

Knowledge Transfer in Open Source Communities of Practice

Diploma Thesis

TDT4735 Software Engineering, Depth Study

Spring 2005



Norwegian University of Science and Technology

Peter John Dalland Evans

Primary supervisor:

Eric Monteiro

To my dearest mom and dad; your inspiring academic achievements have given me the confidence I needed to believe in myself in a jungle of fancy words, intimidating authors, and strong opinions. As the offspring of two such amazing people, I figured at least something had to have rubbed off, although there were times I couldn't possibly see what that might have been.

Abstract

This paper discusses knowledge sharing dynamics in open source communities of practice based on an empirical study of an open source project. The paper describes how the online community in the study displayed many characteristics of an ongoing community of practice (Lave and Wenger 1991), as well as the distinct role technology and artefacts played in collaboration within the community. It is shown that while the theory of communities of practice captures many important aspects of learning and knowledge sharing in the project, it neglects the role of artefacts and the way they can contribute to these dynamics. Concepts of knowledge and knowledge transfer are discussed in order to explain aspects of these, relevant to the observations made in the study.

The purpose of the paper is to offer practical and theoretical contributions to understanding distributed knowledge transfer, as well as characteristics of open source development.

Key words: Communities of Practice, open source development, tacit knowledge transfer, role of technology and artefact, collaboration

Acknowledgements

First and foremost I would like to thank my supervisor professor Eric Monteiro for his constructive feedback. His insight is such that with a single sentence he could make me rethink my entire paper and indeed my fundamental beliefs and perceptions. With an educated glance he would dissect and analyse my writings, and his comments could pierce my heart or lift my spirits. I am also grateful for his understanding and patience with my somewhat unusual methods of work.

I would like to thank my friends and family for encouraging me with friendly cheers. I think it means more than I am aware of.

I am also grateful to ‘Nolar’ for letting me stay in the private Azureus developer channel so I could watch them play.

Lastly I would like to thank all the sceptics who looked at me strangely when I tried to explain what my paper was about, and for doubting in my ability to pull it off. Your doubt made me want to go that extra mile to prove you all wrong.

1. INTRODUCTION	8
PART 1: BACKGROUND THEORY.....	10
2. OPEN SOURCE	11
2.1 Open Source Development.....	11
2.1 Open Source Software.....	13
3. THE KNOWLEDGE CONCEPT	15
3.1. Background for the Knowledge Concepts	15
3.2 Explicit Knowledge	16
3.3 Tacit Knowledge.....	16
3.4 Discussion of Terms	18
4. KNOWLEDGE TRANSFER	20
4.1. Knowledge Conversion.....	20
Socialization: From Tacit to Tacit.....	20
Externalization: From Tacit to Explicit	20
Combination: From Explicit to Explicit	21
Internalization: From Explicit to Tacit	21
4.2. Knowledge Spiral	22
4.3. Communities of Practice	23
4.4 Knowledge Accumulation and Transfer	29
5. COLLABORATIVE TECHNOLOGY AND THEORY	32
5.1 Literature Review	32
5.1.1. CMC Literature	33
5.1.2. CSCW Literature	33
5.2. IRC	35
5.2.1 IRC Use.....	36
PART 2: RESEARCH SETTING AND METHODOLOGY	38
6. AZUREUS	39
6.1 The Azureus Program.....	39
6.1 The Azureus Project.....	39
Coordination and communication	40

7. RESEARCH METHODOLOGY	41
7.1. Why Azureus?	41
7.2. Classifying my Research	42
7.2.1. Qualitative and Quantitative Approaches.....	43
7.2.2. Flexible Research Strategies.....	43
Case study.....	44
Ethnographic study.....	44
Grounded theory study	44
7.3. My Research Approach.....	45
7.3.1. Being ‘Scientific’.....	45
7.3.2. The fundamental principle of the Hermeneutic Circle	46
7.3.3. The principle of interaction between the researcher and the subject.....	47
PART 3: ANALYSIS AND DISCUSSION.....	50
8. KNOWLEDGE TRANSFER IN THE AZUREUS PROJECT.....	51
8.1 The Azureus Community	51
8.2 Collaboration in the Azureus Project	54
8.3 Knowledge Transfer in the Azureus Project	55
8.4 Implications and Suggestions for Further Study	56
9. REFERENCES	58

1. Introduction

The open-source movement has recently attracted increasing attention; mostly because of its mere existence and that the way it works contradicts many existing theories and counteracts common business practices (Kollock and Smith 1997; Kuwabara 2000; Wayner 2000; Kogut and Metiu 2001; Lerner and Tirole 2001; von Hippel and von Krogh 2002; von Hippel and von Krogh 2003).

In open-source software projects expert programmers at different levels, supporters, and users voluntarily contribute to a collaborative software project that is administered via the Internet. They collectively develop software in a decentralized, self-directed, highly interactive, and knowledge intensive process (Raymond 1999; Kogut and Metiu 2001).

The use of knowledge requires the concentration of the knowledge resources at a certain space and time (Nonaka and Konno 1998). Internet technology can be used in various ways to pool and archive knowledge resources (Haythornthwaite et al. 1998), as well as to mediate communication and collaboration. The technology used for these ends provides the frame in which knowledge is concentrated and activated as a resource for creation. While Internet technologies are highly effective at facilitating the transfer of codified knowledge, it is considered difficult to share and create tacit knowledge online and collaborate on tasks with high complexity (Nemiro 2002). “Physical activities and face-to-face interaction are the key to sharing tacit knowledge.” (Nonaka, Reinmoeller et al. 2000).

The theory of communities of practice (CoP), first introduced by Lave and Wenger (1991) has gained significant ground in recent years. This theory is grounded in an anthropological perspective that examines how adults learn through everyday social practices rather than focusing on environments that are intentionally designed to support learning. A community of practice is defined as “a group of people who share an interest in a domain of human endeavour and engage in a process of collective learning that creates bonds between them” (Wenger 2001, p.1).

As this paper intends to show, open source projects can display characteristics of an online - or *open source community of practice*. Yet despite the increasing research effort into such communities (Tuomi 2001; Lanzara and Morner 2003; Gray 2004; Sharratt and Usoro 2003),

existing literature nevertheless leaves us uninformed about how knowledge sharing and creation processes develop at the interface of technology and communal structures that effectively exploit the advantages of Internet technology and at the same time are able to overcome the problem of tacit knowledge transfer (Hemetsberger et al. 2004).

This paper discusses how tacit knowledge transfer takes place in an open source project named Azureus. It is argued that central in this regard is a cyclical interaction between social tacit knowledge transfer mechanisms in CoP theory and tacit knowledge transfer via artefacts, and this interaction is offered as a success factor in open source projects.

Part 1 in this paper offers a relatively extensive discussion of background theory deemed necessary in order to be able to explain the observed phenomena in the study. Special emphasis is put on elaborating knowledge concepts and knowledge transfer theories. In chapter 5 of Part 1, I discuss the role of collaborative technology and its related theory, and in summary elaborate on my choice of angle for this paper.

In Part 2, I describe the object of my study as well as the manner in which I conducted my research and the motivations behind my actions.

Part 3 concludes the paper with a discussion on knowledge transfer dynamics in the Azureus community, as well as offering implications and suggestions for further study.

PART 1: BACKGROUND THEORY

2. Open Source

This chapter is intended as a brief description and background of open source (sometimes called “free” software) software and development in order to provide some context for my empirical study. I base this chapter largely on the summary of Mockus et al. (2002).

Open source (OS) means that the source code¹ of a software program (either in development or finished) is made freely available to all. In practice this means publishing it on a webpage. Anyone can download and view the code free of charge, and – depending on which type of open source the product is - can modify and redistribute the results with varying degrees of freedom.

2.1 Open Source Development

The OS “movement” has received enormous attention in the last several years. It is often characterized as a fundamentally new way to develop software (Dibona et al. 1999; Raymond 1999) that poses a serious challenge (Vixie 1999) to the commercial software businesses that dominate most software markets today. The challenge is not the sort posed by a new competitor that operates according to the same rules but threatens to do it faster, better and cheaper.

The basic tenets of OS development are clear enough, although the details can be difficult to pin down precisely (Perens 1999). As mentioned, OS has as its underpinning certain legal and pragmatic arrangements that ensure that the source code for an OS development will be generally available. Open source developments typically have a central person or body that selects some subset of the developed code for the “official” releases and makes them widely available for distribution.

¹ Source code is a sequence of instructions to be executed by a computer. Software developers write computer software in the form of source code (programming), and also ‘document’ that source code with brief written explanations of the purpose and design of each section of their program. To convert source code instructions into a form that can actually operate a computer, source code is translated into machine code using a software tool called a compiler. The compiling process removes program documentation and creates a ‘binary’ version of the program – a sequence of computer instructions consisting only of strings of ones and zeros. Binary code is very difficult for programmers to read and interpret. Therefore, programmers or firms that wish to prevent others from understanding and modifying their code will release only binary versions of the software. In contrast, programmers or firms that wish to enable others to understand, update or modify their software will provide them with its source code.

According to OS proponents, these basic arrangements to ensure freely available source code have led to a development process that is radically different from the usual industrial style of development. Mockus et al. (2002) lists these main differences:

- OS systems are built by potentially large numbers (i.e., hundreds or even thousands) of volunteers. It is worth noting, however, that currently a number of OSS projects are supported by companies and some participants are not volunteers.
- Work is not assigned; people undertake the work they choose to undertake.
- There is no explicit system-level design, or even detailed design (Vixie 1999).
- There is no project plan, schedule, or list of deliverables.

OS development is an extreme case of geographically distributed development, where developers work in arbitrary locations, rarely or never meet face to face, and coordinate their activity almost exclusively via the Internet. What is perhaps most surprising about the process is that it lacks many of the traditional mechanisms used to coordinate software development, such as plans, system-level design, schedules, and defined processes. These ‘coordination mechanisms’ are generally considered to be even more important for geographically distributed development than for co-located development (Herbsleb and Grinter 1999), yet here is an extreme case of distributed development that appears to eschew them all.

Open source development projects are Internet-based communities of software developers collaborating to produce software. Thousands of these projects exist today, with the number of participants in each varying from a small handful to many thousands. The number of participants in an open source project is usually much larger than the typical ‘closed’ type of software development, but most participants also generally contribute less work than would be expected in a closed project. Development approaches vary from project to project, but all are characterized by the need to cater for contributions to the product from ‘anyone’.

Degrees of involvement for participants vary, and projects are usually centered around a core of people with a relatively high degree of involvement.

Today, the number of open source projects is rapidly growing. Sourceforge.net is the world's largest open source software development website, and their repository of open source projects list roughly 90,000 registered projects, and almost a million registered users. A significant amount of software developed by commercial firms is being released under open source licenses as well.

2.1 Open Source Software

Despite the very substantial weakening of traditional ways of coordinating work, the results from open source software (OSS) development are often claimed to be equivalent, or even superior to software developed more traditionally (Mockus et al. 2002). It is claimed, for example, that defects are found and fixed very quickly because there are “many eyeballs” looking for the problems; Eric Raymond (1999) calls this “Linus's Law”. Code is written with more care and creativity, because developers are working only on things for which they have a real passion (ibid.).

It can no longer be doubted that OSS development has produced software of high quality and functionality. The Linux operating system has recently enjoyed major commercial success, and is regarded by many as a serious competitor to commercial operating systems such as Windows (Krochmal 1999). Much of the software for the infrastructure of the Internet, including the well known ‘bind’, Apache, and ‘sendmail’ programs, were also developed in this fashion. The Apache server is, according to the Netcraft survey², the most widely deployed web server in the world, and accounts for over half of the 7 million or so web sites queried in the Netcraft data collection. Other successful and widely used open source software includes Mozilla (Netscape browser core), the FreeBSD operating system, and the Perl programming language. While the existence of successful software such as these proves that OS processes can, beyond a doubt, produce high quality and widely deployed software, the exact means by which this has happened, and the prospects for repeating OSS successes, are frequently debated (see, e.g., Bollinger et al. 1999 and McConnel 1999).

² <http://www.netcraft.com/survey>.

Mockus et al. (2002) states that proponents claim OSS software to stack well up against commercially developed software both in quality and in the level of support that users receive, although the authors (ibid.) are not aware of any convincing empirical studies that bear on such claims. If OSS really does pose a major challenge to the economics and the methods of commercial development, it is vital to understand it and to evaluate it (Mockus et al. 2002).

This paper aims to contribute towards a better understanding of OS development by providing a description and discussion of underlying collaborative learning mechanisms in the open source project in the study.

3. The Knowledge Concept

“We know more than we can tell”

- Michael Polanyi 1958, p.4

Central to communication, learning and knowledge transfer is the concept of knowledge itself. In this chapter I discuss different approaches to knowledge that I find relevant to the context of my study, with special emphasis on the widely adopted concept of tacit knowledge. The concepts discussed in this chapter and the next are used to describe the knowledge sharing dynamics in the Azureus project.

3.1. Background for the Knowledge Concepts

The concept of tacit knowledge was first introduced by Michael Polanyi in his book *Personal Knowledge* (1958), and later refined in works such as *The Tacit Dimension* (1966). Although the terms were new, one had been speaking of related subjects in epistemology for quite some time. As such, tacit and explicit knowledge are generalizing terms which encompass and group these epistemological aspects.

Related epistemological terms:

Knowledge of experience (body)	Knowledge of rationality (mind)
Simultaneous knowledge (here and now)	Sequential knowledge (there and then)
<u>Analogue knowledge (practice) ~</u>	<u>Digital knowledge (theory) --</u>
Tacit knowledge (subjective)	Explicit knowledge (objective)

Despite being made known to a larger audience by being quoted in the writings of Kuhn (1962), Polanyi’s work was never widely adopted in working life or academia. However his concepts have lately gone through a renaissance due to the writings of Nonaka (1994) and Nonaka and Takeuchi (1995, 1998). These works apply Polanyi’s theory to organizational knowledge creation and knowledge management, and because of the general interest in these topics the concepts of tacit and explicit knowledge are now widely adopted. However, Stenmark (2002) points out that Nonaka uses Polanyi’s terms somewhat different than what

Polanyi did himself, and that this misrepresentation thus also has been widespread. “While Polanyi speaks of tacit knowledge as a backdrop against which all actions are understood, Nonaka uses the term to denote particular knowledge that is difficult to express” (*ibid.*).

I see Nonaka’s contributions more as an attempt to build upon, refine and apply Polanyi’s terms in a practical setting, although no doubt he could have been more explicit about this.

Since Nonaka’s use of these terms are the more widely adopted and are more tailored towards practical settings such as the object of my study, I will focus on his definitions for the remainder of this paper. Although he points out that the terms are not isolated from each other, Nonaka describes tacit and explicit knowledge dichotomously. I discuss these concepts in more detail in the following sections.

3.2 Explicit Knowledge

“Explicit knowledge can be expressed in words and numbers, and easily communicated and shared in the form of hard data, scientific formulae, codified procedures or universal principles. Thus, knowledge is viewed synonymously with a computer code, a chemical formula or a set of general rules” (Nonaka, 1998, p.215). He continues to describe explicit knowledge as the typical western view – deeply ingrained in our traditions, from Frederick Taylor to Herbert Simon, and that the tendency in both western academia and working life has been to overlook knowledge which is not explicit. Nonaka spends little effort in elaborating this claim, and it might seem like he uses this simplification – or lack of elaboration – to stereotype the western view in order to make his point about the importance of tacit knowledge. Whether this is deliberate or not it is a good example of the problems related to writing bias-free articles in social science (Hammersley 2000, Orlans 1973).

3.3 Tacit Knowledge

Eastern epistemology differs significantly from western epistemology in that explicit knowledge is viewed as only the “tip of the iceberg” (Nonaka 1998). Human knowledge is viewed as mainly non-explicit, or as Polanyi first labelled it – tacit.

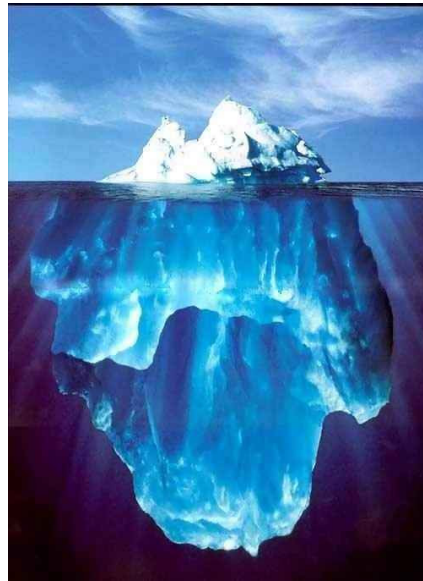


Figure 1: Eastern epistemology views explicit knowledge as the tip of the iceberg

This aspect of knowledge is described as highly personal and hard to formalize, making it difficult to communicate to others or share with others. “Subjective insights, intuitions and hunches fall into this category of knowledge. Furthermore, tacit knowledge is deeply rooted in an individual's action and experience, as well as in the ideals, values or emotions he or she embraces” (Nonaka, 1998).

Nonaka segments tacit knowledge into two dimensions - a technical and a cognitive dimension. “The first (...) encompasses the kind of informal and hard-to-pin-down skills or crafts captured in the term "know-how". A master craftsman, for example, develops a wealth of expertise "at his fingertips" after years of experience. But he is often unable to articulate the scientific or technical principles behind what he knows” (Nonaka, 1998).

That a craftsman is not able to explain any scientific or technical principles behind what he knows is relatively unproblematic. Craftsmen are usually not scientists, after all. More interesting is the fact that craftsmen are often unable to articulate details of their work even using non-technical or non-scientific language, either because they are not aware of their knowledge, because it does not surface out of context, or because it is inherently hard to explain without “doing it”.

The cognitive dimension is more interesting: "It consists of schemata, mental models, beliefs, and perceptions so ingrained that we take them for granted. The cognitive dimension of tacit

knowledge reflects our image of reality (what is) and our vision for the future (what ought to be). Though they cannot be articulated very easily, these implicit models profoundly shape how we perceive the world around us" (Nonaka, 1998).

Nonaka's cognitive dimension is similar to Polanyi's original descriptions of tacit knowledge, in that it is a cognitive framework through which we perceive the world. "Without being aware of or able to express the knowledge that is tacitly embedded in our tradition and culture, we use it as an unarticulated background against which we distinguish the particulars to which we currently attend"(Polanyi 1958/1962).

3.4 Discussion of Terms

The dichotomy between tacit and explicit knowledge is not without its commentators. In accordance with Nonaka, Cook and Brown (1999) argue, in what they claim is in agreement with Polanyi, that "explicit and tacit are two distinct forms of knowledge (i.e., neither is a variant of the other). In contrast, Tsoukas (1996), also building on Polanyi, claims that tacit and explicit knowledge are mutually constituted and should not be viewed at two separate types of knowledge. In a critique of Nonaka, Tsoukas further argues that tacit knowledge is not explicit knowledge internalised. In fact, tacit knowledge is inseparable from explicit knowledge since "tacit knowledge is the necessary component of all knowledge" (*ibid.*, p. 14). Tsoukas believes that the two are so inseparably related that to even try to separate the two is to "miss the point".

While I agree that the terms are interrelated, I find Tsoukas' point to be counterproductive. No one is claiming that these terms precisely pinpoint every particular of human knowledge; they are a contribution to try to classify elements of it. Rather than to give up trying to distinguish between these categories of human cognition, I would like to see them further refined and specified. As it stands, the terms are too general and non-specific; particularly tacit knowledge encompasses too much that is tacit in humans. As mentioned in section 3.3, Nonaka includes hunches and intuition in his definition of tacit knowledge. In my opinion, these are aspects of the human mind which go beyond what it is prudent to call "knowledge". I would rather hold that hunches and intuitions are *feelings* that *derive* from tacit knowledge. The two dimensions of tacit knowledge that Nonaka describes are also quite different aspects, although included in

the same term. In my opinion a further specification and refinement resulting in less fuzzy terms would be helpful in furthering the knowledge discourse.

On the same note I would like to see an elaboration on explicit knowledge. Stenmark (2002) concludes that all knowledge is tacit, and that explicit knowledge is synonymous with information. Building on Choo et al. (2000), who point out that receiving parties of a piece of explicit knowledge (for example some written text) can comprehend and value it in different ways due to different language, different level of maturity, or lack of required capabilities, Stenmark asks; “How, then, can it be knowledge?” (Stenmark 2002). Although he has a valid point that the distinction between tacit and explicit can be problematic, he bases it on the misconception that explicit knowledge is the same as *expressed* knowledge. Although Nonaka states that explicit knowledge *can* be expressed (see definition in section 3.2), he often uses the two interchangeably. It is likely that this has contributed to this common misconception. I would like to emphasize that explicit knowledge is not *necessarily* expressed. It is knowledge that one *potentially* can be fully and consciously aware of, and able to articulate. This example hopefully illustrates my point about the need for a further specification of the terms in question.

It should be noted that there are other concepts which describe many similar aspects of knowledge, such as ‘implicit knowledge’ or ‘knowledge/knowing’. In this paper I will use the discussed terms, and leave their further specification as a suggestion for further study.

I close this chapter with a simplistic summarization. I see tacit knowledge as being more related to ‘who we are’, and explicit knowledge closer to ‘what we know’ as filtered through ‘who we are’. While tacit knowledge certainly affects ‘what we do’, it also affects intangibles in ‘how we do’ to a greater extent than explicit knowledge. Because tacit knowledge is highly personal and contextual, and because of the intangibles related to its manifestation, it is considered inherently hard to transfer to others. In the next chapter I discuss different theories that describe how tacit knowledge can be transferred to others and utilized in a practical setting.

4. Knowledge Transfer

This chapter discusses three theories with different approaches to how knowledge can be shared between people. The first theory presented is of less relevance to my empirical study than the latter two, but introduces useful concepts and vocabulary for describing aspects of knowledge transfer, and I will use these concepts throughout the remainder of the paper.

4.1. Knowledge Conversion

Nonaka (1998) presents a model of how knowledge can be spread throughout an organization through four different modes of knowledge ‘conversion’, which he calls *socialization*, *externalization*, *combination* and *internalization*. His ‘Knowledge Spiral’ model describes what he calls ‘organizational knowledge creation’, and is anchored to a critical assumption that human knowledge is created and expanded through social interaction between tacit knowledge and explicit knowledge. Nonaka speaks of knowledge ‘creation’, and to me this implies a certain epistemological perspective. Rather than embark on the venture of discussing what knowledge really is, this paper focuses on how it can be transferred between people, and I use ‘sharing’ or ‘transfer’ rather than ‘creation’ in my discussion.

Socialization: From Tacit to Tacit

"Socialization is a process of sharing experiences and thereby creating tacit knowledge such as shared mental models and technical skills. An individual can acquire tacit knowledge directly from others without using language" (Nonaka 1998).

Primarily enabled through the activities of observation and re-experience, I believe this to be the primary mechanism for how animals and human infants learn from others. *Socialization* can be envisioned as a subset of - or a mechanism in – *socializing*. The latter is a broader, richer term involving communication, interaction and participation, and the terms should not be mistaken for one another.

Externalization: From Tacit to Explicit

Externalization is a process of articulating tacit knowledge into explicit concepts. Nonaka sees this as a quintessential knowledge-creation process in that tacit knowledge becomes explicit, taking the shapes of metaphors, analogies, concepts or models (Nonaka 1998), and since

Nonaka's model of knowledge transfer relies on tacit knowledge to become explicit before it can be shared, it is essential to his theory. His theory of organizational knowledge creation is discussed in further detail in the next section.

Combination: From Explicit to Explicit

"Combination is a process of systemizing concepts into a knowledge system. This mode of knowledge conversion involves combining different bodies of explicit knowledge. Individuals exchange and combine knowledge through such media as documents, meetings, telephone conversations or computerized communication networks. Reconfiguration of existing information through sorting, adding, combining and categorizing of explicit knowledge (as conducted in computer databases) can lead to new knowledge" (Nonaka, 1998).

Combination is a process of linking explicit knowledge, possibly causing new relationships or causality chains not previously seen. This type of learning encompasses assumptions related to 'traditional' models of learning and teaching as mentioned in section 4.3.

Internalization: From Explicit to Tacit

Nonaka describes internalization a process of embodying explicit knowledge into tacit knowledge. "For explicit knowledge to become tacit, it helps to have knowledge become verbalized or diagrammed into documents, manuals or oral stories. Documentation helps individuals internalize what they experienced, thus enriching their tacit knowledge. In addition, documents or manuals facilitate the transfer of explicit knowledge to other people, thereby helping them experience their experiences indirectly (i.e. 're-experience' them). As an example, Nonaka mentions (...) "if reading or listening to a success story makes some members of the organization feel the realism and essence of the story, the experience that took place in the past may change into a tacit mental model" (Nonaka 1998).

Nonaka relates internalization to 'learning by doing'. While this term is fitting for the example where tacit knowledge is enriched by writing down thoughts, I do not see it as related to the example where listeners or readers of a 'success story' feel and internalize the story. Describing the learning process that the readers/listeners experience as 'learning by doing' is a misrepresentation of the term in my opinion. On one hand, the internalization

process happens when a person writes something down. On the other, it is transferred through others via an artefact such as a document or an oral story.

Note that the themes in these ‘knowledge conversion modes’ have been partially discussed from various perspectives in organizational theory. For example, socialization is related to theories of group processes and organizational culture, and combination has its roots in information-processing.

4.2. Knowledge Spiral

Nonaka describes organizational knowledge creation as a continuous and dynamic interaction between tacit and explicit knowledge, as depicted in Figure 2. I include this model because it is essential to Nonaka’s contributions on learning, and although I do not find it particularly relevant for describing the learning mechanisms in my study, I have included as an example of a different approach to these dynamics.

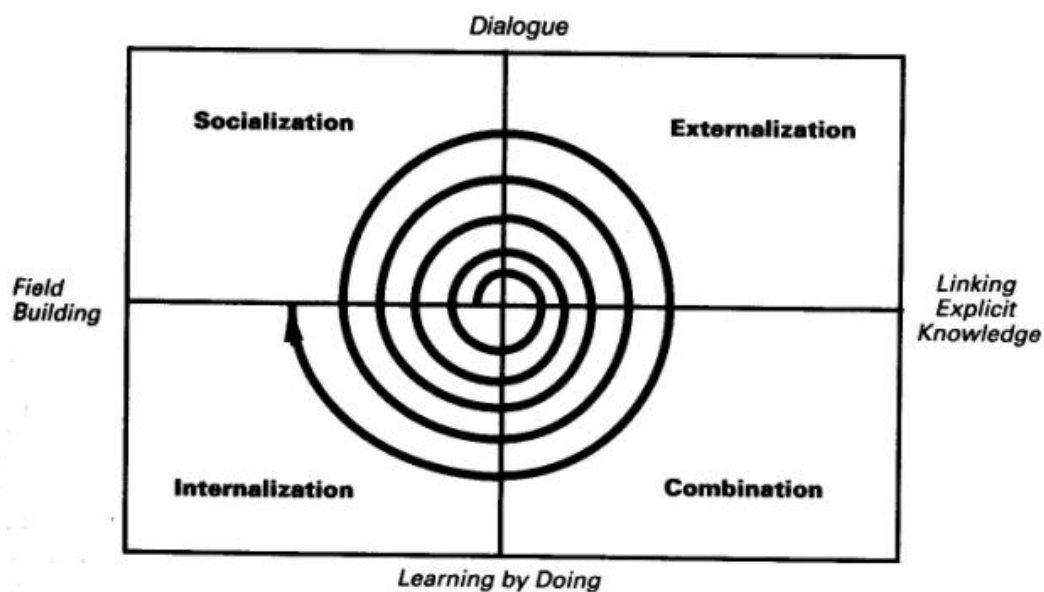


Figure 2: Nonaka’s Knowledge Spiral

According to Nonaka the process of knowledge creation happens as follows:

First, sharing of experiences and mental models (socialization) happens in a “field” of interaction. Second, externalization is triggered by meaningful dialogue. Third, combination is triggered by “networking” newly created knowledge with existing knowledge. To complete the circle, new explicit knowledge is internalized through “learning by doing”.

The model starts at an individual level, and as the spiral continues outward, on an increasing organizational level.

While I find Nonaka's knowledge conversion modes helpful to explain knowledge sharing, I find his model to be overly systematic. The Knowledge Spiral assumes a preset order in which knowledge conversion happens, which I disagree with. While I agree that the modes are interrelated, I do not see them as necessarily happening in the particular order which is depicted in the Knowledge Spiral - or in any fixed order at all.

In this model, tacit knowledge must be formalized in order to be shared throughout an organization, and thus Nonaka emphasizes the importance of externalization. He states that unless shared knowledge becomes explicit, it cannot be easily leveraged by the organization as a whole (Nonaka 1998). Nonaka emphasizes ways to enable his knowledge conversion modes, for example by encouraging use of metaphor, and how leaders can elicit tacit knowledge from members as in his example of Honda City (Nonaka 1998, p.221). His use of the word 'leverage' is a good example of his systematic and formalistic approach. As Nonaka states, the knowledge spiral is driven by an organizational intention. This encourages organizations to structure and systemize knowledge conversion and actively make the knowledge spiral happen. While Nonaka focuses on actively and systematically leveraging tacit knowledge by sharing it in formalized form, the theory of communities of practice (CoP) is less formally concerned with describing how learning takes place socially given a proper setting. I see this theory as more relevant to my empirical study, and discuss it in the following section.

4.3. Communities of Practice

"Bosses used to try to break up the gang by the water cooler. Now they support them with web sites."

(Tom Stewart in Fortune for 28 May 2001, <http://www.fortune.com>)

Many of the ways we have of talking about learning and education are based on the assumption that learning is something that individuals *do*. Furthermore, we often assume that learning "has a beginning and an end; that it is best separated from the rest of our activities; and that it is the result of teaching" (Wenger 1998, p3). The notion that learning instead is

social and comes largely from our experience of participating in daily life contradicted traditional theories of learning, where learning and working often are conceived as separate processes, and formed the basis of a significant rethinking of learning theory in the late 1980's and early 90's. In their important work *Situated Learning: Legitimate Peripheral Participation* (1991) Lave and Wenger presented a model of situated learning proposing that learning involved a process of engagement in a "community of practice". This work has later been augmented by the authors – primarily in Lave (1993) and Wenger (1998, 1999).

CoP theory has made the all too rare transition from a pure academic setting into practice and working life, and has set the scene for some significant innovations in practice within organizations and more recently within some schools (Rogoff et al. 2001). This makes it particularly interesting in relation to a study of practice such as mine.

Lave and Wenger were not alone in developing the theory of communities of practice (CoP). Basing their work on the PhD thesis of Orr (later published as Orr, 1996) Brown and Duguid (1991) illustrate in the Xerox case how knowledge is socially transferred through informal interaction. The Xerox case is about how a group of repair technicians met regularly in informal, common areas and traded stories and insights around their work (repairing different types of copying machines). The workers actually made a point out of spending more time in each others company. This 'slack' initially seemed like an excellent opportunity for productivity improvements, but management realized that these activities were actually a very important part of becoming, being and remaining a good technician. It was central to how they learned, how processes improved, how they formed bonds as a community of practice, and how they transferred knowledge and honed their expertise amongst themselves.

Brown and Duguid (ibid.) explain that through the use of storytelling, or narratives, the technicians were capable of sharing not only the explicit, but also tacit knowledge related to repairing the copying machines. According to Brown and Duguid (ibid.), stories act as repositories of accumulated wisdom, and they allow people to keep track of sequence of behaviour, as well as of the facts and their context. Through the use of narratives, the technicians were able to construct a shared understanding out of bountiful conflicting and confusing data. Returning to Nonaka, the narrators' tacit knowledge was externalized, resulting in stories which were circulated and shared through informal social interaction, and internalized by the recipients.

Wenger (1998) describes a different dynamic which also takes place in situations such as these, and which can be seen in the light of tacit knowledge transfer. In Wenger's words: "Being alive as human beings means that we are constantly engaged in the pursuit of enterprises of all kinds, from ensuring our physical survival to seeking the most lofty pleasures. As we define these enterprises and engage in their pursuit together, we interact with each other and with the world and we tune our relations with each other and with the world accordingly. In other words we learn." (ibid., p.45) In my view, and using Nonaka's terminology, this 'tuning' relates to a *socialization* of the cognitive dimension of tacit knowledge. I see this 'tuning' as a fundamental social dynamic, and as elaborated in chapter 8, as an important factor in knowledge transfer. Wenger continues to explain how this tuning eventually will define the community it takes place in: "Over time, this collective learning results in practices that reflect both the pursuit of our enterprises and the attendant social relations. These practices are thus the property of a kind of community created over time by the sustained pursuit of a shared enterprise. It makes sense, therefore to call these kinds of communities communities of practice." (ibid. p.45) In this sense, cognitive aspects of tacit knowledge are continuously shared through social interaction, and over time it will become a defining factor in a community. This type of tacit knowledge transfer stands in contrast to Nonaka's theory, which repeats that "for tacit knowledge to be communicated and shared within the organization, it has to be converted into words or numbers that anyone can understand" (Nonaka 1998, p.216).

The description of tacit knowledge transfer through narratives, and Nonaka's Knowledge Spiral (Figure 2) give the impression that knowledge conversion always happens in a set order. I believe the knowledge conversion dynamics to be more discontinued and non-systematic, and moreover strongly interrelated. I leave this discussion by viewing communities of practice as settings where these knowledge sharing mechanisms can and do happen in relation to a practice, and focus the remainder of this section on the properties of these communities.

The characteristics of communities of practice vary. Some communities of practice are quite formal in organization; others are very fluid and informal. Members of a community are informally bound by what they do together – from engaging in lunchtime discussions to solving difficult problems – and by what they have learned through their mutual engagement

in these activities (Wenger 1998). A community of practice is thus different from a community of interest or a geographical community in that it involves a shared practice.

According to Etienne Wenger (1998), a community of practice defines itself along three dimensions:

What it is about – its *joint enterprise* as understood and continually renegotiated by its members.

How it functions - mutual engagement that bind members together into a social entity.

What capability it has produced – the *shared repertoire* of communal resources (routines, sensibilities, artefacts, vocabulary, styles, etc.) that members have developed over time (see also Wenger 1999, p73-84). These representations are termed *reifications*, and as described later, also as *boundary objects*.

I see these dimensions are being interrelated. The joint enterprise fuels mutual engagement, and a shared repertoire is fundamental in defining a social entity. Likewise, a social entity is bound to produce some sort of shared repertoire, as the members adapt to – and learn from – each other.

Mark Thompson (2005) describes the ongoing political and participative dynamic present in all CoPs as a *virtuous circle*, where the more people participate, the more they learn, and the more they identify with and become prominent within a group, the more motivated they are to participate even further, and so on. I join Thompson in discussing CoPs in the narrow sense of groups of co-workers exhibiting these fundamental interactive characteristics (Yanow 2000) along Wenger's three dimensions, rather than in reference to any looser attribute (such as altruism, communitarianism) sometimes associated with the term *community*.

In table 1 Wenger's "indicators that a CoP has formed" is listed:

Table 1: Indicators that a CoP Has Formed (Wenger 1998, p125–126)

- (1) Sustained mutual relationships—harmonious or conflictual
 - (2) Shared ways of engaging in doing things together
 - (3) The rapid flow of information and propagation of innovation
 - (4) Absence of introductory preambles, as if conversations and interactions were merely the continuation of an ongoing process
 - (5) Very quick setup of a problem to be discussed
 - (6) Substantial overlap in participants' description of who belongs
 - (7) Knowing what others know, what they can do, and how they can contribute to an enterprise
 - (8) Mutually defining identities
 - (9) The ability to assess the appropriateness of actions and products
 - (10) Specific tools, representations, and other artifacts
 - (11) Local lore, shared stories, inside jokes, knowing laughter
 - (12) Jargon and shortcuts to communication as well as the ease of producing new ones
 - (13) Certain styles recognized as displaying membership
 - (14) Shared discourse reflecting a certain perspective on the world
-

Thus if the virtuous circle that Thompson describes is in operation, it should be possible to identify some or all of these indicators in a community, or expect them to manifest themselves over time. Thompson points out that these indicators comprise both structural forms (Items 10-14), as well as epistemic behaviour (Items 1-9) (Thompson 2005). By 'structural forms' he is referring to boundary objects as described above. His point is valid, although I do not find it helpful to include "shared perspective" as a boundary object. I already find the 'boundary object' term very broad and general, as it seems to comprise anything that can be conceived of as an object and circulated in a community. While this can be pragmatic in some cases, I believe that it can be misleading to group elements such as "documents" and "concepts" within the same term, and I believe including "shared perspective" in this context is a turn for the worse. I see a shared perspective more as the result of the 'tuning' that Wenger speaks of – or in other words a significant transfer of elements the cognitive dimension of tacit knowledge.

Wenger describes how communities of practice move through various stages of development characterized by different levels of interaction among the members as well as different kinds of activities. These stages and activities are presented in figure 3.

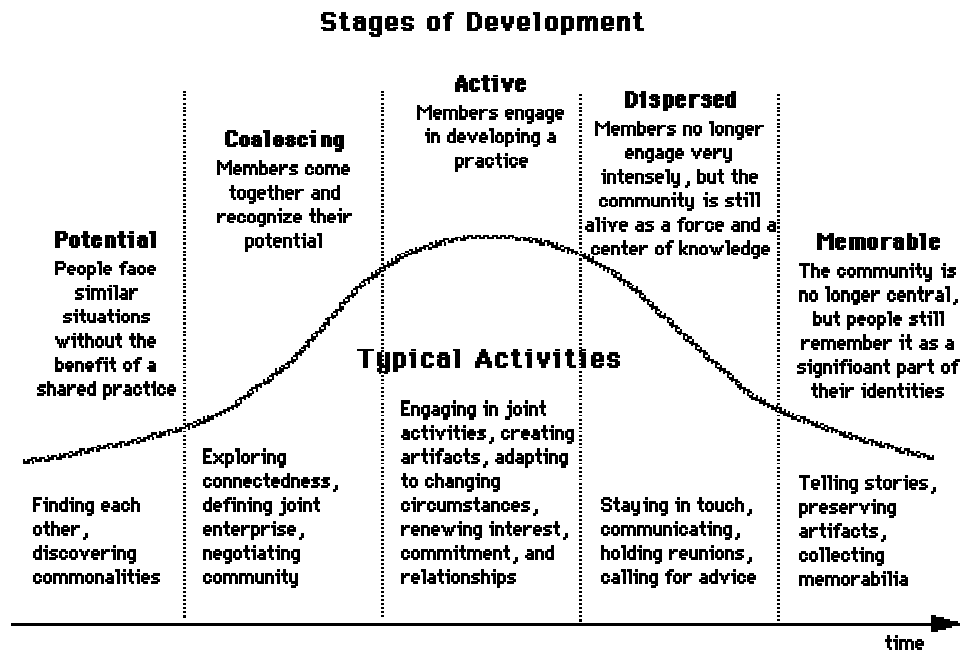


Figure 3: Stages of Development in Communities of Practice

As I will describe later, the Azureus community displays characteristics of being an ‘active’ community of practice. As such it would be expected to find at least some of the indicators listed in table 1. I return to how these indicators manifest themselves in the Azureus community in chapter 8.

The point that learning derives from social interaction is an important one, but in focusing on the social aspects, both Nonaka’s model and CoP somewhat neglect the role that artefacts can play in transferring knowledge. Although Nonaka mentions writing as an act of converting tacit knowledge into articulable knowledge (Nonaka 1998, p.221), artefacts – the written document in this case – are primarily described as conveyors of explicit knowledge. Reifications in CoP may also function as *boundary objects*, through which different communities can relate to each other. These are described as “artefacts, documents, terms, concepts, and other forms...around which communities of practice can organize their interactions” (Wenger 1998, p.105). Artefacts in CoP are usually described either as boundary objects or the central joint enterprise, and these are primarily treated as organizational elements, around which social interaction takes place. The notion that both tacit and explicit knowledge can lie embedded in the artefacts themselves, and that this embedded knowledge can be transferred between people, is neglected in Nonaka’s Knowledge Spiral as well as in

CoP theory. In the following section I discuss an approach to knowledge transfer which emphasizes the role of the artefact to a greater extent.

4.4 Knowledge Accumulation and Transfer

In his novel work *“The ad hoc collective work of building Gothic cathedrals with templates, string and geometry”* David Turnbull (1993) describes how medieval cathedrals were built in a discontinuous process by groups of masons, and in the example of the Chartres cathedral - spanning between 25 and 30 years including several successions of builders. He offers a contribution to understand how the masons could build these tall and complex buildings without knowledge of structural mechanics.

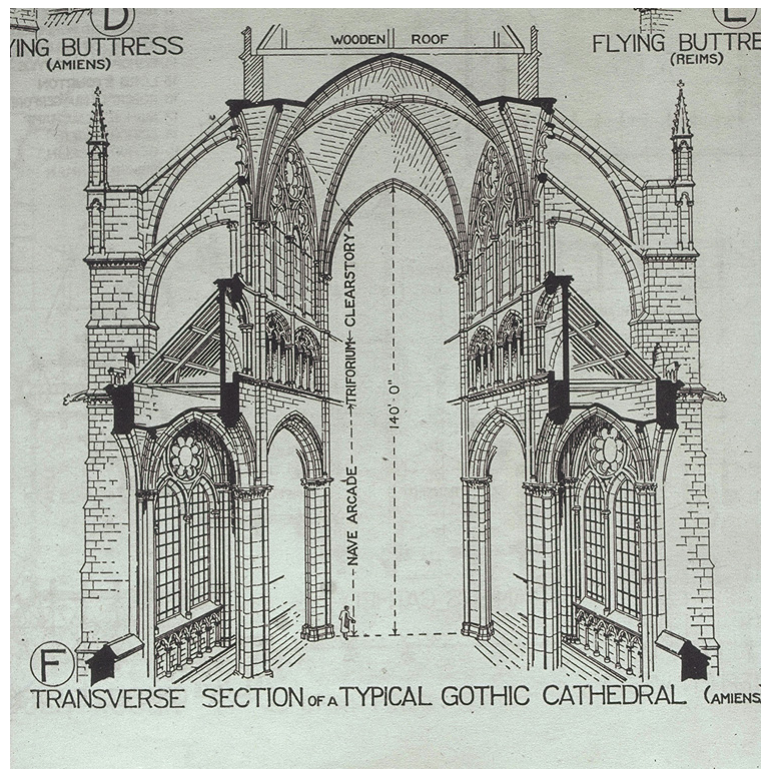


Figure 4: A gothic cathedral with arched (flying) buttresses

During the 13th century 50 cathedrals were raised throughout Europe. Turnbull envisions the cathedral building site as an "experimental laboratory" in which the key elements were the template, geometry, and skill (ibid. p.322). The argument is that the collective work of cathedral builders was not one of human ingenuity alone, but also manifest in tools. The most important of these tools was the template, which is described as a pattern or mold, usually

outlined on a thin piece of wood. The template was used by stonemasons to cut a stone to a particular shape. Turnbull views the templates as accumulations of every design decision that had to be passed on. Because a template is easy to replicate, it could circulate among builders at a site, and among building sites across Europe. In this way, knowledge of gothic cathedral building, as manifested in the template, could circulate and spread.

The way the artefact accumulates knowledge is a primary explanatory factor in Turnbull's work. In seeing artefacts such as texts, computer code or the template as the embodiment of formal abstractions, they only allow the accumulation and transfer of explicit knowledge. Berntsen et al. (2004) see the template as the "embodiment of part of the gothic architects acquired implicit knowledge/knowing". Although I find embodiment a strong word, I believe that the process of designing these templates embedded parts of the architects' tacit knowledge within the template. In designing them, there were several design decisions that needed to be made. These design decisions were made based on the designers' explicit knowledge, as well as both dimensions of their tacit knowledge. This knowledge was reflected by the result of their decisions which was manifested physically in how the template turned out. These manifest decisions within the template could only be truly recognized by others who had appropriate background and knowledge about architecture. A template would mean little to anyone unversed in architecture or stone masonry from that period of time, on the same level as computer code means little to people unversed in programming. Alavi and Leidner (2001) point out that only individuals who have a requisite level of shared background can truly exchange knowledge, and this applies for knowledge transfer through artefacts as well in my opinion. I hold that given the appropriate knowledge, including intangibles of tacit knowledge such as shared experiences or perspectives, the tacit knowledge which influenced the design decisions in the template can be internalized by - and thus transferred to - a receiver, and furthermore that the level of this prerequisite knowledge and background is directly related to the degree that tacit knowledge can be transferred in such a manner. Although I believe this transfer to be possible completely without social interaction, discourse regarding the artefact including both designer(s) and receiver will facilitate it. Moreover, having the artefact at hand in these discussions will be an enriching factor surrounding social interaction in that aspects of the artefact can trigger discourse around its design, which again can result in further embedding of tacit knowledge in the artefact, which through this discourse will include aspects of a *shared* tacit knowledge of the collaborators.

As such I see the knowledge transfer mechanisms of CoP and Turnbull as being complementary as well as mutually enabling and enriching.

To summarize; the transfer of cognitive dimensions of tacit knowledge in a CoP, as described as ‘tuning’ by Wenger, results in shared perceptions, experiences and background, which I see in turn as enabling factors in the transfer of both the ‘skill’ and ‘cognitive’ dimensions of tacit knowledge through artefacts. And as explained above, this latter transfer is facilitated by meaningful surrounding discourse, which in turn enables further tacit knowledge to be embedded in the artefact. I offer this cyclic interaction as an important knowledge sharing dynamic.

Nonaka states that sharing tacit knowledge between individuals through communication is an analogue process, and that it requires a kind of “simultaneous processing” of the complexities of issues shared by the individuals (Nonaka 1998, p.219). However unlike human knowledge the embedded knowledge in the artefact is made *durable*, which alleviates the problems related to the discontinuous process in which they were made. The durable manner in which the artefact accumulates knowledge is a primary factor in Turnbull’s explanation of how knowledge of building gothic cathedrals was spread throughout Europe, and how the building of these could be successfully conducted as a discontinued as they were.

Turnbull’s approach is specific on the role technology plays in transferring knowledge, and stands in contrast to the social focus of Nonaka and CoP. In chapter 8, I will attempt to show that through CoP dynamics and the use of discourse around artefacts, the Azureus project achieves a knowledge sharing dynamic similar to the cyclic interaction described above.

Collaboration in the Azureus project is conducted strictly online, and a slight detour discussing how the use of communication technology affects relevant aspects of collaboration is needed. This is the topic for the next chapter, which also includes an elaboration on my motivation for the adopted angle of this paper.

5. Collaborative Technology and Theory

“The computer is a moron.”

- Peter Drucker

Information technology can ‘support’ our capabilities, we often hear – and interestingly, such claims can be heard in both the standard rhetoric from IT optimists and in CSCW³ - Computer Supported Co-operative Work (Berg 1999). Such imagery, first of all, overlooks the fundamental ways ‘Man’s capabilities’ are *transformed* through their interlockings with IT (Haraway, 1991; Hayles, 1994; Edwards, 1996). ‘Communication’ and ‘memory’ are not given entities that are supported – or hampered – through IT. The very meaning of these capabilities change, take on new dimensions – and an investigation of these changes is an important task (Berg 1999). A thorough investigation of how these capabilities are transformed is an extensive task, and is beyond the scope of this paper. The message I want to get across in this chapter is rather the notion that although Computer Mediated Communication (CMC) is inherently different from physical communication, the primary CMC technology used in the Azureus project is “enough” to support social interaction and community features important for a CoP. I start this chapter with some background, leading to my motivation for the adopted angle in this paper.

5.1 Literature Review

End-user empowerment, changes in organisational structure and developments in technology appear to have heralded the evolution of a new generation of computer systems. These are computer systems that serve as mediators and “supporters” of communication and information for people separated by geographical distance. This new generation of technology is known as collaborative technology. Academic branches dealing with these issues include the study of Computer Supported Co-operative Work (CSCW), and Computer Mediated Communication (CMC).

³ It is, after all, the S in CSCW. See e.g. Robinson (1992); Schmidt and Bannon (1992); Button and Harper (1996).

Collaborative technologies are unique in their abilities to generate “an interface to a shared environment in which users are linked in multiple ways such that they will perceive themselves to be communicating as if they were in the same place” (Gay and Lentini 1995). This sets collaborative technologies apart from more traditional stand-alone technologies.

Conkar (1999) has performed an extensive literature review as part of her Degree of Doctor of Philosophy, and I largely base the following two chapters on her analysis.

5.1.1. CMC Literature

CMC has been described as the process of “human communication via computers” which involves people “situated in particular contexts” for a “variety of purposes” (December 1997). December also provides a more specific definition with more relevance to this paper; “CMC is any computer mediated Information System (incorporating collaborative software and hardware) that enables geographically distanced individuals or groups to communicate, share, and learn together” (December 1997).

Studies in CMC have examined the concept of social community and the technical means to support this. Work includes an analysis of the process of group communication (Powell 1996); the investigation of group communication and transactions in distributed systems (Schiper et al. 1996), the use of communities for the advancement of knowledge (Scardamalia et al. 1996), and the support of informal communications (Whittaker et al 1994).

Conkar states that the main problem with the analyzed CMC literature is that studies appear to remain largely technical in nature, looking at means for improving the computer interface between the users and the technology while largely ignoring the issues of inter-personal interaction (Conkar 1999). Exceptions to this finding include the work of Mynatt et al. (1997) on creating successful network communities which provides a set of principles to apply practically in design, and the work of Scaife et al. (1997) on informant design in interactive learning environments.

5.1.2. CSCW Literature

As Conkar explains, this field includes studies with focus on communication issues related to the use of media and its application by participants (Miles et al 95; Wright et al 1992). Some

research has gone further in looking at design issues related to participant collaboration, with notable examples including Dourish (Dourish 1991; Dourish 1992; Dourish 1993a; Dourish 1993b; Dourish et al 1994). However, Conkar states that these papers remain largely technical, placing emphasis on network set-ups and programming, with only passing reference to the social aspects and consequences of CSCW.

Some research in this field has taken a more practical approach to assessing collaborative work and its impact on participants however, and notable examples are listed as the 'Teamrooms' project (Roseman 1996) using the World Wide Web, the 'Mushroom' project (Kindberg 1997) examining the concept of technically created social boundaries, and the 'Zephyr Help instance' at MIT (Ackerman and Palen 1996) which looks at communities of practice and the adaptation of a technical interface for social means (including group role-play). Other examples include 'LIZA' a GroupWare tool kit (Gibbs 1989), 'AROMA' for supporting presence and mutual awareness (Pederson et al. 1997), 'Groupsketch' for the geographically distributed (Greenberg et al 1991), and 'TOTEM' a group communications system (Moser et al 1996).

Conkar also includes studies that focus on more cognitive issues (Petre et al. 1996; Norman et al. 1994; Hutchins 1990; Hutchins 1995; Kyng 1995; Fischer 1995).

However according to Conkar, the bulk of CSCW literature reviewed for this literature review suggests that CSCW may compartmentalise issues relating to human activities in design rather than examining them in context.

My study is related to both research fields, and drawing upon Conkar's aforementioned criticism of current literature I focused on inter-personal interaction more than technical interface, while examining and analyzing human activities in their natural context rather than focusing on isolated or compartmentalized issues.

Conkar wrote her literature review in 1999, and thus it is becoming somewhat outdated. However as mentioned in the introduction, despite the recent increasing research effort into open-source communities of practice (Tuomi 2001; Lanzara and Morner 2003; Gray 2004; Sharratt and Usoro, 2003), existing literature nevertheless leaves us uninformed about how knowledge sharing and creation processes develop at the interface of technology and

communal structures that effectively exploit the advantages of Internet technology and at the same time are able to overcome the problem of tacit knowledge transfer (Hemetsberger et al. 2004).

In chapter 8 I offer a contribution to understand how these processes take place in the Azureus open source project. As explained in chapters 6 and 7, and elaborated in chapter 8, the most important technology used in these processes is Internet Relay Chat (IRC). The next section discusses this technology, but in fear of an over-emphasis on technicalities of computer interface, or a severe detour of cognitive aspects, it is no way intended to be an extensive treatment. Rather I seek to briefly describe the technology and its use in relevance to the analysis and discussion conducted in chapter 8.

5.2. IRC

IRC stands for "Internet Relay Chat". It was originally written by Jarkko Oikarinen in 1988. Since starting in Finland, it has been used in over 60 countries around the world. IRC is a multi-user chat system, where people (represented by their login nicknames, or "nicks") convene on "channels" (rooms, virtual places, usually with a topic of conversation) to talk in groups, or privately. Channel names are prefaced by a "#" sign. IRC gained international fame during the 1991 Persian Gulf War, where updates from around the world came across the wire, and most IRC users who were online at the time gathered on a single channel to hear these reports. IRC had similar uses during the coup against Boris Yeltsin in September 1993, where IRC users from Moscow were giving live reports about the unstable situation. To join an IRC discussion, you need an IRC client and Internet access. The IRC client is a program that sends and receives messages to and from an IRC server. The IRC server, in turn, is responsible for making sure that all messages are broadcast to everyone participating in a discussion. All servers are interconnected and pass messages from user to user over the IRC network. Some of the more popular chat clients are mIRC, Pirc, and Virc for Windows, and Homer or IRcle for Mac. What program you use doesn't really matter; all of them connect to the same chat networks. When logged into a chat session, you converse by typing messages that are instantly sent to other chat participants. There is no restriction to the number of people that can participate in a given discussion, or the number of channels that can be formed on IRC.

It is possible to use ‘bots’⁴ in IRC channels in order to perform automated tasks like enforcing channel rules. Further channel control can be obtained by giving users ‘operator’ or ‘op’ status, which gives them access to more control of the channel like being able to ‘kick’ or ‘ban’ other users from the channel, or change the channel topic.

5.2.1 IRC Use

In this section I focus on specific aspects of the use of IRC communication as relevant to this paper. The relevance and use of these aspects in Azureus collaboration is discussed in more detail in chapter 8.

An aspect of relevance is the way that IRC can be used for both synchronous and asynchronous communication. Although text typed in the channels scrolls upwards as more text is input, it does not disappear, allowing latecomers in a discussion to view the transpired interaction as a whole. In physical synchronous communication the parties are dependent on understanding the discussions as they happen, on taking notes, or on a good memory.

Although in no way substituting the richness and intangibles of physical communication, the asynchronous aspect of IRC allows for more time to think about response and input, and allows for a revisiting and re-experience of the transpired discussion.

Another distinct aspect of IRC communication is the ease of asking questions or directing comments with a ‘one-to-many’ relation, as text typed in a channel is visible by everyone in it. In online communities with a fluctuating member base and varying degrees of participation like open source projects, it can be difficult to keep track of ‘who-knows-what’, and thus who to turn to with questions. The ease of which IRC supports open discussion alleviates the problem. A complementary factor is the relative anonymity granted by ‘nicks’. This anonymity can reduce the fear of asking ‘stupid questions’, but can also cause people to give less consideration to how they act. I see the implications of this anonymity as important to the mannerism and quality of user feedback, but I do not investigate it further in this paper.

The last discussed aspect is the ability to use the ‘cut/paste’ functions in operating systems to introduce previously written text to the channel. Using this function to provide links or text

⁴ Short for “robot” – a small program that runs automatically. Bots have nicks just as normal users do.

snippets, coupled with the ‘one-to-many’ aspect, is an easy and effective way of providing participants in a discussion with a shared access to the topic of conversation.

As detailed in chapter 8, IRC seems to provide enough aspects of human interaction for the members in the Azureus project to display community characteristics. In section 5.1.2 I explained my adopted angle for my study, and in the next two chapters describe how I went about this task.

PART 2: RESEARCH SETTING AND METHODOLOGY

6. Azureus

In this section I present the object of my empirical study and the adopted research methodology. I start with a description of the Azureus program followed by the Azureus project. I keep these descriptions short, primarily emphasizing aspects that are relevant for the topic of this paper.

6.1 The Azureus Program

Azureus is an advanced multiplatform BitTorrent client written in Java that includes a tracker⁵ and a torrent⁶ maker, thus providing all the components needed to run a BitTorrent solution. BitTorrent is a file distribution protocol that reduces the load on central servers by using peer-to-peer technology. Instead of downloading from a single overburdened server, a client can download from hundreds (or thousands) of other peer clients simultaneously, allowing for efficient and high-speed content delivery. Azureus differs from other current BitTorrent clients in its from-the-ground-up protocol implementation, a multi-platform graphical interface, and its extensive plug-in support.

Azureus has a staggering 250,000 lines of code, and enjoys roughly 75,000 downloads a day on average. The core developers believe that about 200,000 users are using Azureus on a daily basis. They also state that more than a million users have tried the product at least once. Azureus runs under Windows, Linux and OS X among others, and is licensed under the GPL.

6.1 The Azureus Project

The Azureus project officially started on the first of June, 2003. Like most open source projects, the number of developers has been fluctuating. However its member base seemed to have settled down before I started my study. During the period of my observations, the project had the same four core developers, as well as a relatively stable base of more peripheral contributors, numbering around 20 or so. In addition, users giving feedback in some form of another numbered in the hundreds or thousands.

⁵ A tracker is a program run off of a server. The tracker keeps track of all the users that are downloading a particular file. Clients connect to the tracker, and are given a list of other users to download from.

⁶ A .torrent file (aka metafile) is a small file (with the extension .torrent) containing the information necessary for your BitTorrent client to download a larger file.

Coordination and communication

As explained in the next chapter, the primary means of communication and collaboration in the project are the Azureus IRC channels. Details of this collaboration are elaborated in chapter 8. The primary channels are #azureus-users, #azureus, and #azureus-dev, in addition to some language specific user help channels. The first is a channel primarily used for user feedback and service. It is a big (roughly 40-50 people on average), noisy channel where users come to interact with project members, usually to get help with some problem. It can be pretty chaotic at times, and is strictly moderated by “operators” and “bots”. The latter is a very small channel where the core developers collaborate. Chat in this channel is almost exclusively on-topic regarding Azureus development. #azureus is something in between; here developers – core and others – interact with each other and with knowledgeable users, numbering around 25 on average. As elaborated in chapter 8, interaction here is characterized by a mix of work-related issues and general social interaction. Users asking elementary questions in this channel are redirected to #azureus-users.

In addition to the IRC channels, a CVS⁷ and a change/commit log were used to aid coordination and awareness of development. A webpage⁸ and a wiki⁹ comprised the “front” of the project. There was also a forum which was primarily used by users to give feedback about the program. Posts made by developers and community members on this forum were primarily directed at answering questions from users. Discourse among community members was almost exclusively conducted in the IRC channels described above.

I kept this chapter short and descriptive with the intention of providing some context of the topic of the paper. Further details of IRC can be found in chapter 5, while chapter 8 elaborates on the collaboration and community of Azureus. The next chapter describes my research and the motivations for my actions regarding this study.

⁷ Short for *Concurrent Versions System*, a program that allows developers to keep track of different development versions of source code.

⁸ <http://azureus.sourceforge.net/>

⁹ A collaborative website comprising the perpetual collective work of many authors.

7. Research Methodology

The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.

Sir William Bragg (1862-1942)

In this chapter I describe the form and nature of my research, the manner in which it was conducted, and the objective behind it.

7.1. Why Azureus?

This paper builds on a preliminary study of the Azureus project which I conducted in the fall of 2004. At that time my objective was to gain a qualitative understanding of how open source projects “worked”, without really knowing what I was looking for beyond “some collaborative mechanisms”. As such I had primarily practical reasons for choosing Azureus.

The *size* of the project was perhaps the most important criteria; in order to maintain an overview of the project it could not be too big, and since I wanted to study collaborative mechanisms within a community it could not be too small. I saw the Azureus project as being just the right size for these ends.

I also wanted an *active* project, primarily because I did not want the project to die on me just as I was about to get a solid understanding of its inner workings. While this might have been interesting, I wanted to ensure that I would be able to continue studying the project while writing this paper, and not just for the duration of the preliminary report. Azureus was listed as the most active project on Sourceforge at the time (and still is) so it seemed fitting. That the project was also in the “active” stage of development as described by Wenger in figure 3, and thus visibly would display indicators of a CoP, was more a stroke of good luck (considering the topic of this paper) than a conscious factor in my choice of study.

I also wanted to study a *successful* project because I thought factors for success were interesting than factors for failure. Since the Azureus project also topped the list of numbers of downloads (and still does), it seemed to at least to be doing something right.

Visibility was a very important criterion for me. If I were to observe collaboration I needed to ensure that I had access to the important channels of communication. Although the core developers sometimes used phone or emails, I was assured by one of them that the IRC channels were the primary mode of communication, and that little relevant interaction took place in private channels.

Lastly I wanted a project that displayed a high degree of *collaboration*. I was looking for a project where people collaborated on their work rather than enforcing work assignments. I was looking for something along the lines of what Beck (2000) describes as “code ownership”. This is a concept where code may be assigned to someone, but other developers still “own” the code as well, are responsible for it, and can change and contribute to it. One of the core developers made a comment relevant to this in an interview with sourceforge in association with Azureus being voted “project of the month” in September 2004¹⁰.

“Bugs are assigned to the most appropriate person, but one developer may still correct someone's else bug” (core developer).

Judging from this comment and the quite visible collaboration in the Azureus IRC channels, it seemed my last criterion was met.

As I started on this paper, the choice whether to continue my studies on the Azureus project or change to another was relatively easy. I felt I already had a decent grasp of the collaborative dynamics in the Azureus project, and since I now had decided to focus on knowledge transfer, I decided I needed depth before breadth in my study, and so I continued my observations of Azureus. As such, my empirical studies of the Azureus project spanned for about 10 months total.

7.2. Classifying my Research

This section offers a description of my research approach, as well as some background theory. For this section I primarily use Colin Robson’s “*Real World Research*” (2002) as framework and vocabulary.

¹⁰ <http://sourceforge.net/potm/potm-2004-09.php>

7.2.1. Qualitative and Quantitative Approaches

Within social science it is common to separate between ‘qualitative’ and ‘quantitative’ research, having their roots in relativism and positivism, respectively. (Robson, 2002, p.24)

The nature of a research project in its entirety can rarely be labelled as wholly quantitative or wholly qualitative, and a study that is primarily based on qualitative data will often look at quantitative data as well. The difference lies in how you regard the data; it is quite possible to have a non-positivist approach to quantitative data, and vice versa. The ‘qualitative/quantitative’ label speaks of research methodology, while the ‘(non)positivist’ label speaks of the underlying philosophy. Labelling a research project as either qualitative or quantitative neglects this distinction.

Robson (2002) uses a different set of labels, and separates between two extremes in research design – *flexible* and *fixed* strategies.

*“A **fixed design strategy** calls for a tight pre-specification before you reach the main data collection stage. If you can’t pre-specify the design, don’t use the fixed approach. Data are almost always in the form of numbers; hence this type is commonly referred to as a **quantitative strategy**.*

*A **flexible design** evolves during data collection. Data are typically non-numerical (usually in the form of words); hence this type is often referred to as a **qualitative strategy**.”*

(Robson, 2002, p.87)

The main goal of my research was understanding and insight. This entails an interpretation of a dynamic setting, and thus a flexible design was the natural choice for me. I also didn’t know what I was looking for at first, and I needed to be able to adapt my approach as I uncovered data.

7.2.2. Flexible Research Strategies

Robson presents three traditional flexible design research strategies; case studies, ethnographic studies and grounded theory studies. A short description of these approaches is given here.

Case study

Cases studies entail development of detailed, intensive knowledge about a single ‘case’, or of a small number of related ‘cases’ (Robson 2002). The details of the design typically ‘emerge’ during data collection and analysis.

Typical features:

- selection of a single case (or a small number of related cases) of a situation, individual or group of interest or concern.
- study of the case in its context.
- collection of information via a range of data collection techniques including observation, interview and documentary analysis.

Ethnographic study

Ethnographic studies seek to capture, interpret and explain how a group, organization or community live, experience and make sense of their lives and their world (Robson 2002).

They typically try to answer questions about specific groups of people, or about specific aspects of the life of a particular group (Bentz and Shapiro, 1998, p.117).

Typical features:

- selection of a group, organization or community of interest or concern.
- Immersion of the researcher in that setting.
- User of participant observation.

Grounded theory study

The central aim of a grounded theory study is to generate theory from data collected during the study. They are seen as particularly useful in new, applied areas where there is a lack of theory and concepts to describe and explain what is going on. Data collection, analysis, theory development and testing are interspersed throughout the study.

Typical features:

- applicable to a wide variety of phenomena.

- commonly interview-based.
- a systematic but flexible research strategy which provides detailed prescriptions for data analysis and theory generation.

In the light of the descriptions given above, my research comes off as a mix between a case study and an ethnographic study. Robson emphasizes that these are ‘ideal types’, and that combined, or hybrid, types are often used. He also makes a point that the research design should be tailored to fit the research purpose, and not the other way around (Robson, 2002, p.80). Because of this I did not attempt to adapt my research according to one of the abovementioned approaches, but rather focused on doing what I found adequate and helpful in order to further my goal of a qualitative (non-positivist) understanding of collaborative mechanisms, and eventually, of knowledge transfer. My study is perhaps best described as a case study during the fall of 2004, and an ethnographic study in the fall of 2005. I will not dwell any further on classifying my study, but rather move on to describing my actions and their motivation.

7.3. My Research Approach

My research was primarily influenced by guidelines given by Robson (2002), and by research principles given by Klein and Myers (1999). Although partially overlapping, their advice helped me conduct my study in a manner which seemed appropriate. I describe their application in the following sections.

7.3.1. Being ‘Scientific’

In conducting my research I started off with adopting some general advice. Robson (2002) offers some guidelines to adopt what he calls a ‘scientific attitude’. By this he means that the research is carried out *systematically, sceptically and ethically*.

‘Systematically’ entailed me putting serious thought to what I was doing and how and why I was doing it. Related to my observations, it required me to be explicit about their nature, the circumstances and context in which they were made and the role I took in making them. In considering the role I played in my observations I bore in mind Klein and Myers’ ‘*principle of interaction between researcher and subjects*’ as described shortly.

‘Sceptically’ required me to subject my ideas and perceptions to possible disconfirmation, and also subjecting my observations and conclusions to scrutiny. In making observations I would check and double check my logs in order to make sure that I perceived them in the same manner, and before drawing any conclusions, subject my perceptions to scrutiny in an incremental manner.

‘Ethically’ entailed me to follow a code of conduct for the research which ensured that the interests and concerns of those taking part in, or possibly being affected by, the research were safeguarded. Specifically this meant that I should remain conscious that I was observing interaction which for many of the people involved, were part of their daily life. I have been mindful to not include specifics which might reflect badly on any of the subjects, have made anonymous any quotations, and will not reveal specifics of my observation logs.

Klein and Myers (1999) suggest a set of seven principles for the conduct and evaluation of interpretive field research in information systems (Klein and Myers, 1999). The authors encourage researchers to judge whether, how and which of the principles to use, even though they are, to some extent, interdependent. I will not give a full description of them here, as many are already included in the adoption of Robson’s ‘scientific attitude’; the principles of ‘multiple interpretations’ and ‘dialogical reasoning’ are related to being ‘sceptical’, and the principles of ‘abstraction and generalization’ and ‘contextualization’ are related to being ‘systematic’. However I will describe two principles in more detail in the following sections, as well as their adaptation in my study.

7.3.2. The fundamental principle of the Hermeneutic Circle

This principle suggests that all human understanding is achieved by iterating between considering the interdependent meaning of parts and the whole that they form, and that this interaction between parts and the whole enriches the understanding of both. This principle is described as being fundamental to all the other principles (Klein and Myers 1999)

In accordance with this principle I iterated my attention between analysing details of observed communication in the Azureus IRC channels as parts, and my general understanding of interaction and knowledge transfer in a wider context in order to interpret the observed interaction as a whole

This approach seemed to be working, as through this iteration observed occurrences took new meaning as I returned to them after analysing the whole picture, and detailed observations yielded new considerations and input to my wider understanding of collaboration and knowledge transfer. As an example, one observation I made was a long argument between an angry user and two community members. The incident eventually resulted in the user being banned from the channel, upon which he promptly created a new channel (this is very easy in IRC) with a derogatory name indicating his feelings about the two members, and invited other people to enter and celebrate their dislike towards them. No one entered except the two community members and myself, and the quarrel eventually resulted in the user apologizing, giving “op” status to one of the members, upon which the member banned the user from his own channel. Besides getting a chuckle, I originally did not perceive this as anything more significant than an amusing incident, and I did not see the actions of the community members as anything but bordering on harassment. However as my knowledge of communities of practice (CoP) as applied to the Azureus community increased, I thought of this incident and returned to it in my logs. The episode then took on new meaning, and I saw the members’ actions as related to asserting identity within the community, their ‘sticking up’ for each other as evidence of a sense of belonging, their seemingly coordinated actions as suggestion of a shared perspective, and the whole incident as a shared experience as relevant to CoP. This new perception of a part in turn enriched my understanding of interaction in the Azureus community as a whole, as this episode now could be seen as an indicator that a CoP was in place, which again enriched my understanding of observations, and so on.

7.3.3. The principle of interaction between the researcher and the subject

This principle emphasizes the importance of critical reflection how the research materials (or ‘data’) were socially constructed through the interaction between the researchers and participants. There are two primary points relating to this principle; one is how the researcher affects the object of his study by introducing disturbing elements or by the researchers

presence in his observed setting, and the other is to which extent the researcher ‘creates’ meaning to observed phenomena. I discuss these two points in the order mentioned above.

I wanted to observe how collaboration took place in this project, and I made a conscious choice not to involve myself in the project because in doing so I would inevitably affect my observations. For example, if I were to contribute to the community by answering questions posed by users, it would affect the manner in which others answered them, thus ruining my chance to observe the occurrence as though it would have transpired without my interference. This principle was a deciding factor in my choice to not use interviews but instead rely on observation as my primary method of collecting data. Since the Azureus community had relatively few members I would have to interview a relatively large portion of the community in order to get data which could be subject to generalization. Interviewing 10 people – a decent, yet small number for getting generalizable data, would still almost constitute half of the regular member base. 5 would still be a considerable portion, and as few as 3 could not be assumed to give a representative impression of the community. I saw conducting interviews – even very informal ones, as possibly resulting in a higher level of self-consciousness within the community, which could affect the manner in which they interacted. Furthermore, Thompson (2005) described self-consciousness around community behaviour as an important attribute in CoPs, and I wanted to disturb this level as little as possible. Another factor for my decision not to use interviews was the fact that interaction and collaboration in the community was handily available, as most of it was conducted in IRC channels open to everyone. The exception to this was the #azureus-dev channel where the core developers interacted. The existence of this channel was not public knowledge, and I only became aware of it from a passing comment made in #azureus. As I entered this channel I was promptly but politely asked to leave by one of the core developers. In order to be allowed to stay in this channel I had to reveal my intentions to this one developer. He agreed to keep my intentions a secret, and the one time one of the other developers commented on my presence, he cannily brushed the matter aside with a joking comment. No more comments were made about me after that, and it seemed as I could conduct my observations with little disturbance of my subjects. I made sure to never say anything at all in the channels, as well as staying in the channels constantly in order to remain as anonymous as possible. It seemed to me that people either grew accustomed to me, or didn’t notice me at all, and that I could continue to be a ‘fly on the wall’ to some extent.

Since I had already “blown my cover”, so to speak, with this one developer, I used the opportunity to ask him supplementary questions about the Azureus development from time to time. As such, I conducted a discontinued informal interview of sorts with him that spanned several weeks. Since I only had one interviewee I was conscious to minimize room for him to give me non-representative information and opinions, which could give me impressions not shared by others in the community. Instead my questions were of a simple and practical nature, and it was this developer that ensured me that the Azureus IRC channels were the primary communication channels for the project.

The second point of the ‘principle of interaction among researcher and subject’ made me mindful about the meaning of my observations. In relying on observation as the primary mechanism for collecting data, I was basing my data on my own interpretations to a significant extent. In doing so there was a real danger that I, by having the theories of knowledge transfer as discussed in chapter 4 in mind, was constructing meaning to my observations and ‘fitting them into’ theory instead of objectively relating them to theory. By logging all observed communication I have been able to revisit occurrences of interest and interpret them over time, hopefully gaining a correct interpretation of the observed occurrence. In being mindful of the danger of constructing ‘things’ that aren’t there, I would use these logs to scrutinize and revisit my perceptions of occurrences. To further alleviate the problem I have emphasized aspects that manifested themselves on several occurrences rather than base my opinions on one-time occurrences.

As explained in this chapter, my empirical study attempted to identify and understand the collaborative and communicative dynamics relating to knowledge transfer in the Azureus community, primarily through observation. After discovering that the Azureus IRC channels were the main venue for interaction, I focused my attention on monitoring them. As I will show in the next chapter, this observation enabled the identification of common mannerisms and styles of informal behaviour that were consistent with items in Wengers’ “indicators that a community of practice has formed” (1998, p125) as set out in table 1, and also the distinct role that artefacts played in the collaboration on developing and maintaining the Azureus program.

PART 3: ANALYSIS AND DISCUSSION

8. Knowledge Transfer in The Azureus Project

This chapter attempts to show how the Azureus Community shows characteristics of an active community of practice. In light of a distinct collaborative use of artefacts, as well as of its CoP traits, it is argued that knowledge transfer dynamics in the Azureus project takes place through a cyclic interaction between CoP characteristics and the artefacts. It concludes by offerings some implications and suggestions for further practice.

8.1 The Azureus Community

In this section I analyze interaction among the Azureus members, as well as between users and members as observed in the Azureus IRC channels.

This interaction was marked by a distinct informality. Conversations between members didn't really have a start and an end as made visible by a near complete absence of "hello" or "goodbye" messages except from users entering or leaving the channels. Conversations were often conducted as wholly or partially asynchronous, and the parties would just pick up the conversation where it was left off. In addition to general IRC jargon, community-specific lingo and abbreviations was common. For example, "DHT" meant the distributed hash table used by Azureus, "az" was commonly used instead of "Azureus" and numbers like 2.3.0.2 described different versions of Azureus. Most members in the Azureus community were logged in to the channels for the greater part of the day, and some members were on continuously. Although obviously not active the entire time, it seemed clear that being present in the Azureus IRC channels was a part of their daily life. Also, there was no clear distinction between work and non-work; although Azureus-related issues was the most common topic, it was intermixed with discussions about general technical issues or chat about topics like motorcycles or the news, as well as just "messaging about" – sharing funny stories and socializing in an informal manner. In mixing social life with work the Azureus community displayed traits of a "lived sociality" as described by Wenger (1991).

In this "messaging about" members displayed several reoccurring inside jokes and pranks; In the #azureus-users channel a 'bot' named Azbot was used to control chat in the channel by kicking out people who violated the channel rules such as providing links to illegal sites, overusing capitalized letters or using excessive character repeats such as "hello!!!!!!". Azbot

was programmed not to kick established members in the channel for these violations, and members would tease users by typing things that would normally warrant a “kick” from the channel. At one time a user was kicked for this violation, and as he came back again one member promptly typed “beeeeeeeee careful.....”, much to the confusion of the recently kicked user. Azbot also enabled members to talk “through” it in such a way that what the member typed would apparently originate from Azbot instead of the member. On several occasions members used this feature to make it seem like the bot had an advanced Artificial Intelligence by for example having it reply to comments about it, in order to astound unknowing users and amuse themselves.

In performing these pranks they flaunted their position within the community and it seemed clear to me that this was something they took pride in it. Supporting my impression was the observation that position and status in the community was something that was visibly vied for. If someone said something wrong there was a “jump” to correct it and put that person in place. Likewise if a non-trivial question was posed, there was visible competition among “people in the know” to answer it first, and since others quickly would correct a wrong or incomplete answer there was incentive to answer it properly in order to assert position in the community. For more peripheral members this presented a chance to be recognized as participating members. Here we see the contours of Thompson’s virtuous circle as described in chapter 3. The more people participate, the more they learn, and the more they identify with and become prominent within a group. This competitiveness would also manifest itself in involved discussions where an objective seemed to be to question or find something wrong with statements others made. An interesting observation was that even though the core developers sometimes participated in this discourse, only very rarely did any non-core member protest or question them. Likewise, the core developers did not display the same urge to put others in place. They were the ones who had been members since the early days. They had been there longest, participated most, and identified the strongest, and clearly were the more “legitimate”. Frequently referred to as “the devs”, their status was apparent and undisputed, and they did not seem to need to actively assert it. I see this status as evidence supporting that the dynamics of Thompson’s virtuous circle were in place within the Azureus community. This claim is strengthened by one of the core developers’ description of his own participation, given in an interview with sourceforge in connection with the Azureus project being the sourceforge “Project of the Month”:

“I first installed Azureus about a year ago, when it was still in its infancy. Because there were a few bugs that bothered me and some features missing that I wanted, along with my desire to get into Java programming, I downloaded the source code and began playing with it. My first real contribution to the project was the setting up of a nightly CVS snapshot Web page (which I still maintain). Once I became more familiar with the code, I began to contribute patches on a regular basis, eventually becoming a core developer.”

In my view his description incorporates all aspects of the circle; participation, learning and identity.

CoPs are seen as effective learning settings, and evidence of their existence should usually be a welcome sign. However, members eagerly asserting their identity can also take a somewhat negative manifestation. One member would on several occasions treat users in a less than courteous manner. While not a core developer, this member was one of the most visible and active members in the community. Although young, he was proficient in several languages and was an “op” in most of the language-specific user channels, as well as in #azureus-users and #azureus. While being very active in answering questions from users, he would display very little patience with them, and would often kick - or even ban – users for minor offences, or on some occasions even for just rubbing him the wrong way. In accordance with the virtuous circle this member had a high degree of participation, was clearly very knowledgeable, and displayed a strong sense of identity which he asserted on every opportunity. Although his actions were at times inappropriate, and although some users complained about him, his role in the community in regards to being expelled or reprimanded was not discussed publicly. Although this delicate matter could have been handled in private messages in which I was not privy to, it did not seem that a core developer or someone influential told him to treat users better, as his mannerism was more or less consistent throughout the period of observation. I see this as an example of the community’s ability to tacitly share an assessment of the appropriateness of his actions, and although possibly on the verge of acting inappropriately, he did not cross the invisible line.

This member was not the only one showing derogatory attitudes towards non-members. Using the Azureus program requires a small degree of prerequisite knowledge, and users asking elementary questions or otherwise displaying lacking knowledge about Azureus were on more

than one occasion referred to as “idiots” or similar upon their departure. An often used term among the community members was “pebkac” – Problem Exists Between Keyboard And Computer, indicating that the user was causing the problems, and not hardware or software. In general there was a clear distinction between “us” and “them”, something which I see as a good indication of a sense of belonging to - and identifying with – a community; a healthy sign as long as it doesn’t negatively affect customer relations to a significant extent. At one time a user asked a developer what the best BitTorrent client was, upon which the developer became visibly annoyed and replied “I am an Azureus contributor, what do you expect me to say?” - A good example of the degree that these people identified with their community and their product.

8.2 Collaboration in the Azureus Project

In the previous section I discussed community-related interaction in the Azureus project, and related it as indicators of a CoP. In this section I discuss interaction which is more relevant to how Azureus developers collaborated on making their product, primarily as observed in the #azureus and #azureus-dev channels.

As described in chapter 4, narratives are a central factor in explaining how knowledge is transferred in CoPs, in the way that they convey sequence of behaviour, as well as the facts and their context allowing for a shared understanding of the content in the narrative.

With so many other characteristics of a CoP I initially expected to see widespread use of narratives in the Azureus community as well. However, to my surprise I observed a near total absence of work related narratives in the project. The nature of collaboration instead usually was characterized by providing links or cut/pasting code snippets or software feedback and a discourse around the problem in a synchronous manner. Rather than explaining the problem, the problem was *presented* for others to experience. After they had muddled around the problem presented by the link/code/message, developers would often return to the problem at a later stage, and since discussions were conducted openly, this discontinuity allowed developers not present at the time to also chime in; since text in the IRC channels is not ephemeral like a physical conversation is, they would have access to the links, code snippets or similar as well as the discussion on the topic. Awareness of changes made to the code, and oversight of the problem was facilitated through the changelog and CVS.

Turnbull mentions that in designing or creating templates, as discussed in chapter 4, the artefact was circulated, discussed, and contributions were given by masons and architects in an incremental and discontinued process. This process seems inherently similar to the way Azureus developers collaborated using artefacts, and this similarity is elaborated in the next section.

8.3 Knowledge Transfer in the Azureus Project

A glance back at table 1 confirms the presence of almost all of Wenger's indicators that a CoP has formed: shared styles, discourses, and a customary manner of interaction that appears idiosyncratic to an outside observer; participative interaction – although sometimes discontinued, still in the manner of a sustained conversation requiring little prior explanation, a link between continued participation and personal identification with the group, and the continued evolution of a shared repertoire.

In displaying these CoP traits it seems prudent to assume that Azureus community members share elements of the cognitive dimension of their tacit knowledge among themselves through a socialization process as described by Nonaka (1998), and relating to the 'tuning' that Wenger (1998) speaks of. Supporting this assumption is the observation of shared perceptions, understandings and beliefs as described in section 8.1.

As discussed in chapter 4, this shared knowledge and background is an enabling factor for tacit knowledge transfer through artefacts, and is facilitated by meaningful discourse with the artefact at hand. In this process both the "skill" and "cognition" dimensions of tacit knowledge can be transferred. On the same note, collaboration around links, code snippets or software output – the artefacts in the Azureus case – is an enriching factor related to social interaction in that aspects of these artefacts can trigger discourse around their attributes, which again can result in design ideas and decisions which further embed the collaborators shared tacit knowledge in the artefact. I hold that that tacit knowledge transfer in the Azureus community of practice at least partially can be explained through the cyclic interaction introduced in chapter 4, and as described in this chapter. Moreover, the near total absence of work related narratives suggests that some 'other' mechanism for tacit knowledge transfer had taken its place, and I believe the described cyclic interaction to play a significant part in this substitution.

It is conceivable that a community could display CoP traits without transferring tacit knowledge, and that the role and importance of artefacts in knowledge transfer is a purely theoretically construct made by the researcher. However, the success of the Azureus project and the relatively good track record of OS projects in general is such that it seems unlikely that this would be achievable without at least some degree of tacit knowledge transfer. This paper should be seen as a contribution in order to explain how this tacit knowledge transfer can take place in online communities such as the Azureus project.

In this paper I have attempted to show how the Azureus project overcomes the problem of tacit knowledge transfer by social interaction as described by CoP, as well as by re-experience and tacit knowledge transfer via artefacts and their surrounding discourse, related to each other through a cyclic interaction. I offer this concept and its adaptation as an important knowledge sharing dynamic, as well as a contributing factor for success in open source projects.

8.4 Implications and Suggestions for Further Study

While narratives are primarily seen as a knowledge sharing mechanism in CoP, they also enable *externalization* (Nonaka, 1998) by the narrator. In replacing narratives with the cyclic interaction repeated in this paper, this particular dynamic is lost. Although these processes are overlapping in function, and that it seemed that the latter had replaced the former, I do not believe them to be mutually exclusive. If a conscious effort is made to supplement artefact presentation with a narrative, the externalization mechanism could be tapped into, presumably for the benefit of the narrator and for the listeners.

Another action that would presumably aid the knowledge transfer mechanisms is to aid internalization by having programmers document code and describe classes in a certain way using metaphors and analogies.

However, both these actions would entail a level of consciousness around collaboration methods which the Azureus project currently does not display. Open source projects tend to be more evolutionary with little degree of control (Mockus et al. 2002), and it would seem hard to convey the importance and meaning of such actions to a fluctuating and largely autonomous member base with varying degrees of periphery. That is not to say that there is

nothing a core group of developers could do to nurture CoP dynamics and knowledge transfer, but it would entail practical problems related to implementation that are beyond the scope of this paper.

I will reserve myself from suggesting further implications for practice until the importance of the described knowledge sharing dynamics in this paper is uncovered. In order to judge the importance of these factors, a larger study observing several OS projects, and which identifies varying degrees of characteristics of these dynamics and relates them to the success of their respective projects would seem a relevant matter of investigation. If they are found to be as important as I suspect they are, further investigation around their adoption and implementation would seem pending.

9. References

- Ackerman** M S, Palen L, (1996), The Zephyr Help Instance: Promoting Ongoing Activities in a CSCW System, in CHI96, April 13-18, Vancouver, Canada, ACM Press, p268-275
Baecker (ed.): *Readings in Groupware and Computer-Supported Cooperative Work: Assisting*
- Berg**, M. (1997). On Distribution, Drift and the Electronic Medical Record: Some Tools for a Sociology of the Formal, *Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work*, ECSCW'97, Kluwer, pp. 141-156.
- Berg, Marc**. Accumulating and Coordinating: Occasions for Information Technologies in Medical Work. *Computer Supported Cooperative Work* 8: 373-401, 1999.
- Berntsen**, Kirsti; Munkvold, Glenn and Østerlie, Thomas (2004). Community of Practice versus Practice of the Community: Knowing in collaborative work, *Proceedings of the 27th Information Systems Research Seminar in Scandinavia*, August 14-16 2004, Falkenberg, Sweden. Reprinted in the ICFAI Journal of Knowledge Management, Desember, 2004.
- Bollinger**, T., Nelson, R., Self, K. M., and Turnbull, S. J. 1999. Open-Source Methods: Peering Through the Clutter, *IEEE Software*, vol. July/August, 8-1.1
- Brown**, J.S. and Duguid, P (1991), Organizational Learning and Communities of Practice: A unified View of Working, Learning and Innovation, *Organization Science*, Vol. 2, No. 1, pp. 40-56.
- Button**, Graham and Richard Harper (1996): The Relevance of 'Work-Practice' for Design. *Computer Supported Cooperative Work*, vol. 4, pp. 263–280.
- Choo**, C. W., Detlor, B., and Turnbull, D. (2000), *Web Work: Information Seeking and Knowledge Work on the World Wide Web*, Kluwer Academic Publishers, Dordrecht.
- Choo**, C. W., Detlor, B., and Turnbull, D. (2000), *Web Work: Information Seeking and Knowledge Work on the World Wide Web*, Kluwer Academic Publishers, Dordrecht.
- Conkar**, T. *Submitted for the Degree of Doctor of Philosophy* (1999).
- Cook**, S. D. N. and Brown, J. S. (1999), 'Bridging Epistemologies: The Generative Dance Between Organizational Knowledge and Organizational Knowing', *Organization Science*, Vol. 10, No. 4, pp. 381-400. Cornell, CA: Cornell Univ. Press.
- December J**, (1997), Notes on Defining a Computer Mediated Communication, World Wide source: [Http://www.december.com/cm/mag/1997/jan/december.html](http://www.december.com/cm/mag/1997/jan/december.html)
- December J**, (1997), Notes on Defining a Computer Mediated Communication, World Wide source: [Http://www.december.com/cm/mag/1997/jan/december.html](http://www.december.com/cm/mag/1997/jan/december.html)
- Dibona**, C., Ockman, S. S., and Stone, M. 1999. *Open Sources: Voices from the OpenSourceRevolution*. Sebastopol, CA: O'Reilly

Dourish P, (1991), Godard: A Flexible Architecture for Audio/Video Services in Media Space, in *Technical Report EPC-1991-134*, Rank Xerox

Dourish P, (1992), Computational Reflection and CSCW Design, in *Technical Report EPC-1992-102*, Rank Xerox

Dourish P, (1993a), Designing for Change: Reflective Metalevel Architectures for Deep Customisation in CSCW, in *Technical Report EPC-1993-101*, Rank Xerox

Dourish P, (1993b), Towards a Reflective Model of Collaborative Systems, in *Technical Report EPC-1993-114*, Rank Xerox

Dourish P, (1995), Accounting for Systems Behaviour: Representation, Reflection and Resourceful Action, in *Technical Report EPC1995-101*, Rank Xerox

Dourish P, Adler A, Henderson A, Bellotti V, (1994), Your Place or Mine? Learning from Long-term Use of Video Communication, in *Technical Report EPC-1994-105*, Rank Xerox

Dourish P, Button G, (1996), Technomethodology: Paradoxes and Possibilities, in Proceedings of CHI96 Human Factors in Computing Systems, 13-18 April and in *Technical Report EPC-1996-101*, Rank Xerox

Edwards, P.N. (1996): *The Closed World. Computers and the Politics of Discourse in Cold War America*. Cambridge: MIT Press.

Endsley M.R.: 1995, Toward a Theory of Situation **Awareness** in Dynamic Systems, *Human Factors*, 37(1), pp.65-84

Fischer G, (1995), Distributed Cognition, Learning Webs and Domain Oriented Design Environments, in *Proceedings of CSCCL95, October 1995*, p125-129

Gay G, Lentini M, (1995), "Use of Communication Resources in a Networked Collaborative Design Environment", Interactive Multimedia Group, Cornell University, also available at WWW source *Online Journal of Computer Mediated Communication* 1(1), Los Angeles,

Gray, B. *Journal of Distance Education*. 2004. vol. 19, No 1, 20-35.

Greenberg S, Bohnet R, (1991), Groupsketch: a Multi User SketchPad for Geographically Distributed Small Groups, in Graphics Interface 1991, 260 *Proceedings of Graphics Interface 91*, Calgary, Alberta, p 207-215, also available as Research Report 90-414-38, Department of Computer Science, University of Calgary, Alberta

Hammersley, Martyn. (2000). "Taking Sides in Social Research: Essays on Partisanship and Bias." London: Routledge.

Haraway, Donna J. (1991): "Simians, Cyborgs, and Women: The Reinvention of Nature". New York: Routledge.

Hayles, N. Katherine (1994): *Boundary Disputes: Homeostasis, Reflexivity, and the Foundations of Cybernetics*. *Configurations*, vol. 3, pp. 441–467.

Hemetsberger, A. and Reinhardt, C. *Accepted for The Fifth European Conference on Organizational Knowledge, Learning and Capabilities in Innsbruck, Austria, 2004*

Herbsler, J.D., and Grinter, R. E. 1999. Splitting the Organization and Integrating the Code: Conway's Law Revisited. In *21st International Conference on Software Engineering (ICSE 99)*, Los Angeles, CA.
<http://cwis.usc.edu.dept/anneberg/vol1/issue1/IMGJCMC/ResourceUse.html>

Hutchins E, (1990), *The Technology of Team Navigation*, in *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, edited by Gulegher J R, Kraut R E, Egido L, Lawrence Erlbaum Associates, New York

Hutchins E, (1995), *How a Cockpit Remembers its Speed*, in *Cognitive Science* 19, p 265-288

Kindberg T, (1997), *Mushroom – A framework for Collaboration and Interaction Across the Internet*, Source,<http://www.dcs.qmw.ac.uk/research/distrib/Mushroom/CSCWeb.html>

Kogut, B. and A. Metiu (2001). "Open-Source Software Development and Distributed Innovation." *Oxford Review of Economic Policy* 17(2): 248-264.

Kollock, P. and M. Smith (1997). "Communities and Cyberspace". New York, Routledge.

Krochmal, M. 1999. *Linux Interest Expanding*, in *TechWeb*. Available at <http://www.techweb.com/wire/story/TWB19990521S002>

Kuhn, T. S. (1962), "The Structure of Scientific Revolutions", University of Chicago Press, Chicago.

Kuwabara, K. (2000). "Linux: A Bazaar at the Edge of Chaos." *First Monday* 5(3).

Kyng M, (1995), *Making Representations Work*, in *Communications of The ACM*, September 1995, Vol. 38, No 9, p46-55

Lanzara, G. F. and M. Morner (2003). "The Knowledge Ecology of Open-Source Software Projects". 19th EGOS Colloquium. Copenhagen.

Lave, J. and Chaiklin, S. (eds.) (1993) "Understanding Practice: Perspectives on Activity and Context", Cambridge: University of Cambridge Press.

Lave, J., E. Wenger. 1991. "Situated Learning: Legitimate Peripheral Participation." Cambridge University Press, Cambridge, UK

Lerner, J. and J. Tirole (2001). "Some Simple Economics of Open Source", L'Institut d'Economie Industrielle.

McConnel, S. 1999. Open-Source Methodology: Ready for Prime Time?, *IEEE Software*, vol. July/August, pp. 6-8.

Miles V C, McCarthy J, Monk A, Harrison M, (1995), "Using a Minimal System to Drive the Conceptual Analysis of Electronic Conferencing", The University of York

Mockus, A., Fielding, R., and Herbsleb, J. "Two Case Studies of Open Source Software Development: Apache and Mozilla," *ACM Transactions on Software Engineering and Methodology*, 11(3):309346, 2002

Moser L E, Melliar-Smith P M, Agarwal D A, Budhia R K, Lingley-Papadopoulos C A, (1996), Totem: A fault Tolerant Multicast Group Communications System, in *Communications of the ACM*, April 1996, Vol. 39, No 4, p54-63

Mynatt E, Adler A, Ito M, O'Day V, (1997), Design for Network Communities, in CHI97, 22-27 March, Atlanta, Georgia, ACM Press, p210-211

Nemiro, J. E. (2002). "The creative process in virtual teams." *Creative Research Journal* 14(1): 69-83.

Nonaka, I. (1994), A Dynamic Theory of Organizational Knowledge Creation, *Organization Science*, Vol. 5, No. 1, pp. 14-37

Nonaka, I. and Takeuchi, H. "A theory of the firm's knowledge-creating dynamics", in A.D.Chandler jr, P.Hagstrøm and Ø.Sølvell (eds): *The dynamic firm. The role of technology, strategy, organization and regions*, Oxford Univ. Press, 1998, p.214-241.

Nonaka, I. and Takeuchi, H. (1995), "The knowledge-creating company: how Japanese companies create the dynamics of innovation", Oxford University Press, New York, NY.

Nonaka, I., P. Reinmoeller, et al. (2000). "Integrated IT Systems to Capitalize on Market Knowledge". *Knowledge Creation - A Source of Value*. G. von Krogh, I. Nonaka and T. Nishiguchi. London and New York, NY, Macmillian Press: 89-109.

Norman D A, Zhang J, (1994), Representations in Distributed Cognitive Tasks, in *Cognitive Science* 18, p 87-122

Orlans, Harold. (1973). "Contracting for Knowledge". San Francisco, CA: Jossey-Bass Publishers, Inc.

Orr, J. (1996), "Talking about machines: An ethnography of a modern job", Cornell, CA: Cornell Univ. Press.

Pederson E R, Sokoler T, (1997), AROMA: Abstract Representation of Presence Supporting Mutual Awareness, in *CHI97*, 22-27 March, Atlanta, Georgia, p51-57

Perens, B. 1999. The Open Source Definition. in *Open Sources: Voices from the Open Source Revolution*. DIBONA, C., OCKMAN, S. AND STONE, M. Eds. Sebastopol, CA: O'Reilly, 171-188.

- Petre**, M Blackwell A F, Green T R G, (1996), Cognitive Questions in Software Visualisation, from <http://www.mrcapu.cam.ac.uk/~alan.blackwell/CognitiveQuestions.html>
- Polanyi**, M. (1958/1962), "Personal Knowledge", Corrected edition, Routledge, London.
- Polanyi**, M. (1966/1997), The Tacit Dimension, in Prusak, L. (Ed.) *Knowledge in Organizations*, Butterworth-Heinemann, Newton, MA, pp. 135-146.
- Powell** D, (1996), Group Communication, in *Communications of the ACM Journal*, April 1996, Vol. 39, No 4, p 50-53
- Raymond**, E. S. (1999). "The Cathedral and the Bazaar", First Monday.
- Robinson**, Mike (1992): Computer Supported Co-operative Work: Cases and Concepts. In *Ronald M.*
- Rogoff**, B., Turkanis, C. G. and Bartlett, L. (eds.) (2001) "Learning Together: Children and Adults in a School Community, New York": Oxford University Press.
- Roseman** M, (1996), Managing Complexity in TeamRooms, a Tcl Based Internet Groupware Application, <http://www.cpsc.ucalgary.ca/projects/grouplab/papers/teamroomstcl96/>
- Scaife** M, Rogers Y, Aldritch F, Davies M, (1997), Designing for or Designing with? Informant Design for Interactive Learning Environments, in *CHI 97*, 22-27 March, Atlanta, Georgia, ACM Press, p343-350
- Scardamalia** M, Bereiter C, (1992), "Collaborative Knowledge Building", in De Corte E, Linn M C, Mandl H, and Verschaffel L, (Eds.), *Computer Based Learning Environments and Problem Solving*, Springer Verlag, Berlin, p41-66
- Schiper** A, Raynal M, (1996), From Group Communications to transactions in Distributed Systems, in *Communications of the ACM*, April
- Schmidt**, Kjeld and Liam Bannon (1992): Taking CSCW Seriously: Supporting Articulation Work. *Computer Supported Cooperative Work*, vol. 1, pp. 7-40.
- Smith**, M. K. (2003) 'Communities of practice', *the encyclopedia of informal education*, www.infed.org/biblio/communities_of_practice.htm. Last updated: 30 January 2005.
- Stenmark**, D. (2002). Information vs. Knowledge: The Role of intranets in Knowledge Management. In *Proceedings of HICSS-35*, IEEE Press, Hawaii, January 7-10, 2002
- Thompson**, M. (2005). Structural and Epistemic Parameters in Communities of Practice. *Organization Science*, forthcoming.
- Tsoukas**, H. (1996), The Firm as a Distributed Knowledge System: A Constructionist Approach, *Strategic Management Journal*, 17, Winter Special Issue, pp. 11-25.
- Tuomi**, I. (2001). "Internet, Innovation, and Open Source: Actors in the Network", First Monday.

Turnbull, D. (1993), The ad hoc collective work of building Gothic cathedrals with templates, string and geometry, *Science, Technology and Human Values*, Vol. 18, No. 3, pp. 315-340.

Vixie, P., P. 1999. Software Engineering. In *Open Sources: Voices from the Open Source Revolution*, C. Dibona, S. Ockman, and M. Stone, Eds. Sebastopol, CA: O'Reilly, 1999, 91-100. Vol. 18, No. 3, pp. 315-340.

von Hippel, E. and G. von Krogh (2002). Exploring the Open Source Software Phenomenon: *Issues for Organization Science*.

von Hippel, E. and G. von Krogh (2003). "Open Source Software and the Private-Collective Innovation Model : Issues for Organization Science." *Organization Science: forthcoming*.

Wayner, P. (2000). "Free for All: How Linux and the Free Software Movement Undercut the High-Tech Titans". New York, Harper Business.

Wenger, E. (1998). Communities of Practice: Learning as a Social System. *Systems Thinker*, June 1998.

Wenger, E. (1998) Communities of Practice. Learning as a social system, *Systems Thinker*, <http://www.co-i-l.com/coil/knowledge-garden/cop/lss.shtml>. Accessed December 30, 2002.

Whittaker S, Frohlich D, Daly-Jones O, (1994), Informal Workplace Communication: What is it like and how may we support it?, in *Human Factors in Computing Systems CHI94, Celebrating Interdependence*, April 24-28, Boston, Massachusetts, p 131-137

Wright P C, McCarthy J C, Monk A F, (1992), Coherence in Text Based Electronic Conferencing: Coupling Text and Content, *Journal of Language and Social Psychology*, Vol. 11, No 4, p267-277