Open to Grok. How do Hackers' Practices Produce Hackers?

Vincenzo D'Andrea[1], Stefano De Paoli and Maurizio Teli[2]
1 Department of Information and Communication Technology
vincenzo.dandrea@dit.unitn.it
2 Department of Sociology and Social Research, University of Trento, Italy
{name.surname}@soc.unitn.it

Abstract. How do hackers' practices produce hackers' identities? In this paper we argue that the association between science fiction and software programs is rooted in hackers' practices, defining how hackers' knowledge emerge. The mediation is the one of the Heinlein verb “to grok”, part of the Jargon file and of the name of a code browser, OpenGrok, the technology mediating the relationship between OpenSolaris developers and the code base. Starting with a description of the peculiarity of the verb “to grok”, and its connection with a non-Cartesian view of knowledge, we discuss how the history of OpenGrok and its use by developers make this knowledge part of hackers' practices and identities, as someone involved in a true, deep understanding of software.

1 Introduction[1]

In this article we explore the importance of Science Fiction (SF) in organizing hackers’ practices of software development and identities. SF is often part of hackers’ everyday life and this claim can be verifiable thanks to several stories and technologies [2],[11].

Nevertheless, research focusing on the role of SF in terms of influencing hacker practices and identities has not been carried out. We maintain that, in hackers’ practices, SF has an active role which goes beyond any metaphoric use of it.

Before entering into our contribution we account for two domains of description: identities in FLOSS and the role of SF within organizational studies. The interest of organizational studies for SF is not new, nevertheless in many cases this interest has limited SF to the role of a source of concepts and metaphors, used in the interpretation of concrete cases [1].

For example Srinivas [13] used the concept of Android, inspired by Philip K. Dick’s novels. This concept represents the situation of a human being that is no longer entirely human, since he/she finds him or herself caged in a process of commodification. In such sense the contribution provided by Srinivas is that Dick’s android becomes a literary resource that can encourage the organizational subjects to withstand the process of “androidization” which subjugate them.

[1] We want to thank Camilla Rossi and the anonymous reviewer.

Even though SF has been found in hackers' story-telling, it has not been studied as a domain for the production of hackers' identities. Trying to understand "who is an hacker?", we found a rational approach to hackerdom, simplified by motivational research (“why do they do that?”), that forgets to consider how hackers' practices produce hackers, focusing only on why they are motivated to participate [12].

Against these uses we claim that the explanatory role of the SF results diminished in the above accounts, because SF is not only a source of interpretative concepts, but it can also be an active part of the organizational processes, contributing to the production of hackers identities. To deepen this claim we focus our analysis on the Martian concept of “Grok”, described in the novel “Stranger in a Strange Land” by Robert Heinlein [6]. In order to understand in which ways the concept of Grok participates in software development and use, we studied a software development project known as OpenGrok, paying particular attention to the practices that surround and constitute it.

2 Stranger in a strange land

The novel “Stranger in a Strange Land” narrates the story of Valentine Michael Smith (Mike), a human being raised by Martians on the planet Mars. The novel describes Mike’s interactions with the terrestrial culture after its return on Earth after his adolescence. Mike is the child of two members of the first earthling mission on Mars. Because of the death of his parents Mike grew up as a Martian. Mike can then be considered a “man of Mars”, in other words he is a human being in his physical structure but a Martian in his mental structure: “He's a man by ancestry, a Martian by environment.” [6, p. 20].

The second mission on Mars brought Mike on Earth so that he could be submitted to careful studies. Once arrived on Earth, the condition of Mike is that of a “Stranger in a Strange Land”, a Martian man among the humans that observes with Martian eyes the events of which he is protagonist. The novel is focused on the story of Mike, from his initial contact with the human beings, to his understanding of their culture, to the impact of his actions to the human society.

At the same time, in Heinlein’s novel emerges that the human beings attempt to understand the knowledge activity of Mike and especially how this activity is mediated by the Martian language. The Martian dictionary articulates around a fundamental verb that is "to grok". This verb, according to the Martian tongue, assumes a multiple meaning: to drink, to understand, to love or “to be one with” a person. To grok means therefore to know a thing or a person in his/her completeness, in the totality of his/her being and at the same time to become part of this thing or person; if we want to express this with a sentence: to grok a thing means “to be one” with that thing.

“'Grok' means to understand so thoroughly that the observer becomes a part of the process being observed [...] and it means as little to us as colour means to a blind man." [6, p.266]. Grok separates human beings from Martians, defining different identities and knowledge practices.

2.1 Knowledge and Grok

Why it is interesting to investigate the concept of grok? Firstly, this verb connotes a knowledge that is superior to that perceived or realized by an external observer. The relationship between the observer and the object of investigation is essential here: the knowledge realized by the man of Mars seems to be able to revert the perspective, typical of western modernity, of the separation among who observes and the object that is investigated.

Secondly, the knowledge realized in "grokking" does not seem to have a place in the breast of the Cartesian dualism between subject and object [3]: in grokking activities the observer and observed object become one single thing, which is not part of any of dualism. We can remember that the meaning of grok is to drink and it means to become one with the water that is drunk. Comparing the knowledge realized in grokking - in which a unity is realized between the person and the water - with that of the observer that investigates the water as external object, we can locate the verb to grok in a place at the opposite of the Cartesian perspective.

2.2 Grokking the code

The description provided by Heinlein is a point of departure to show the knowledge dynamics linked with the concept of grok. This task requires an empirical investigation and a discussion of how grok is articulated in practice.

For many aspects, a breakup in the relationship among the observer and an observed object can be found in those practices defined as hacking practices and in the relationship between hackers and calculators. According to the description offered by many [16],[14] this relationship doesn't seem to be instrumental, as that of an external observer towards an object .

Weizenbaum for example describes the differences between "professional programmers", which think of the calculator as an instrument that is separated from its user, and hackers, that only exist through and for computers [16]. The same rhetoric is presented in Turkle, for whom hackers see computers as "things in themselves", as opposed to the programmers idea of computers as instruments [14].

Given the consideration above, the verb "to grok" could provide an interesting way to describe hacking practices. Our goal is not to use the verb "to grok" for interpreting hacking practices, on the contrary we would like to observe how this verb is both active part of such practices and how our understanding of such practices would profit from an empirical inquiry.

According to the Jargon File, to grok "Connotes intimate and exhaustive knowledge. When you claim to 'grok' some knowledge or technique, you are asserting that you have not merely learned it in a detached instrumental way but that it has become part of you, part of your identity. For example, to say that you “know” LISP is simply to assert that you can code in it if necessary — but to say you “grok” LISP is to claim that you have deeply entered the world-view and spirit of the language, with the implication that it has transformed your view of programming.”[2]. Thus, a person that groks a certain technology (e.g. LISP) becomes “one with” that technology in a way that this act of grokking has changed radically his/her own personality. Hackers define themselves as involved in grokking software, and grokking involves hackers' identities.

3 OpenGrok: grok in practice

What said so far is not enough to justify our point of view on SF. To understand how the concept of grok participates to software development activities, we studied a development project called OpenGrok, focusing on the practices surrounding and constituting it. OpenGrok is the tool to surf the OpenSolaris operating system code base and, in this way, it participates to the developers' understanding of the OpenSolaris source code. In relation to these practices, the name OpenGrok reminds the association between surfing the code of a FLOSS project and the knowledge translated with the reference to Heinlein's novel. In the following part, we show how this association implies a definition/redefinition of the identities of actors involved, which are allowed by technological mediation to participate to this kind of knowledge; on the other side, we show how the same association between intimate knowledge and this program, is embedded in pre-existing practices and participates to their modification.

In order to study the OpenGrok case we chose to base our theoretical framework on the perspective of the Sociology of associations [7], [8], [9].

We follow the concept of “translation” considered as the interpretation given by technology builders “of their interest and that of the people they enroll” [7, p. 108]. This concept underlines how technological projects are built in attempt to involve more and more people in the projects.

In addition we must consider these practices as “practices of association” where developers build technology mixing together (associating) human and non-human elements.

These acts of enrolling and associating elements is possible thanks to technological mediation. In Latour [8] terms, mediators are entities that “transform, translate, distort, and modify the meaning of the elements that they are supposed to carry” (p.39). Following a mediator helps to understand the definition/redefinition of the elements involved into a translation.

From sociology of associations, we also assume the ethnomethodological perspective. Ethnomethodology [4] is a sociological discipline that aims at understanding the ways people make sense of their world and the ways they produce a shared social order. While other sociologies provide accounts of the social order

[2] http://www.catb.org/~esr/jargon/html/G/grok.html

which competes with that of the people who are members of the society, ethnomethodology aims at describing the practices these people use in their actual descriptions of the social order.

As such, the socio-technical order can be considered as an effect caused by the association between them, and sociology of translation aims at describing the practices that technology builders use in their actual construction of technology.

Finally, we assume that these practices of association are part of an ensemble of pre-existing practices, materials, and textual elements [9], that technologies are not built starting from scratch but they are embedded in pre-existing practices and participate to the modification of such practices.

Methodologically, we observed the discussions taking place in the mailing lists osol-discuss@opensolaris.org and opengrok-discuss@opensolaris.org, hosted at the OpenSolaris website, we analyzed the blog entries related to OpenGrok, and conducted three email interviews with three OpenGrok developers, the two main developers (CH, TR) and another one (KN). The interview outline has been constructed after the analysis of documents and mailing lists, according to a grounded theory approach [5]. Interviews and observation have been conducted during a two – years long cyberethnography [15] of the OpenSolaris project.

3.1 Toward OpenGrok: a security sentry problems

Initially, we have tried to understand how the association between OpenGrok and to grok emerged, and how it translated the meaning of the Heinlein's verb in practices different from the SF genre. OpenGrok has been developed by CH, a security sentry at Sun Microsystems. In his blog, telling the story of OpenGrok, CH writes: “I keep a watch on reports of newly discovered security holes and then check if any Sun software is affected by them”. The relationship between subject (developer) and object (software) involved practices of view and control; these practices involved the participation of another program, cscope, allowing a series of searches about the relationship between different functions of a program written in C language. Two elements of cscope were considered a limitation in the relationship between programmer and software: first, Solaris is a “Wad of Stuff”, with the presence of source files and binary ones; second, some of the programs observed didn't allow access to cscope indexes. These boundaries for the reproduction of the sentry status have been the input to create new relationships with the software, new kinds of relationship that brought to OpenGrok.

3.2 Toward OpenGrok: from collecting information to understand them

As shown, OpenGrok (formerly rob.pl) has been developed with the aim of enlarging the number of information available for programmers, particularly collecting text strings from binary files; information able to shorten the distance between what a program does (cscope) and what a sentry search. Obtaining this kind of information is not the unique element of “grokking the code”: the practice of control about the presence of security bugs is improved by a search engine able to order the information disposable. CH choice was to use Lucene, “The good thing about Lucene, is that it does not understand document content. You will have to write analyzers for your own content. So you have the control and freedom to interpret different kinds of files the way you want. Lucene does a good job storing your interpretation and searching it.” . The relationship between programmers and the code is mediated by the file interpretation that the same programmer considers relevant. The construction of information useful to reproduce the CH practices is not built as independent from the programmer, but the same programming practices participate to that. The boundary between the object and the knowing subject has been canceled through the mediation of a program which is rooted in and strengthens the unity between subject and object. It seems clear how the association between grok and this program has been sustained by a pre-existing practice, that has defined the need for it and has been redefined by this definition. What has brought to the stabilization of this association in the name “OpenGrok”?

3.3 Toward OpenGrok: relishing the code

It's the same CH, in an email interview, who describes the choice of the name: “We wanted a short, Google unique name [...]. A colleague suggested to use the word 'grok' as it means to fully understand something.” (emphasis added). In this statement, we find another practice, the use of the Google search engine to do web searches, pushing for a short name and an unambiguous recognition of the project. This practice is embedded in the developer field of action, the offices of Sun Microsystems, and the cooperative practices around projects, which are able to associate rob.pl+Lucene to a kind of full understanding, which is summarized by the word “grok”. The analogy with Heinlein story goes on when grok had to be declined to become an unambiguous name; it's still CH telling: “We had initially thought of 'Groktose' to rhyme with Fructose and Glucose. After some discussions with Marketing it became OpenGrok.”. When we asked: “which was the aim of this "sweet" flavour you wanted for your program?”, the answer was “You could through your source code in a tar ball and hope that public will catch it, or you can 'serve' it with OpenGrok and hope people will relish it :-)”. Other two elements emerge in connection with OpenGrok and the relationship between programmers and code: first, involving competences and people of the marketing area, the practices to spread OpenSolaris connect the project name with the same operating system, through the use of the prefix “Open”, used also in other Sun's projects; and second, the programmers conceptualization strengthen the idea of physical relationships, like Heinlein's “to drink”, that in this case become “to relish”.

3.4 Beyond Grok: OpenGrok redefines the concept

Something differentiates the SF concept from its practical and discursive actualization: if, according to Heinlein, to grok is a concept humans can't understand, constructing a boundary between Mike and human beings; according to CH, OpenSolaris developer, and the Sun marketing area, OpenGrok, and the possibility to relish the code, can be interpreted as a way to enlarge the number of participants to OpenSolaris , allowing a more direct relationship with the code. They put the full understanding of the system as basic requirement for participation and as a way to overcome boundaries between participants and not-participants, as the references to “public” and “people” make clear. At the same time, the public able to be attracted is not indistinguishable and undetermined, as another OpenGrok developer (KN) underlined in an interview: “Having a name that just says what the product does is too dull, so you try to find a name which both tells something about the product and gives some other associations”. The access to OpenGrok and OpenSolaris by the public is mediated by other elements: first, the capacity to interpret the name meaning, as connecting “to grok” to a full understanding; second, other practices can allow the contact between developers and the program, postponing the activity of attribution of meaning to the name. For example, in the KN case, being a Sun employee in Norway, engaged in the development of Apache Derby, as another OpenGrok developer, allowed KN to begin using OpenGrok without knowing the meaning of “to grok”: “In fact, I was not familiar with the word "grok" when I started using OpenGrok, so I needed to search for it to find out what it really meant. The exact meaning is still not entirely clear to me, but I think I've understood enough of it to say that it's probably a suitable name. :)”. We are able to evaluate another element which makes relevant the study of SF as an active discursive resource: the association between a program and a SF expression constitutes one of the ways through which the same meaning of what hackers are and do is redefined, and put developers in a learning process about “what it really meant”.

We can summarize the different ways through which the association “rob.pl + Lucene + grok” participates to hacking practices: a form of full understanding of code is part of the requirement of development practices, like the sentry ones. This knowledge translates in having programs that reduce the distance between programmer and source code, as the choice of Lucene shows; the association of this understanding with the word grok, on one side strengthen the physical relationship between programmer and code, like the suggestion of Groktose shows, on the other side it constructs boundaries around the practices that can produce subjects (hackers) able to participate to this knowledge, practices that do not require to interpret the name OpenGrok. In conclusion, grok associated to a program, translates a meaning rooted in pre-existing programming practices, it gives them new meanings connected with the relationship programmer/code, it also participates to establish how is it possible to access to the practices translated by the same “to grok”, and what an hacker is.

4 Conclusions

Our aim with this paper, has been to inquiry in which way SF is an active part of hacking practices, and how its translation as a discursive resource helps defining what an hacker is. We started from the consideration that previous studies looked at SF as a source of interpretative concepts in organization studies, without considering how this can be considered also an active part in organizing processes; and how those studies have considered identities as expressed in motivations, and not as a result of practices.

To empirically strengthen our argument, we have investigated the concept of "grok" as described in Heinlein's novel "Stranger in a Strange Land", observing how this represents a knowledge practice though which the observer gets blurred and becomes one with the observed object. The shift to an empirical field brought us to consider this kind of knowledge as part of the hacking practices, as described in the Jargon File, a discursive – conceptual resource for the hacker social world. In the following part, we have described an empirical case, the program OpenGrok: its birth; its lexical association with Heinlein's word; and its relationship of anchorage to a translation of hacking practices.

With this empirical analysis, we have shown how "to grok", taken as an example of the SF imaginary, participates to hacking practices by defining modalities of relationships between the source code of a program and developers, and being itself translated by the participation to the same practices: the hackers' grok allows humans to become Martians, that is to become hackers. Coming back to Heinlein's novel: «the words in English are a mere tautology, empty. In Martian they are a complete set of working instructions.» [6, p.490].

5 References

[1] A.A. V.V. (1999), Special Issue on Science Fiction, *Organization* 6(4), pp. 579-692.

[2] Crispin, M. (1978). Software Wars, http://panda.com/tops-20/software-wars.txt [13/12/2007]

[3] Descartes, R. (1637). *Discourse on the Method*, It. Trans. Il discorso sul metodo, 3rd edizione, Paravia.

[4] Garfinkel, H. (1967). *Studies in Ethnomethodology*. Englewood Cliffs. Prentice-Hall, NJ.

[5] Glaser, B. G & Strauss, A. L. (1967). *Discovery of Grounded Theory: strategies for qualitative research*, Aldine Transaction, Chicago

[6] Heinlein, R. (1961). *Stranger in a strange land* – Original uncut version, Penguin, New York, 1991.

[7] Latour B. (1987). *Science in Action: How to Follow Scientists and Engineers* Through Society, Milton Keynes, Open University Press.

[8] Latour B.(2005). *Reassembling the Social*, Oxford Univ. Press, Oxford, NY.

[9] Law J. (2004). *After Method:: Mess in social science research*. London: Routledge.

[10] Raymond, E. S. (ed.) (2003). The Jargon File. Version 4.4.7, http://www.catb.org/~esr/jargon/ [13/12/2007]

[11] Raymond, E. S. (1998). Unix Wars (version 1.1), http://www.catb.org/~esr/writings/unixwars.html [13/12/2007]

[12] Risan, L.. "Hackers produce more than software, they produce hackers," Version 2.1 http://folk.uio.no/lr isan/Linux/Identity_games/ [13/12/2007]

[13] Srinivas, N. (1999). Managers as Androids: Reading Moral Agency in Philip Dick, in *Organization* 6(4), pp. 609-624.

[14] Turkle, S. (1984). *The Second Self: Computers and the Human Spirit*, Simon, New York.

[15] Ward, K. (1999). Cyber-ethnography and the emergence of the virtually new community. *Journal of Information Technology*, 14, 95-105.

[16] Weizenbaum, J. (1976). *Computer Power and Human Reason*, San Francisco, W.H. Freeman.