

Boundary Objects in Hybrid Commercial/Open-Source Software Development Firms

Warren Allen

The iSchool at Drexel University
Philadelphia, PA 19104

warren.s.allen@drexel.edu

ABSTRACT

Describes a possible agenda for exploring the use of boundary objects in hybrid commercial/open-source software development firms.

Categories and Subject Descriptors

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – *organizational design, theory and models, computer-supported cooperative work.*

General Terms

Management, Human Factors.

Keywords

Boundary objects, free/libre open-source software.

1. INTRODUCTION

A minority of commercial software firms have developed complex relationships with free/libre open source software (F/LOSS) projects, complicating the rules and practices of entering the resulting hybrid community. Commercial software firms tend to be formal, hierarchically-organized, departmental, and highly procedural; F/LOSS communities, conversely, “resemble a great babbling bazaar of differing agendas and approaches out of which a coherent and stable system could seemingly emerge only by a succession of miracles.” [7] Current models of F/LOSS membership practices say little about the boundary objects used to facilitate the rules and practices of community inclusion and exclusion.

The motivation of the proposed research is to extend boundary object theory to the study of F/LOSS communities, and to organizations which span the commercial and open-source software paradigms. The incorporation of boundary object theory into existing F/LOSS models is suggested.

2. BACKGROUND

2.1 Boundary Objects

The concept of boundary objects, developed by Star [11] describes work artifacts that “sit in the middle” of diverse knowledge groups, establishing a “shared and sharable” context for distributed problem solving. [1] These artifacts are “both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites.” [11] An important distinction of

boundary objects from work product is that boundary objects are not the result of collaboration, but a means to facilitate it. They are “reminders that trigger knowledge” or “conversation pieces that ground shared understanding.” [3]

Carlile, adapting original boundary object categories described by Star, [11] identifies three types of boundary objects:

- *Repositories*, providing common reference point for across functions that provide shared definitions and values for solving problems.
- *Standardized forms and methods*, providing a shared format for solving problems across different functional settings.
- *Objects, models, and maps*, which are simple or complex representations (e.g., sketches, assembly drawings, parts, prototype assemblies, mockups, and computer simulations) that can be observed and then used across different functional settings. [1]

Boundary objects are used in conventional organizations to enable knowledge sharing between diverse knowledge groups. F/LOSS communities typically lack the organizational complexity that results in diverse knowledge groups. However, because anyone can engage the developers in a F/LOSS community, the members of the community “often have to take steps to manage their boundaries to serve people who need some service, are curious, or intend to become member.” [12] This boundary between skilled and non-skilled members of a F/LOSS community is an important site of discourse and knowledge exchange.

2.2 Legitimate Peripheral Participation

Artifacts at the peripheral boundary of a skilled knowledge group serve the community by facilitating inclusion and exclusion of outsiders. They do so first by enabling the learning and acculturation to practice of newcomers by facilitating *legitimate peripheral participation*. Newcomers to the community embark on an *inward trajectory*: “As the newcomer passes through the various stages of learning, he/she must necessarily connect with others performing actual practices.” [12]

Secondly, boundary objects in the knowledge periphery facilitate the exclusion of individuals who seek knowledge from within the group but who do not wish to become part of the community. Boundary objects allow an organization to provide information “without overwhelming the community itself with the task of accommodating outsiders’ demands.” [12]

2.3 The Onion Model of Free/Libre Open Source Software Community Building

Crowston and Howison hypothesized a “a hierarchical or onion-like structure” model of F/LOSS development. At the center of the onion (and top of the hierarchy) are “core developers” followed by an outer ring of patch developers and big fixers, followed by a ring of “active users” and finally an virtually unbounded population of passive users. [2] Research by von Krogh, et al. constructs a similar model of community-joining, adding the notion of a community-defined “joining script” which new members must follow if they wish to navigate towards the core of a F/LOSS community. [5] Finally, Herraiz, et. al. study a hybrid community consisting of F/LOSS and hired-for-pay developers, observing that the model for joining differed between the two groups as new members entered hybrid community. [4] Though the practices found in these models resemble inward trajectory and legitimizing peripheral participation, F/LOSS research has not explicitly studied the roles of boundary objects.

3. APPLICATION

When asked how a developer gets involved in an existing project, a former participant in the community for Snort, a popular open-source network security system describes an inward trajectory:

“Generally what happens is you’ll look at open bugs or feature requests and work on them, submit diffs to the current developers and they’ll commit the diffs. In addition [you’ll be] participating on mailing list discussions and such. Eventually someone will take you under their wing and give you commit rights. But every project is different, that is just a general way. Generally, what they look for is people that know the code base, people that know the direction of the project, and, obviously, talented developers and people that they work well and collaborate well with.” [9]

Boundary objects are often used to acculturate an individual to practices particular to a community. Looking at open bugs and actively engaging the community are common legitimizing activities for new members of a F/LOSS community. These tasks are an explicit phase of the Onion Model, describing necessary tasks in the joining script. However, little has been said about the role of boundary objects as part of the inward trajectory of a new member.

Boundary objects serve to facilitate knowledge transfer between incumbent members and new members of a community; they also manage the boundary that separates members and non-members. Queries from passive users and outsiders are redirected to the same artifacts, such as in this interaction on the Snort user forum:

Outsider post: For my present research, I need to know the architecture of the detection engine of snort. Could anyone of you please provide me with some documentation regarding the same.

Response, member 1: to be honest, it’s a [bellyache] i know, but a quick read of some of the source code might yield some answers to the question you are asking.

Response, member 2: read the lisapaper or look at the source. [10]

Two artifacts are mentioned here, and both are boundary objects. The source code is the human-readable version of the Snort application. Public availability of source code is a key feature of open-source software development, enabling skilled and innovative users to tailor and enhance existing programs or form a separate formal project. Source code is the work product of open-source and commercial software developers alike, but in F/LOSS communities, source code also serves as a repository of information about a software project.

The second boundary object – the lisapaper – refers to the paper presented by the creator of Snort, Martin Roesch, to the 1999 Large Installation Systems Administration (LISA) Conference. The paper is an urtext of the Snort community – it “discusses the background of Snort and its rules-based traffic collection engine, as well as new and different applications where it can be very useful as a part of an integrated network security infrastructure.” [8] As a (partial) account of the history of Snort, the lisapaper also articulates a single, shared narrative. Narrative is “a central mechanism by which social knowledge is conveyed, [and] provides a bridge between the tacit and the explicit, allowing tacit social knowledge to be demonstrated and learned.” [6] Thus, the lisapaper provides a socio-historical map of Snort, as well as a functional map of the prototypical Snort system.

4. RESEARCH AGENDA

4.1 Incorporate Boundary Objects into Free/Libre Open Source Software Research

Von Krogh, et. al., suggest future researcher might attempt to explain variance of joining scripts across F/LOSS projects. [5] Herraiz, et. al. raise issue with the Onion Model and its “static picture” of the rules and practices of community inclusion that are dynamic and evolving. [4] Some artifacts such as the source code will remain of consistent use and value over time for obvious practical reasons. Other objects – particularly those like the lisapaper which contribute to the socio-historical aspects and the narrative of the community – might waiver in and out of importance over the course of time, and be interpreted and re-interpreted differently according to the practice (commercial or commons-based) of an individual.

4.2 Extend Updated Free/Libre Open Source Software Model to Hybrid Communities

Commercial organizations will continue to develop relationships with F/LOSS communities. The Snort community, for example, is populated by volunteer bug fixers as well as employees of Sourcefire, the commercial organization started by Roesch. The core development team for Gnome – the F/LOSS community studied by Herraiz, et. al., [4] – consists of volunteer and hire-for-pay developers. There are two ways into this hybrid community: as a volunteer and as an employee of the relevant commercial firm, and research suggests the joining scripts are different for each. [4] The two paths into the core do not share in all their values or goals. They do, however, share boundary objects, making these artifacts worth understanding as they will become critical sites of discourse and knowledge exchange between the two sides of the hybrid community.

5. REFERENCES

- [1] Carlile, P.R. and Schoonhoven, C.B. A Pragmatic View of Knowledge and Boundaries: Boundary Objects in *New Product Development*. *Organization Science* 13, 4 (2002), 442-455.
- [2] Crowston, K. and Howison, J. The social structure of free and open source software development. *First Monday* 10, 2 (2005).
- [3] Fischer, G. and Ostwald, J. Knowledge Communication in Design Communities. *Barriers and biases in computer-mediated knowledge communication-and how they may be overcome*. Dordrecht, Netherlands: Kluwer Academic Publisher, (2005).
- [4] Herraiz, I., Robles, G., Amor, J.J., Romera, T., and Barahona, J.M.G. The processes of joining in global distributed software projects. *Proceedings of the 2006 international workshop on Global software development for the practitioner*, ACM (2006), 27-33.
- [5] von Krogh, G., Spaeth, S., and Lakhani, K.R. Community, joining, and specialization in open source software innovation: a case study. *Research Policy* 32, 7 (2003), 1217-1241.
- [6] Linde, C. Narrative and Social Tacit Knowledge. *Journal of Knowledge Management* 5, 2 (2001), 160-170.
- [7] Raymond, E.S. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, 2001.
- [8] Roesch, M. Snort Lightweight Intrusion Detection for Networks. *Proceedings of the 13th Conference on Systems Administration*, (1999), 229-238.
- [9] Sineath, B. Getting into an open-source project. Personal Communication. 2008.
- [10] Snort - the de facto standard for intrusion detection/prevention. http://www.snort.org/reg-bin/forums.cgi?forum_id=4&topic_id=6569.
- [11] Star, S.L. The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving. In *Distributed Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1989, 37-54.
- [12] Wenger, E. Communities of Practice and Social Learning Systems. *Organization* 7, 2 (2000), 225-246.