release the upgrade versions all at once every year at the end of June. The vendor developing an Eclipse-related product or service easily makes a plan by simultaneous release.

I. Summary of Architecture

We can provide the answer to RQ1 "What is the mechanism embedded in the Eclipse ecosystem?"

The software architecture of Eclipse adopts a modular structure and we can add new features to the platform as a plug-in by using an API. In addition, the structure of the modularization is evolving to adopt more open and standard technology. This is considered to have aimed at an entry cost reduction of plug-in implementation, while widening the developer base by adopting standard technology.

The system of the Eclipse foundation is described by the structure of Fig. 1 that participating IT vendors concentrate on technology innovation, and the Eclipse foundation supports other activities such as IP management, project management, and promotion as much as possible.

Furthermore, the simultaneous release for maximizing the effect of co-evolution among the participants in the Eclipse ecosystem is also implemented.

III. INVESTIGATION THROUGH INTERVIEW

As explained above, Eclipse is equipped with a mechanism which supports a participating vendor in both sides of technology and system. How do the participating vendors consider these mechanisms? In order to clarify this, interviews with

some vendors were held in EclipseCon 2009. The member companies of Eclipse which met in EclipseCon were targeted as interviewees. The interviews were carried out mainly with ventures.

As a result of the interviews, the following four points were discovered as participating motives of Eclipse.

- The Eclipse community is attractive as a market.
- A reduction of the development cost is expected.
- The IP managing process is serviced.
- The EPL is suitable for business.

A. Eclipse Marketability

Because of high marketability, there were a number of venture businesses which decided to participate in the Eclipse ecosystem.

According to the Eclipse Foundation report [14], the number of downloads is more than 1 million a month at the report time. Also he reports that Eclipse is used all over the world. Especially, it is actively used in Asia. It is natural that ventures are attracted by the number of Eclipse installed bases and worldwide usages.

ProteCode, one of Eclipse member companies, simply determined to participate in Eclipse by the difference in the number of user bases comparing with other open source Java IDEs.

Sopera, a German venture company of SOA (Service Oriented Architecture), said that the reason for having participated in Eclipse was as follows:

The most efficient way of approaching a user community was donation of the code to Eclipse. They start a SOA project in Eclipse based on their contributed source codes, and lead where the codes evolve. Their business is done by providing services like support, consulting, etc. instead of the software itself.

It is in the situation where huge competitors like IBM and Oracle already exist in the market of SOA. And it is difficult generally for a small venture to enter the market. However, it is possible for a small venture to enter the market if it contributes source codes to Eclipse as open source.

Actuate is one of the successful companies which participates in Eclipse actively. Actuate reported the market cultivation power of Eclipse in the session of EclipseCon 2009. Actuate is a vendor of BI (Business Intelligence) tool, and is developing a visualization tool for business data. In the participation in Eclipse, they contributed the engine part of the visualization as an open source. As a result, the visibility of Actuate improved. Actuate succeeded in penetrating the existing market and in expanding new markets.

According to Actuate's report, the number of downloads in 2008 recorded 6,500,000 or more. Only 1 % of the downloaded users became their commercial customers. Although 1% seemed to be small, since the number of total downloads was huge, it was realized as business. Actuate said it was able

⁵<http://www.eclipseplugincentral.com/>

to enter new markets in which Actuate had not done business. Actuate’s main market was the financial market, and it had hardly entered other industry markets. However, it could enter the new markets like merchandising, manufacturing, telecom business, etc., by contributing their source code to Eclipse as open source. Also, Actuate could gain new users in overseas markets, like in India and China, which they had not been able to enter.

B. Development Cost Reduction

There are two kinds of costs in software. One is developing cost for new functions, and the other is maintenance cost. The former is reducible by developing a common function jointly with other vendors. For the latter, a comprehensive test is achieved by getting as many users as possible. Besides, the more there are users, the more bugs are detectable. In maintenance, the bug patch cost is expensive, and the bug detection cost is also quite expensive.

Webtide, a venture of the Eclipse members, said it was determined to participate in Eclipse expecting to increase the number of developer bases. Jetty web server which Webtide is developing uses the same OSGi technology as Eclipse uses as the base. By participating in Eclipse, it could be said that a synergistic effect due to the increase in a developer base was expected.

On the other hand, there is also a venture which expected a reduction of the maintenance cost. Sopera is one of such companies. In the interview, the CTO of Sopera told that “By contributing our code as open source, many people are able to use it, and then many faults are discovered at an early stage.” He expected a 50% reduction in maintenance cost.

C. IP Management Process

There were some vendors which indicated the IP management process as a reason to join Eclipse.

As explained in II-F, IP contamination is one of the most severe problems for companies which use open source. However, the Eclipse foundation provides the IP management process which avoids the contamination of the codes. It eliminates vendor’s anxiety and gives a sense of security.

D. Summary of the Interview

Here we provide the answer to RQ2. There are three reasons for vendor participation: “marketability of Eclipse,” “development cost reduction,” and “IP management process.”

Vendors feel attracted to the marketability of Eclipse. This means obtaining profits from Eclipse, namely, they are trying to do “value acquisition” from the Eclipse ecosystem. On the other hand, a development cost reduction means cooperating with the participants in Eclipse. That is “value sharing.” The IP management process provides the exclusion of risk. That is a “sense of security.” These are summarized as follows:

In the following section, based on the interviews, we will construct a model of co-creation process implemented in the Eclipse ecosystem.

TABLE I
REASONS FOR PARTICIPATION

Marketability of Eclipse	→	Value Acquisition
Development Cost Reduction	→	Value Sharing
IP Management Process	→	Sense of Security

IV. CO-CREATION PROCESS MODEL

Fig. 2 illustrates a co-creation process model of the Eclipse ecosystem. To construct the process model, we consider two axes, that of value acquisition and sharing, and that of qualitative and quantitative development.

- Horizontal axis
 - Value Sharing
 - Value Acquisition
- Vertical axis
 - Qualitative Development
 - Quantitative Development

Quantitative development means that the scale of the ecosystem is increased by participating new vendors and users. In contrast, qualitative development means that values like new technology and knowledge are increased through innovation. Ecosystem is a growing network in which a variety of vendors are participating and building relationships with each other. To explain the developments from a network perspective, quantitative development means that the number of nodes increase, and qualitative development means building new relationships and strengthening relations between nodes.

We explain both quantitative and qualitative development processes in detail, and describe how these two processes work in order to keep the ecosystem active.

A. Quantitative Development Process

The lower cycle in Fig. 2 is the process of quantitative development. It is the process which leads to increase participants, especially vendors.

The details are explained as follows:

- 1) First of all, an out-of-box open source software is provided. In Eclipse, the development environment for Java corresponds to this. Eclipse is an outstanding software which is easy to install, user-friendly and which improves development productivity.
- 2) When the out-of-box software is provided, users come in the ecosystem to use it. To increase the number of users, it is also important that the software be easy to use. Eclipse is provided as a free open source, moreover, it can be downloaded from anywhere in the world. Therefore, the barrier for users to enter the Eclipse ecosystem is very low. If the software is outstanding, a user will invite other users to use it.
- 3) If users increase in number, a market will be formed and a business opportunity will be produced.

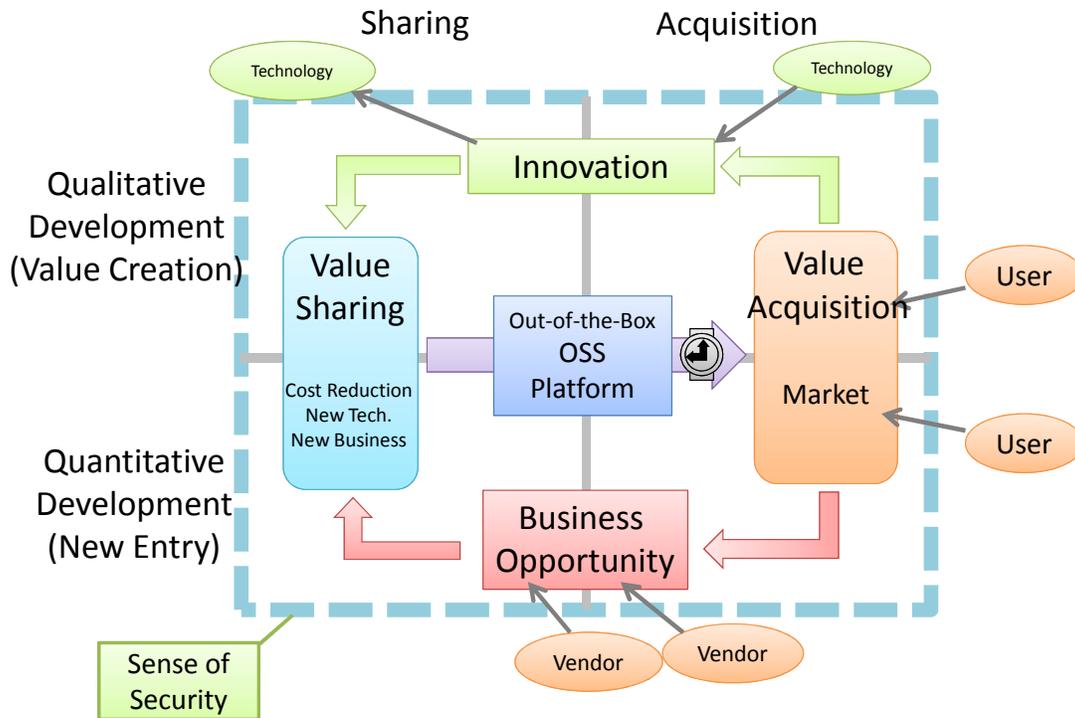


Fig. 2. Eclipse Ecosystem Co-creation Model

To catch this business opportunities, vendors such as niche ventures participate in the ecosystem, bringing new values like new functions and services. Some vendors donate source codes to Eclipse and launch an open source project aiming a market penetration and a reduction of development cost. In that case, as explained in the previous section, the project starting support service and the IP management process which the Eclipse foundation provides eliminate anxiety and give a sense of security.

4) Value Sharing

Value is shared while two or more participants cooperate with each other. In the case of an Eclipse ecosystem, the shared technical value is embodied in the form of software, and serves as public goods with a source code. In this way, the newly added value will attracts a new user to use it.

The above is a process of the quantitative development in an Eclipse ecosystem.

B. Qualitative Development Process

The cycle of the upper row of Fig. 2 is a qualitative development process. It is a process which amplifies new technology and knowledge. It can be said also to be a process of knowledge creation. In this process, users increase in number in quest of the outstanding open source, a market is formed, and business is performed for that market. And

profits are produced from business, and reinvestment in a new innovation is performed.

This detailed process is the following phases. The phase where users increase in number is the same as in quantitative development process.

- 1) Out-of-the-box open source software is provided.
- 2) Its users come in the ecosystem to use the software.
- 3) If the users increase in number, a market will be formed, and vendors which do business also come in the ecosystem.

The vendors appear to gain profits by targeting the users of this open source. They provide the commercial products which have additional features, or new services like consulting, education, etc.

- 4) An innovation creates new value.

The vendor who has been able to gain profits from the market seeks an increase in further sales, and invests in a new innovation. Or when technology is exhibited by the open source, a vendor, a university, or a research institute may appear from outside, and new technology which they already have will be introduced. It creates an opportunity of innovation. In this way, the innovations for the platform are returned to the ecosystem, and others can be the vendor's original added value.

Because the platform is provided with the EPL, any modification or innovation for the platform must disclose its source code. In this way, platform innovation will certainly

be returned to an ecosystem. It is so to speak, like a tax of the ecosystem.

If the tax is useful to others, it will contribute to reproduce others' worth. Because of this clever mechanism, the ecosystem keeps developing itself.

C. Summary of the Model

As one can see in Fig. 2, the upper qualitative development process and the lower quantitative process form a mirror structure around the open source platform. The important thing in this model is that the Eclipse ecosystem has a turning mechanism of the two upper and lower cycles which increases the whole value by reproducing the value of the ecosystem in each spin cycle. These cycles in the ecosystem exist in each project community. They are synchronized by the annual release explained in section II-H. The simultaneous annual release optimizes the rotation cycles.

This is the answer to RQ3 "What is the developing process of the Eclipse ecosystem?"

V. CONCLUSION

This paper clarifies the mechanism of an open source ecosystem development by investigating Eclipse as a case study.

We show that the Eclipse ecosystem draws a clear boundary line between value sharing and value acquisition in both of technology and system. As a result of interviews, we discovered that participating vendors are attracted to value sharing and acquisition, and sense of security of the ecosystem. We suggest a co-creation process model of the ecosystem.

This paper provides implications about the case where a vendor designs an ecosystem himself strategically, and the case where he participates in an existing ecosystem.

REFERENCES

- [1] J. F. Moore, "Predators and prey: A new ecology of competition," *HARVARD BUSINESS REVIEW*, vol. 71, pp. 75–75, 1993.
- [2] —, *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*. Collins, May 1997.
- [3] —, "Business ecosystems and the view from the firm," *ANTITRUST BULLETIN*, vol. 51, no. 1, p. 31, 2005.
- [4] M. Iansiti and R. Levien, "Strategy as ecology," *Harv Bus Rev*, vol. 82, no. 3, pp. 68–78, 2004.
- [5] H. W. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Pr, Mar. 2003.
- [6] J. West and S. Gallagher, "Challenges of open innovation: the paradox of firm investment in open-source software," *R&D Management*, vol. 36, no. 3, pp. 319–331, 2006.
- [7] D. Thomas and A. Hunt, "Open source ecosystems," *IEEE Software*, vol. 21, no. 4, pp. 89–91, 2004.
- [8] J. West and S. O'mahony, "The role of participation architecture in growing sponsored open source communities," *Industry & Innovation*, vol. 15, no. 2, pp. 145–168, 2008.
- [9] A. Johri, O. Nov, and R. Mitra, "'cool' or 'monster'?: company takeovers and their effect on open source community participation," in *Proceedings of the 2011 iConference*, ser. iConference '11. New York, NY, USA: ACM, 2011, pp. 327–331. [Online]. Available: <http://doi.acm.org/10.1145/1940761.1940806>
- [10] T. Fujimoto, A. Takeishi, and Y. Aoshima, "Business architecture," *The Strategic Design of Products Organizations and Process (Eds.) [in Japanese]*, *Yuhikaku*, 2001.
- [11] G. Cernosek, "A brief history of eclipse," 2005. [Online]. Available: <http://www.ibm.com/developerworks/rational/library/nov05/cernosek/>
- [12] A. Gawer and M. A. Cusumano, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Harvard Business School Pr, Apr. 2002.
- [13] "SCO sues AutoZone, DaimlerChrysler over use of linux," Mar. 2004. [Online]. Available: http://www.usatoday.com/tech/news/techpolicy/2004-03-03-sco-autozone-linux_x.htm
- [14] Eclipse Foundation, "Eclipse members' meeting," 2009. [Online]. Available: <http://eclipse.org/membership/slides.pdf>