Tracing the Development and Spread Patterns of OSS using the Method of Netnography†

– The Case of JavaScript Frameworks –

Kang, Heesuk* · Yoon, Inhwan** · Lee, Heesang***

<　　약>

The purpose of this study is to observe the spread pattern of open source software (OSS) while establishing relations with surrounding actors during its operation period. In order to investigate the change pattern of participants in the OSS, we use a netnography on the basis of online data, which can trace the change patterns of the OSS depending on the passage of time. For this, the cases of three OSSs (e.g. jQuery, MooTools, and YUI), which are JavaScript frameworks, were compared, and the corresponding data were collected from the open application programming interface (API) of GitHub as well as blog and web searches. This research utilizes the translation process of the actor-network theory to categorize the stages of the change patterns on the OSS translation process. In the project commencement stage, we identified the type of three different OSS-related actors and defined associated relationships among them. The period, when a master commences a project at first, is refined through the course for the maintenance of source codes with persons concerned (i.e. project growth stage). Thereafter, the period when the users have gone through the observation and learning period by being exposed to promotion activities and codes usage respectively, and becoming to active participants, is regarded as the 'leap of participants' stage. Our results emphasize the importance of promotion processes in participants' selection of the OSS for participation and confirm the crowding-out effect that the rapid speed of OSS development retarded the emergence of participants.

Key Words: Open Source Software, Actor-network Theory, GitHub, jQuery, MooTools, YUI, Rails, Netnography

논문접수일: 2017년 08월 10일　수정일: 2017년 08월 25일　게재확정일: 2017년 08월 28일

† This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2015R1D1A1A01059978)

* Engineering Growth, Green Factory, NAVER Corporation(First Author), nborn06@hanmail.net

** Associate Research Fellow, Future Strategy Research Department, Land & Housing Institute(Corresponding Author), ihyoon@lh.or.kr

*** Professor, Graduate School of Management of Technology, Sungkyunkwan University, leehee@skku.edu

Ⅰ. Introduction

For open source software (OSS), which anyone can access source codes, free modification and the comprehensive acceptance of licenses are guaranteed. With the advent of various success cases of OSS (e.g. Linux, Apache, and Eclipse), leading companies such as IBM, Oracle, and Redhat also have paid attention to and participated in the OSS.

Previous studies on OSS include diverse fields that range from the social phenomena related to the occurrence of OSS to the economic benefits and knowledge-sharing benefits of OSS. Earlier researcher focused fundamental research such as on 'What is OSS?' and 'What type of business model is it?,' and practical one for successful factors of OSS, which have continuously been investigated (Bonaccorsi and Rossi, 2003; West and Gallagher, 2006; Comino et al., 2007; Ke and Zhang, 2009; Subramaniam, 2009; Wang, 2012). Meanwhile, with the development of social media and the changes in the OSS operation method, a social network perspective or a community perspective has been used to study OSS (Madey et al., 2002; Grewal et al., 2006; Wu and Tang, 2007). In addition, many studies have paid attention to the motivation of participants such as by answering the questions 'Who participates in OSS?' and 'By what, for what and how do OSS participants participate?' (Hars and Ou, 2002; Di Penta and German, 2009; Bitzer and Geishecker, 2010). Although these studies found the operating patterns, successful factors, and the motivation of participants of OSS, they are limited to the change and characteristics of OSS operation tools.

For these reasons, this paper investigates the development and spread patterns of OSS from a longitudinal perspective, focusing on cases that involved JavaScript frameworks. We compare jQuery as the dominant OSS of JavaScript frameworks to others. The purpose of our study is to understand the characteristics of OSS by observing its multilateral change patterns, which cannot be examined in the cross-sectional studies. For this, this research utilizes an actor-network theory (ANT) and netnography as the longitudinal methods. The ANT is used as the foundation for organizing the time series analysis of OSS and the netnography is a research technique that combines the Internet and ethnography. Longitudinal studies of OSS have employed netnography that includes the study in which the term 'computer-aided ethnography,' which combined ethnography and visualization software in OSS project analyses, inspired by the 'computer-aided sociology' from the study of Latour (Teil and Latour, 1995; Ducheneaut, 2005; Cromie and Ewing, 2008; Hemetsberger and Reinhardt, 2009; Subramaniam et al., 2009; Sigfridsson and Sheehan, 2011).

The reminder of this study is organized as follows. In Section 2, the literature on OSS and ANT is reviewed. In Section 3, the methodology and data used in the OSS analysis were explained, including netnography. In Section 4, the development of the OSS and the change patterns of its participants were investigated using each ANT translation process. Finally, Section 5 provides conclusions and suggestions for further research.

Ⅱ. Literature Review

1. Open Source Software

Among the OSS operation tools, source forge types, which can configure the overall software development stages, project definition and product management from the idea stage and which enable bug tracking, were dominant (Riehle, 2009). In many previous studies, source forges have been commonly used as the data sources of the empirical studies (Comino et al., 2007; Subramaniam, 2009; Wang, 2012). However, previous studies are limited to be seen as the changes and characteristics of OSS operation tools. They focused on transient and immediate aspects such as the number of downloads and the number of participants, based on OSS communities. However, in the case of cross-sectional studies, the contributions of outsiders are not considered, and the perspective is the same as that of traditional software (Ducheneaut, 2005). The cross-sectional studies on OSS cannot examine the life-cycle aspects of OSS and are insufficient for examining companies' participation patterns and external influence factors as the representative cases of open innovation.

Git, which is an operation tool that combines source forges and Facebook, has been recently used as a distributed source controller and can obtain human-related data such as the relationships among participants and the activities within GitHub, in addition to source codes. As OSS has a collaboration type whereby records are kept in computers, the data change patterns can be visually examined using charts or graphs, based on online repositories. A source forge or GitHub provides Open API which can examine the data for each OSS. Based on the user information and commit information, status, issues (debugging), mailing lists, users' following relationships, and activities obtained from the Open API of GitHub, it is possible to perform time series analysis and longitudinal studies.

2. Actor-network Theory

The actor-network theory is a sociological theory that was first developed by Bruno Latour, Michel Callon, and John Law in 1981. From the perspective of this theory, non-human elements (e.g. objects and organizations) and humans are relevant to human actions, and human actors and their surrounding material environment are not separated from each other (Latour, 1986). This indicates that actors who correspond to participants, actants that correspond to source codes, and new actors that emerge as networks evolve have equivalent weights, and the process of change of these actors should be observed with that of the surrounding environment (Callon and Latour, 1981). From the perspective of socio-technical interaction networks (STINs) that were inspired by the ANT, it is also suggested that socio-technical networks consist of heterogeneous elements such as humans (including organizations) and equipment, data and various resources (money, technology and situations) (Kling, 2003). This research perspective indicates the limitations of examining transient data, which suggests that OSS-related networks are constructed and the success of OSS is determined either based on an internal element or an external element, or either from the source code itself or by an actor, as mentioned earlier (Scacchi, 2005; Erbizzoni et al., 2006).

Ⅲ. Methodology and Data Source

1. Netnography

In previous cross-sectional studies on OSS, the success of OSS is determined by using the quantitative values within the projects, and the successful factors or the elements have been deductively analyzed (Grewal et al., 2003; Ke and Zhang, 2011; Wang et al., 2012). These types of cross-sectional studies are useful to understand the success of OSS projects. However, they have limitations in that the relationship data between the OSS and developers and the sponsorship of companies are constantly changing. Therefore, it is important to understand the relationship between the measurement of OSS success and the characteristics of OSS depending on the time (Subramaniam et al., 2009).

Netnography is a new social science approach to the study of online communications through ethonographic research that combines participation and observation of online communications with new forms of digital and network data collection, analysis, and research representation (Kozinets, 2015). The term was coined by Kozinets, R.V. in 1998. Netnography is devised to facilitate efficient marketing by analyzing the habits of consumers in online communities and cyber-culture and can be regarded as a type of ethnography. It is also called ‘online ethnography’, ‘virtual ethnography’, ‘digital ethnography’, and ‘cyber ethnography’. In typical ethnography, a researcher directly enters the lives of the subjects to be observed (e.g. groups, organizations, tribes, or ethnic groups) and investigates their relationships geographically or historically. In contrast, netnography focuses on online communities or individual actors, and can observe subjects extensively by analyzing users in an online space, rather than subjects in a regionally restricted space.

Ethnography has flexible characteristics and performs observations in different situations and conditions. This is because the status of subjects, the surrounding environment and geographical situations are constantly changing. Netnography is also flexible, but a method used by a researcher could serve as a guide that can be delivered to others (Kozinets, 2002). In other words, the analysis of the act data of online consumers, which was studied by Kozinets, can be repeated by other researchers under the same conditions.

Researchers can access the log of the activities using the OSS, but it is difficult to find meaningful implications from much information. Therefore, in order to extensively examine the human elements of data (e.g. the participants) and the non-human elements of data (e.g. the source codes and email data), netnography is appropriate for use in the study of OSS's (Ducheneaut, 2005).

2. JavaScript Frameworks

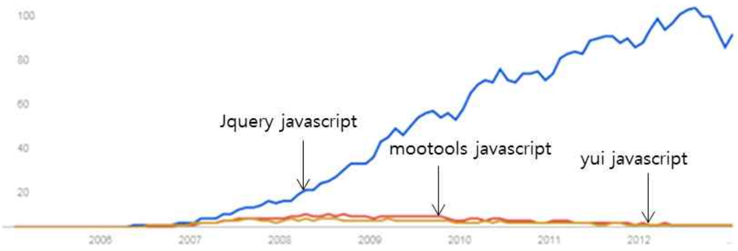
In this study, JavaScript cases (jQuery, MooTools, and YUI) were compared and examined. The three frameworks were first released in the same year, but their current market shares clearly differ. As shown in Figure 1, the interest levels of users in such OSS's were similar from 2006 to early 2007, when the three frameworks were first released, based on Google Trends results. However, jQuery has gone far ahead of the others since then (see Table 1).

For jQuery, its usage has steadily increased since its first release, and its market share exceeded 90% in 2013, among JavaScript frameworks. In contrast, MooTools and YUI had market shares of 7.7% and 2.5% in 2013, during which time the demand for them gradually decreased.

Among the top 10 OSS's (see Table 1), the repositories of Prototype, Script.aculo.us, and Dojo were found on GitHub, in addition to jQuery.

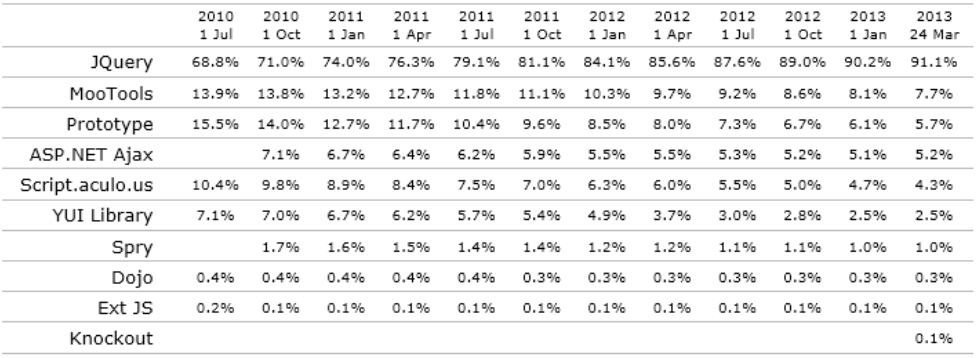
For jQuery, its usage has steadily increased since its first release, and its market share exceeded 90% in 2013, among JavaScript frameworks. In contrast, MooTools and YUI had market shares of 7.7% and 2.5% in 2013, during which time the demand for them gradually decreased.

Among the top 10 OSS's (see Table 1), the repositories of Prototype, Script.aculo.us, and Dojo were found on GitHub, in addition to jQuery, MooTools, and YUI. However, they were excluded from the comparison in this study because the number of commit participants, which was checked with Open API, was as small as from 10 persons to 20. The data from GitHub, from July 2010 (i.e. the commencement of the three OSS projects) to February 2013, were examined. For jQuery, we only identified relevant data from GitHub, starting from March 2006; for MooTools, from June 2008, and for YUI, from October 2007, respectively.



<Figure 1> Comparison of the results from Google trends for the jQuery, MooTools, and YUI

<Table 1> Market share trends of JavaScript framework



	2010 1 Jul	2010 1 Oct	2011 1 Jan	2011 1 Apr	2011 1 Jul	2011 1 Oct	2012 1 Jan	2012 1 Apr	2012 1 Jul	2012 1 Oct	2013 1 Jan	2013 24 Mar
JQuery	68.8%	71.0%	74.0%	76.3%	79.1%	81.1%	84.1%	85.6%	87.6%	89.0%	90.2%	91.1%
MooTools	13.9%	13.8%	13.2%	12.7%	11.8%	11.1%	10.3%	9.7%	9.2%	8.6%	8.1%	7.7%
Prototype	15.5%	14.0%	12.7%	11.7%	10.4%	9.6%	8.5%	8.0%	7.3%	6.7%	6.1%	5.7%
ASP.NET Ajax		7.1%	6.7%	6.4%	6.2%	5.9%	5.5%	5.5%	5.3%	5.2%	5.1%	5.2%
Script.aculo.us	10.4%	9.8%	8.9%	8.4%	7.5%	7.0%	6.3%	6.0%	5.5%	5.0%	4.7%	4.3%
YUI Library	7.1%	7.0%	6.7%	6.2%	5.7%	5.4%	4.9%	3.7%	3.0%	2.8%	2.5%	2.5%
Spry		1.7%	1.6%	1.5%	1.4%	1.4%	1.2%	1.2%	1.1%	1.1%	1.0%	1.0%
Dojo	0.4%	0.4%	0.4%	0.4%	0.4%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%
Ext JS	0.2%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Knockout												0.1%

(Source: W3Techs.com)

After the first release of the OSS, the time of the abrupt proliferation, the change in participants, and the beginning of new acts are presented in this paper in graphs. The influence factors were analyzed by organizing the external factors that belonged to the corresponding period. For the external factor data used in this study, the issues (e.g. conferences or events that were introduced in the OSS project blogs, sponsorship of companies, changes in project developers, release of new versions, and emergence of complementary goods) were organized for each period. As the observed OSS blogs steadily posted the issues and events within the developers, a considerable amount of data was obtained. Additionally, the external data, which were not obtained from the blogs, were acquired by searching Google Trends and web sites' news.

This paper organized the change processes of the three JavaScript frameworks based on the translation process of the ANT. The analysis of the change patterns of the OSS showed that some internal factors were affected by external factors. The external factors included the sponsorship of companies and the emergence of complementary goods, whereas the internal factors, which were affected by the external factors, included the activity of the participants and the number of participants, as well as the source code itself. We utilized the ANT to understand and accept the relationships among the factors. A series of OSS change processes can be examined by the translation process of the ANT. The translation refers to a process in which an actor affects other actors and combines with them to construct new networks.

In this study, each JavaScript framework was regarded as an actor, and its influence on other actors (actants) and the pattern of the translation process were examined. Previous studies also focused on the external networks that affected the success of the OSS. This indicates that from the perspective of the ANT, the success of an OSS is also affected by environmental factors (referred to as 'external factors' in this study) and not merely the OSS (Erbizzoni et al., 2006).

3. GitHub Data

This paper observed the OSSs based on the data that were obtained using GitHub Open API. GitHub is a website that hosts OSS projects using a distributed source code controller and enables practitioner to perform functions such as blogging, wiki publishing, features, and bug tracking of projects, and to control version using Git (Heller et al., 2011). In addition, most data, which are similar to the data that were shown on websites, can be obtained through Open API. Figure 2 shows the conceptual diagram of the use of GitHub. In this study, the change patterns of the OSS's depending on the time were examined based on three kinds of data, such as fork, pull request, and commit data.

We acquired the commit and committer information, the fork and pull request information, and the information on relevant operators that include masters through the Open API of GitHub. The sizes of and changes in the source codes were also checked at the GitHub site. Data were collected from the OSS commencement on GitHub, from March 2006 to February 2013. In order to trace the development acts (i.e. commit, pull request, and fork) for each time, the time for the emergence of users, and the types of users, we focused on the time and user data among the acquired data.

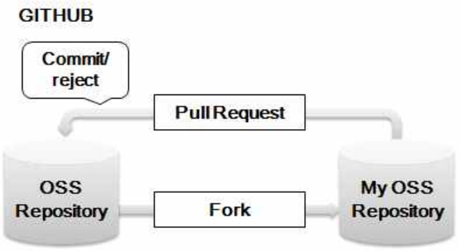
Ⅳ. Analyses

1. Translation of OSS

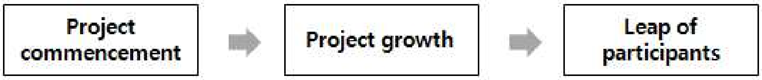
We utilize Callon's translation processes to examine the change patterns of the OSS development and participation. For this, they are divided into four stages (e.g. 3 translation stages and 1 generalization stage). Previous studies in the field of technology adoption (e.g. Zmud, 1982; Grover and Goslar, 1993; Damanpour and Schneider, 2006) divided technology adoption stages into 3 stages, such as initiation, adoption, and implementation. We suggest the OSS translation process (see Figure 3) that is the change pattern based on the OSS project, following information technology (IT) introduction stage, and present a research process (see Figure 4) that analyzes each translation stage,

1.1 Project Commencement

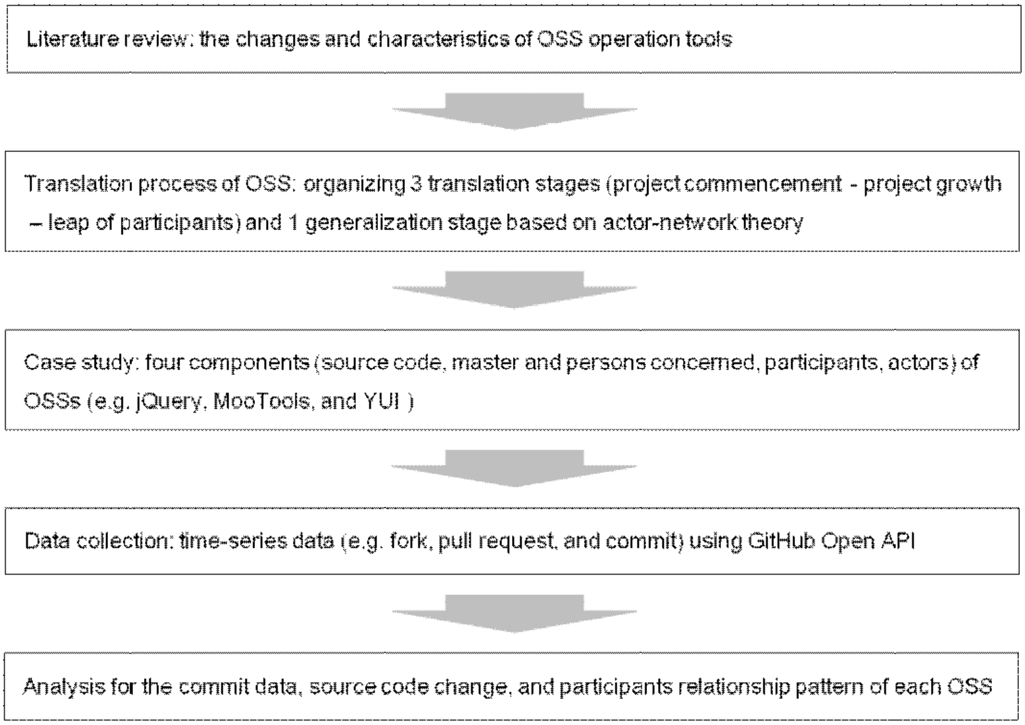
The initial stage is that OSS first emerges in the previously established OSS networks. This stage corresponds to 'problematization' in Callon's translation processes. In this study, the commencement of the OSS projects is considered as their first release on the Web, and actors are defined as follows, using the blogs(e.g. http://jquery.com, http://mootools.net, and http://yuilibrary.com), Google Trends searches, and web searches, in addition to the data obtained from GitHub.



<Figure 2> Use of jQuery on Github



<Figure 3> Translation process of OSS



Literature review: the changes and characteristics of OSS operation tools

Translation process of OSS: organizing 3 translation stages (project commencement - project growth – leap of participants) and 1 generalization stage based on actor-network theory

Case study: four components (source code, master and persons concerned, participants, actors) of OSSs (e.g. jQuery, MooTools, and YUI)

Data collection: time-series data (e.g. fork, pull request, and commit) using GitHub Open API

Analysis for the commit data, source code change, and participants relationship pattern of each OSS

<Figure 4> Research process

The four components of the OSS in this study are as follows. First, This is the source code that constitutes the OSS. Participants (e.g. master, participant, and company) go through different types of translation processes depending on the source codes. Source codes can be primarily changed by masters and participants who are directly participating in the project, and can be secondarily changed by companies or external participation.

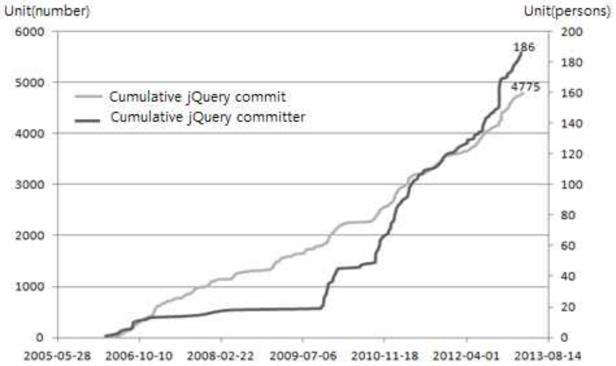
Second, an OSS is commenced on the web by a master and persons concerned. A master has the authority to manage sources and participants. Authorized persons besides the master (i.e. persons concerned) can also perform the management. However, the purpose of having a master is to ensure that the commenced source code will be modified and widely distributed for a long time and will also consider other people's interests.

Third, As users of OSS, internal participants express their interest in the sources commenced by an OSS master on the Web by starting a fork and through a pull request. They can freely modify and distribute OSS source codes, and participate and quit any time. In order to improve their abilities, they participate in projects based on their personal interests, and occasionally for their own satisfaction (Di and German, 2009; Ke and Zhang, 2009).

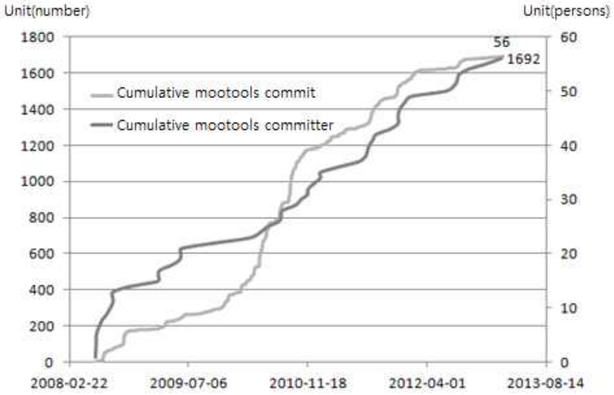
Finally, In the OSS operation process, the actors, such as companies and complementary goods, can be classified as external actors. Companies establish relationships with projects in the form of support (e.g. sponsorships) rather than by directly acting on the sources. Company-related participants can be involved in internal acts such as source commits; but in such cases, they are regarded as internal participants.

1.2 Project Growth

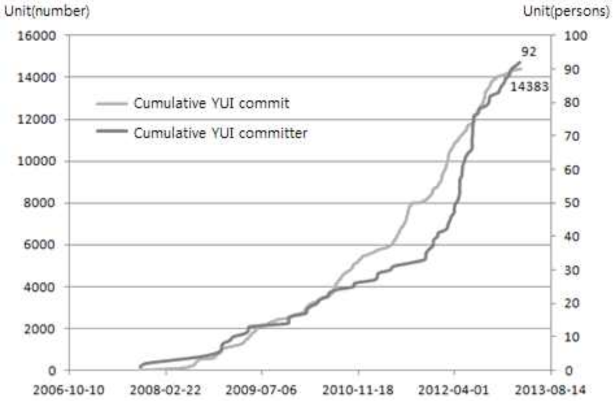
The project growth stage corresponds to 'interessement' in Callon's translation processes. In this stage, we discuss how other actors are separated from existing networks and how new negotiations are performed. The project growth stage was subdivided into the growth by promotion stage and the growth by development stage. Figures 5, 6, and 7 show the increasing trends in OSS participation through the number of commits performed by committers, from when masters commenced jQuery, MooTools, and YUI on GitHub repositories to February 2013. We suggest implications from the analysis results of cumulative graphs (see Figure 5, 6, and 7).



<Figure 5> Cumulative graphs of the committer and commit of jQuery (period: 2006.03 - 2013.02)



<Figure 6> Cumulative graphs of the committer and commit of MooTools (period: 2008.06 – 2013.02)



<Figure 7> Cumulative graphs of the committer and commit of YUI (period: 2007.10 – 2013.02)

1.2.1 Growth by Promotion

OSS promotion is an act that is performed continuously during the life cycle of a software program. Thus, we investigated the number of committers until its sudden increase. For MooTools, as no noticeable increase in the total number of committers was observed, we considered the sudden increase in the number of general participants as proxy of growth stage.

The main point of the growth by promotion is to observe the period that potential participants can be sufficiently acquainted with OSS projects. According to the technology acceptance model, the attitude formed by the perceived usefulness and perceived ease of use of the OSS affects its actual acceptance (Davis, 1989). The importance of an observation period can also be found based on the fact that the selection of OSS in which one first participated or the motivation for the selection of OSS in which one recently participated are significantly affected by the usefulness for participants and their technical interest (David and Shapiro, 2008). That is, time to acquaint OSS is a criterion of the usefulness for participants.

As for the external promotion activities for the three OSSs (i.e. jQuery, MooTools, and YUI), jQuery was the most actively promoted, and the scale of its promotion was also superior. jQuery and YUI began blog posting along with the first release of their source codes. They tried to promote the corresponding projects and to communicate with users. On the other hand, MooTools first released its source codes in September 2006, and began blog posting in May 2007. Since MooTools's perception for the importance of the communication with external users via the blog was rather late, blog posting on MooTools began in October 2007. Meanwhile, jQuery had active communication on the blog through questionnaire surveys and events such as user contests. This promotion attracted the attention of potential users and participants, and then jQuery had continuous education activities through workshops and regular meet ups with users, book publishing, and conferences. jQuery and YUI have been holding a huge conference yearly since 2008 and 2009, and have been respectively, actively staging education activities using pod-casts. In addition, their confidence and control level for IT users have increased as their communication with the users. These have positively affected the attitude of users towards the use of the corresponding IT.

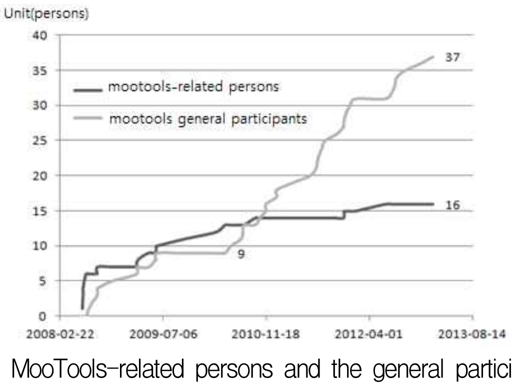
jQuery and YUI have abrupt proliferation of participants. However, a continuous increase in the number of participants in MooTools was observed, not noticeable proliferation. Before the abrupt leap in the number of participants due to the OSS promotion activities, we observed the 'plateau of participants' that the number of new participants hardly changed anymore. The plateau of participants can be regarded as the stage in which the users who entered the 'learning period' emerged from the potential users. In this stage, we found external participants who participated with a prior interest in OSS's that were similar to JavaScript frameworks, and majority (except innovators and early adopters) entered the learning period. This period can be regarded as the final phase of the project growth stage.

1.2.2 Growth by Development

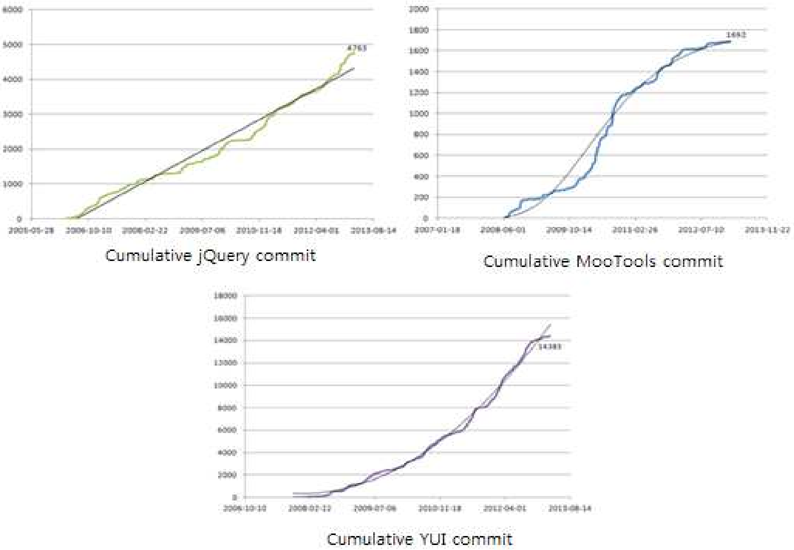
The project growth stage can be examined based on the dominant development stage, along with the promotion of the OSS. The OSS participant composition data indicated that the corresponding OSS-related persons participated more actively during this period. As for the composition of the jQuery participants, there were 186 commit participants in the entire observation period, from March 2006 to February 2013, with 34 jQuery-related persons and 152 general participants. The aforementioned jQuery-related persons refer to the development team members of the actual jQuery team, and we traced the corresponding data and the team member data of GitHub, based on the blog and the support organization of jQuery. We observed activities of 17 participants (about 9% of the total 186 participants in 2013) during the project growth stage that the number of participants increases, from March 2006 to September 2009. Among them, there were 11 jQuery-related persons and 6 general participants, and the corresponding project was led by the OSS-related persons.

In order to analyze the composition of the general participants in the project growth stage, we examined the relationships among the participants based on their profile data on GitHub, their affiliations and project interests, and their following information. The results indicated that the 6 general participants out of the initial 17 participants participated with an interest in JavaScript OSS's that are similar to jQuery. In Figure 8, we found the abrupt emergence of the MooTools-related persons in the early phase of the project commencement, whereas the abrupt increase in the number of the general participants was observed in May 2010. Figure 9 shows the trends in the development status based on the cumulative commit, from the commencement of each OSS on GitHub to the observation (i.e. February 2013). jQuery shows a linear development trend from the first commencement on GitHub, and its curve is the gentlest compared to that of MooTools and YUI. Meanwhile, a clear S-curve of MooTools indicated that its development reached a stable period faster than the other two OSS's. As for YUI, a super-linear trend represented the most development activities, reaching a stable period in late 2012. Therefore, this analysis contributes to the understanding of life-cycle curve of OSS.

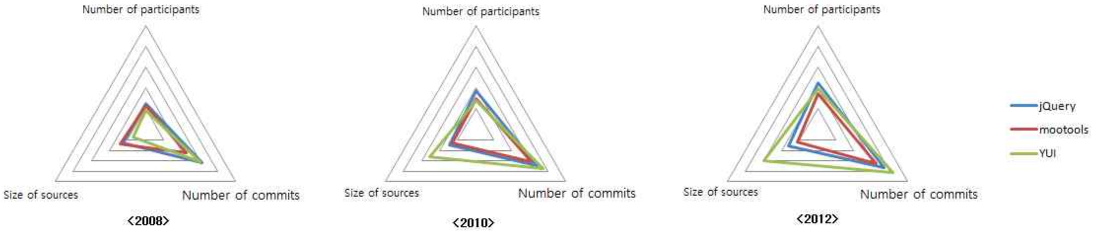
Figure 10 represents the number of participants depending on the number of source code commits and the change in the source codes for the three OSSs every two years. YUI had much more commits and much larger sources than jQuery and MooTools, but these were not directly connected to the number of participants. These suggest that large sources and frequent modification of sources cannot be regarded as the influential modular innovation that attracts participants. Therefore, we observed that many source changes were performed by only a few participants in YUI. We also found that YUI could not attract participants even though big source code changed.



<Figure 8> Emergence of the MooTools-related persons and the general participants(period: 2007.05-2013.02)



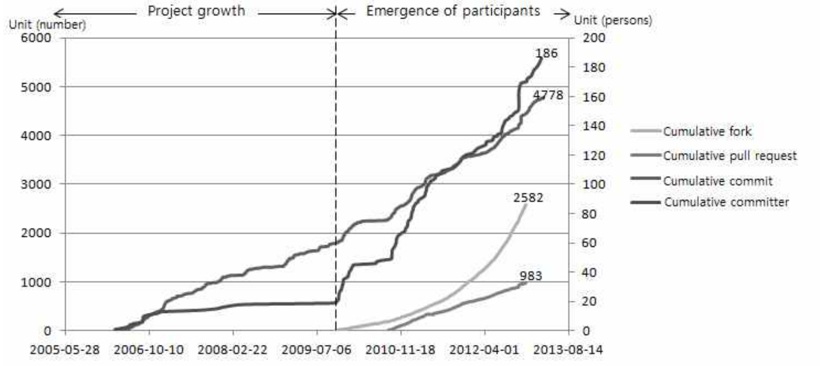
<Figure 9> Trends in the cumulative commit data for jQuery, MooTools, and YUI



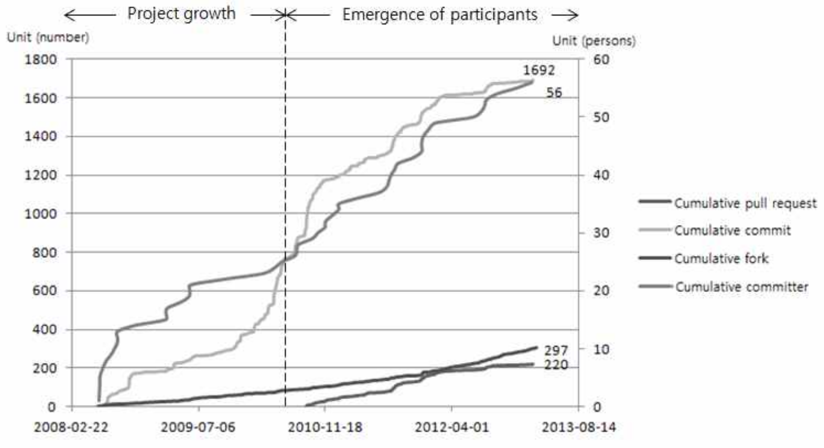
<Figure 10> Source code change and participants relationship pattern for jQuery, MooTools, and YUI

1.3 Leap of Participants

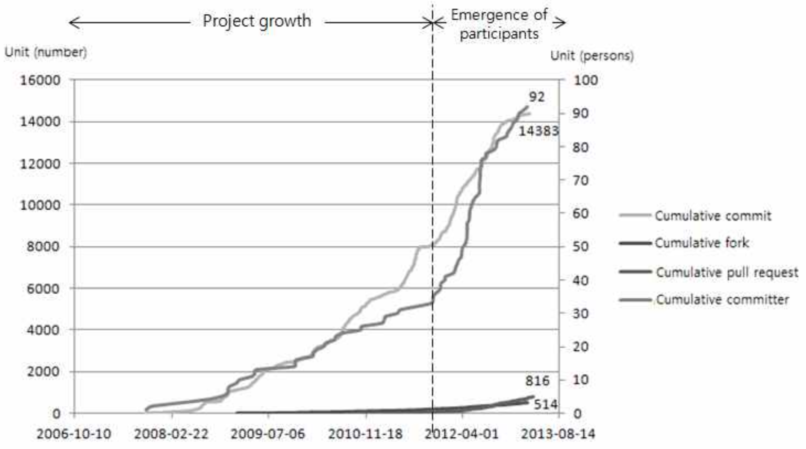
This stage that corresponds to the third stage (i.e. enrolment) of Callon's translation is designated as the stage for the active leap of internal and external participants. In this stage, the inflow of new actors into networks occurs based on the persons concerned and the potential participants who were changed in the project growth stage. We analyzed the spread stage of jQuery and YUI (see Figure 11 and 13), in which the active leap of participants occurred using the time for the primary proliferation of the participants. However, the spread stage of MooTools was analyzed based on the time for the proliferation of the general participants (see Figure 12). As shown in Figure 11, 12, and 13, the pull request increased after the increase in the number of participants, and this corresponded to the participation period of the potential OSS participants. The results highlight that the earliest proliferation of participants (i.e. jQuery case) is a key factor for the emergence of dominant JavaScript framework.



<Figure 11> Development and participation patterns of jQuery (period: 2006.03 - 2013.02)



<Figure 12> Development and participation patterns of MooTools (period: 2008.06 - 2013.02)



<Figure 13> Development and participation patterns of YUI (period: 2007.10 - 2013.02)

As for the cumulative committer data of jQuery, an abrupt increase in the emergence of committers was again observed in September 2010, during the period of the leap of participants. The emergence of jQuery Mobile in August 2010 was found to have influenced the second abrupt increase in the number of participants. During the period of the emergence of jQuery Mobile, there were distinct changes within jQuery and source changes. In other words, jQuery Mobile affected jQuery by serving as a new inflow path. Therefore, the expansion of inflow channels can positively affect the acceptance of jQuery.

The major affiliation of the participants changed significantly, from the persons concerned to the general participants. This may provide a evidence to confirmed by the fact that new external participants enter into the network and again create new networks when the persons concerned prepare the foundation of the OSS. This is closely related to the ideal 'open innovation' that an OSS pursues (Raymond, 1999). Additionally, during the leap of participants stage, the promotion activities in the growth stage positively affect both internal participants and external participants. Contrary to promotion activities, the speed of development activities has no influence on the leap of participants stage.

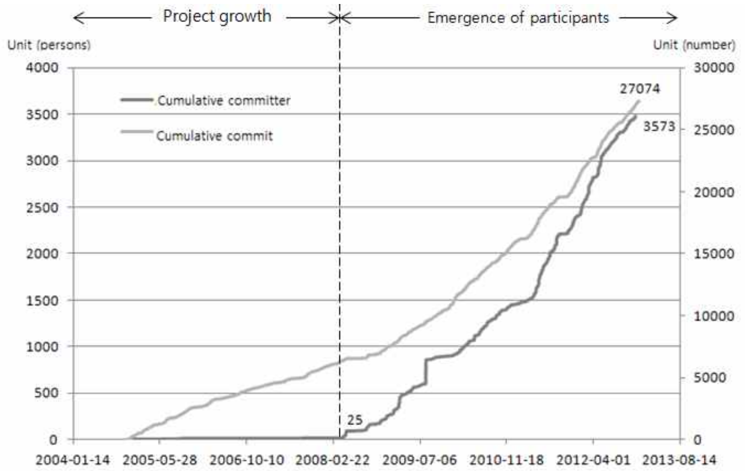
2. Generalization of the Project Case

In order to generalize our cases, we compared the translation process of jQuery to a different OSS framework (e.g. Rails). This process corresponds to the 'mobilization' of Callon's translation. Rails is the most mature web programming development framework for easy use of the Ruby programming language and has the advantage of creating a universal term, Ruby on Rails (RoR). Global web services, such as Twitter, GitHub (data source in this study), and Jobster, are implemented using RoR. This represents that Rails is appropriate for this work that compare with jQuery as the dominant OSS.

When comparing Figure 11 (i.e. jQuery) and 14 (i.e. Rails), the cumulative patterns of committer and commits are similar. In effect, the cumulative commit data of Rails have steadily increased after the project commencement, and diverse communication with participants has also been performed through the blog, similar to jQuery. As for Rails, education activities using pod-casts (e.g. http://podcast.rubyonRails.org/) have been very active, and the conference (e.g. RailsConf.) has been held every year.

Rails also showed the stage of the leap of participants after the project growth stage, and 'Phusion Passenger' is considered to be the influential factor. Phusion Passenger is a module for easy connection of Rails applications on the Apache Web Server. Phusion Passenger was officially commenced in April 2008 and was founded in the form of a company that provided the Ruby & Rails product service support and consulting from 'Phusion' in the Netherlands (http://www.phusion.nl/), rather than just a module. This is also similar to appendTo, which was launched to provide commercial support for jQuery and includes the role of the component innovation within Rails through external innovation. Phusion Passenger was found to have acted as the complementary goods of Rails and have led to the leap of participants.

Rails is a continuously growing and changing OSS. We found that Rails went through the translation process in a manner similar to the dominant design of JavaScript frameworks, based on the 7-year Rails data from 2006. Therefore, we insist that the case of the JavaScript framework in this study has the representativeness of OSS.



<Figure 14> Cumulative graphs of the committer and commit of Rails(period: 2004.11 - 2012.11)

Ⅴ. Conclusions

We argue that the aim of OSS is to attract various participants via open innovation and the stage of the leap of participants emerges by different factors. The findings of our cases show that continuous promotion and development activities are the most effective inducement to attract the direct participation of users. From the commencement of the OSS to the leap of participants, continuous communications with potential participants are essential by continuously maintaining source codes, blogs, and communities.

This paper articulates the translation process of OSS by utilizing the actor-network theory. The findings show that these processes sequentially continue from the project commencement to the project growth to the leap of participants. In the project commencement stage in which a master commences a project at first, by maintaining source codes with persons concerned, this stage proceeds to the project growth. The

OSS provides potential participants with external promotions, communications, and learning activities to acquaint with OSS projects in this stage as the dominant development stage. Additionally, we found open innovation on OSS by investigating the number of participants depending on the number of source code commits and the change in the source codes for the three OSSs. In turn, corresponding OSS-related persons participated more actively during this period. Thereafter, the period when the users have gone through the observation and learning period by being exposed to promotion activities and codes usage respectively, and becoming to active participants, is regarded as the 'leap of participants' stage.

This study provides the following practical and academical implication. First, our results confirm the importance of promotion processes in participants' selection of the OSS for participation and the crowding-out effect wherein the rapid speed of OSS development retarded the emergence of participants. Second, from the case of jQuery Mobile and Moobile, the OSS expands the inflow channels of new participants through the related diversification of the OSS. Finally, we found the usefulness of netnography to analyze the change patterns of an OSS based on empirical evidence, such as quantitative data (e.g. the passage time of three OSSs) and case studies. Additionally, netnography enables researchers to analyze diverse cases the same framework. We expect that our approach will extends and complements existing cross-sectional studies on OSS.

This paper has the following limitations. Due to the nature of an OSS, individual modification and distribution are possible without changing the original source such as GitHub, but this study is limited to investigate quantification for our cases. Although OSSs are market goods that are influenced by demand and supply, the level of our analysis is not of companies and organizations but of a individual. Thus, we suggest that the nature of an OSS and the characteristics of its translation processes are generalized in several categories through the analysis of diverse cases in further research.

References

1. Bitzer, J. and Geishecker, I.(2010), "Who contributes voluntarily to OSS? An investigation among German IT employees", *Research Policy*, 39(1), 165-172.
2. Bonaccorsi, A. and Rossi, C.(2003), "Why open source can succeed", *Research Policy*, 32(7), 1243-1258.
3. Callon, M. and Bruno L.(1981), *Unscrewing the Big Leviathan: How Actors Macro-structure Reality and How Sociologists Help Them to Do So*, in K. Knorr-Cetina and A. V. Cicourel eds., Advances in Social Theory and Methodology: Toward an Integration of Micro and Macro-sociologies, 277-303. London: Routledge.
4. Comino, S, Manenti, F. M. and Parisi,

M. L.(2007), "From planning to mature: On the success of open source projects", *Research Policy*, 36(10), 1575-1586.

5. Cromie, J. and Ewing, M.(2008), "Squatting at the digital campfire: Researching the open source software community", *International Journal of Market Research*, 50(5), 631-653.
6. Damanpour, F. and Schneiger, M.(2006), "Phases of the Adoption of Innovation in Organizations: Effects of Environment, Organization and Top Managers", *British Journal of Management*, 1(17), 215-236.
7. David, P. and Shapiro, J.(2008), "Community-based production of open-source software: What do we know about the developers who participate?", *Information Economics and Policy*, 20(4), 364-398.
8. Davis, F. D.(1989), "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly*, 13(3), 319-339.
9. Di Penta, M. and German, D. M.(2009), "Who are source code contributors and how do they change?", *Proceedings-Working Conference on Reverse Engineering*, 11-20.
10. Ducheneaut, N.(2005), "Socialization in an Open Source Software community: A socio-technical analysis", *Computer Supported Cooperative Work*, 14(4), 323-368.
11. Erbizzoni, E., Teli, M., Campagnolo, G., De Paoli, S. and D'Andrea V.(2006), *Free/Open Source ERP and Translation Processes: Four Empirical Cases, Research and Practical Issues of Enterprise Information Systems*, in Tjoa, A.M., Xu, L. and Chaudhry, S. eds, 695-704, Boston: Springer.
12. Grewal, R., Lilien, G. L. and Mallapragada, G.(2006), "Location, location, location: How network embeddedness affects project success in open source systems", *Management Science*, 52(7), 1043-1056.
13. Grover, V. and Goslar, M. D.(1993), "The Initiation, Adoption, and Implementation of Telecommunications Technologies in U.S. Organizations", *Journal of Management Information Systems*, 10(1), 141-163.
14. Hars A. and Ou S.(2002), "Working for Free? Motivations for Participating in Open-Source Projects", *International Journal of Electronic Commerce*, 6(3), 25-39.
15. Heller B., Marschner, E, Rsenfeld, E. and Heer, J.(2011), "Visualizing Collaboration and Influence in the Open-Source Software Community", *Proceedings-International Conference on Software Engineering*, 223-226.
16. Hemetsberger, A. and Reinhardt, C.(2004), "Sharing and Creating Knowledge in Open-Source Communities : The case of KDE", *Fifth European Conference on Organizational Knowledge*, Learning, and Capabilities, Innsbruck.
17. Ke, W. and Zhang, P.(2011), "Effects of Empowerment on Performance in

Open-Source Software Projects", *IEEE Transactions On Engineering Management*, 58(2), 334-346.

18. Kling, R., Kim, G. and King, R.(2003), "A Bit More to IT: Scholarly Communication Forums as Socio-Technical Interaction Networks", *Journal American Society for Information Science and Technology*, 54(1), 47-67.
19. Kozinets, R. V.(2002). "The Field Behind the Screen: Using Netnography for Marketing Research in Online Communities", *Journal of Marketing Research*, 39(2), 61-72.
20. Kozinets, R. V.(2015), *Netnography:* Redefined, London: Sage.
21. Latour, B.(1986), *The Power of Association, in John Law ed. Power, Action, and Belief: A New Sociology of Knowledge?*, London: Routledge.
22. Madey, G., Freeh V. and Tynan, R.(2002), "The Open Source Sofware Development Phenomenon: An Analysis based on Social Network Theory", *Proceedings-The Americas Conference on Information Systems*, 1806-1813.
23. Raymond, E.(1999), *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, Sebastopol, California: O'Reilly & Associates.
24. Riehle, D.(2009), "Open collaboration within corporations using software forges", *IEEE Software*, 26(2), 52-58.
25. Scacchi, W.(2005), S*ocio-Techinical Interaction Networks in Free/Open Source Software Development Processes*, Software Process Modeling International Series, 1-27, Springer US.
26. Sigfridsson, A. and Sheehan A.(2011), "On qualitative methodologies and dispersed communities: Reflections on the process of investigating an open source community", *Information and Software Technology*, 53(9), 981-993.
27. Subramaniam, C., Sen, R. and Nelson, M. L.(2009), "Determinants of open source software project success: A longitudinal study", *Decision Support Systems*, 46(2), 576-585.
28. Teil, G. and B. Latour(1995), "The Hume Machine: Can Association Networks do More than Formal Rules?", *Stanford Humanities Review*, 4(2), 47-65.
29. Wang J., Michael, Y. and Shanker, M.(2012), "Human agency, social networks, and FOSS project success, *Journal of Business Research*, 65(7), 977-984.
30. Wu, J. and Tang, Q.(2007), Analysis of survival of open source projects: A social network perspective", *Proceedings -Pacific Asia Conference on Information Systems 2007*.
31. West, J. and Gallagher, S.(2006), "Patterns of open innovation in Open Source software", *Open Innovation: Researching a New Paradigm*, 235(11), Oxford University Press.
32. Zmud, R. W.(1982), "Diffusion of Moderen Software Practices: Influence of Centralization and Formalization", *Management Science*, 28(12), 1421-1431.

Abstract

네트노그라피를 이용한 공개 소프트웨어의 개발 및 확산 패턴 분석에 관한 연구†

– 자바스크립트 프레임워크 사례를 중심으로 –

강희숙* · 윤인환** · 이희상***

본 연구의 목적은 공개 소프트웨어(Open Source Software, 이하 OSS)가 운영 기간 내 주변의 행위자들과 관계를 수립하는 동안 OSS의 개발 및 확산 패턴을 확인하는 것으로, OSS 참여자들의 변화 패턴을 조사하기 위해 OSS 통과시간을 기반으로 그 변화 양상을 추적할 수 있는 온라인 데이터와 네트노그라피 방법을 이용하였다. 이를 위해 대표적인 OSS 자바스크립트 프레임워크인 jQuery, MooTools, YUI 등 이상 세 가지 사례에 대하여 블로그, 웹 서치와 함께 GitHub 공개 API(Application Programming Interface)로 수집된 데이터를 활용하였다. 본 연구에서는 OSS 변형 과정의 변화 패턴을 분류하기 위하여 행위자-네트워크 이론의 전환(translation) 과정을 적용하였으며, 관찰된 OSS 변형 과정을 살펴보면 다음과 같다. 먼저, '프로젝트 개시' 단계에서 소스 코드, 프로젝트 책임자 및 관계자, 내부 참여자 등과 같은 세 가지 유형의 OSS 관련 행위자들을 확인하였고, 그들 사이의 관계성을 개념화하였다. 이후 프로젝트 책임자가 최초로 프로젝트를 착수하는 '프로젝트 성장' 단계는 관계자들에 의해 소스 코드가 유지 보수되는 과정을 통해 개선된다. 마지막으로 OSS는 홍보 활동을 통해 참여자들의 관찰기를 갖고, 소스 코드 사용을 통해 학습기를 거친 사용자가 본격적으로 등장함으로써 '참여자의 도약' 단계로 진입한다. 이 시기에는 기업과 외부 관계자들도 출현하는 모습도 살펴볼 수 있다. 본 연구 결과는 OSS 참여자들이 OSS를 선택하는데 있어 홍보 과정의 중요성을 강조하고, OSS의 급속한 개발 속도가 오히려 참여자의 출현을 지연시키는 구축 효과(crowding-out effec)가 발생하는 것을 확인하였다. 본 연구는 행위자-네트워크 이론을 토대로 주요 OSS 사례를 네트노그라피를 활용하여 종단적인 관점에서 분석함으로써 OSS의 발전 과정을 일반화시키기 위한 노력을 시도했다는 점에서 학술적인 의의가 있으며, OSS가 지배적인 위치에 오르기 위한 단계별 영향 요인, 세부적인 변화 양상 등을 확인함으로써 OSS 개발자와 관리자들에게 다양한 시사점을 제공할 것으로 기대된다.

핵심키워드: 공개 소프트웨어, 행위자-네트워크 이론, GitHub, jQuery, MooTools, YUI, Rails, 네트노그라피

† This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2015R1D1A1A01059978)

* 네이버 그린팩토리 Engineering Growth(제1저자), nborn06@hanmail.net

** 한국토지주택공사 토지주택연구원 미래전략연구실 책임연구원(교신저자), ihyoon@lh.or.kr

*** 성균관대학교 기술경영전문대학원 교수, leehee@skku.edu